

Proceedings of NZCSRSC '07, the Fifth New Zealand Computer Science Research Student Conference

Te Kohinga Mārama Marae
The University of Waikato
Hamilton, New Zealand

April 10–13th, 2007

<http://www.cs.waikato.ac.nz/nzcsrsc2007/>

Preface

These Proceedings contain papers by Computer Science graduate students from all over New Zealand and overseas who participated in the 5th NZ Computer Science Research Student Conference (NZCSRSC'07), April 10–13th, 2007, at the Waikato University Marae in Hamilton.

The conference was organised by a group of postgraduate students with support from a few academic and other staff members. With 50 submissions and 70 participants this is by far the largest NZCSRSC held so far. A novel student-driven peer review process was adopted: each submission was reviewed by three other students who had also submitted papers. Juggling this fairly complicated process and putting all submissions together into an interesting conference program and these Proceedings was capably accomplished by Program Chairs Judy Bowen, Stefan Mutter and Andrea Schweer.

Conference organisers have been lucky to attract four renowned keynote speakers, who came at their own expense and spent hours answering students' questions: Tyrone McAuley from Sidhe in Wellington, Craig Nevill-Manning from Google in New York, Ian Foster from the University of Chicago and Nigel Scott from Park Road Post in Wellington. Fruitful discussions took place in a panel on *Where to head off once graduated*, a tutorial session on *How to successfully complete a research degree*, and the *Suits 'N' Geeks* networking event sponsored and organised by WaikatoLink, where NZ industry representatives chatted with NZ graduate students.

The academic program was accompanied by many less formal events on Te Kohinga Mārama Marae. The social program, enthusiastically organised by Sam Bartels, Marian McPherson, Veronica Liesaputra, Scott Raynel and Te Taka Keegan, included a traditional Maori welcome, the Kawa Powhiri, followed by a Hangi dinner on the first evening, a relaxed BBQ at the mineral waters of the Waingaro Hot Springs, and a Hakere or ceremonial feast to finish off the conference and send guests away with full bellies.

Huge efforts have been put into this conference by all members of the organising team, including (not mentioned above): Rob Akscyn, Reynaldo Giganto, Kathryn Hempstalk, Annika Hinze, Anna Huang, Ian McDonald, David Milne and Gottfried Vossen from the University of Waikato; Craig Anslow and Sergio Hernandez from the University of Victoria; Phil McLeod from the University of Otago; Guy Kloss and Natalia Nehring from Massey University at Albany; Alan McKinnon from Lincoln University; Nick Hay and Carl Schulz from the University of Auckland; Tim Bell, Richard Green and Amali Weerasinghe from the University of Canterbury; and the multitude of other contributors.

Last but not least we would like to thank our supporters and sponsors, without whom this conference could not have taken place. In particular, Ian H. Witten was initiator of the whole idea and a consistent mentor for the student organisers. Mark Apperley, Dean of the School of Computing and Mathematical Sciences of the University of Waikato, enthusiastically offered support right from the very beginning and covered most of the cost from School funds. The Waikato Department of Computer Science offered to sponsor the attendance of all its students and staff members, and fully supported the local organisers, many of whom had to put their PhDs on hold. We are also grateful to our

industry sponsors: DL Consulting, Endace, WaikatoLink, and Google Inc.

Olena Medelyan

April 2007

<http://www.cs.waikato.ac.nz/nzcsrsc2007/>

Contents

Full Papers with Presentation at the Conference

In the order of presentation at the conference

Olena Medelyan: Topical Indexing with WordNet

Moffat Mathews and Antonija Mitrovic: Introducing Problem Templates and Their Effect on Learning in Intelligent Tutoring Systems

Veronica Liesaputra: Turning the page of an electronic book

Amali Weerasinghe: Towards a General Model of Self-Explanation for Constraint-based Tutoring Systems

Carl Schultz: A Framework for Supporting the Application of Qualitative Reasoning

Jay Holland and Antonija Mitrovic: A Constraint-Based Intelligent Tutoring System for the Java Programming Language

Gregory Caza: Computational Model of Plan Competition in the Prefrontal Cortex

David Milne: Computing Semantic Relatedness using Wikipedia Link Structure

Craig Anslow, Stuart Marshall and James Noble: X3D Software Visualisation

Vrushank Mehta and Napoleon Reyes: Parallel Logit-Logistic Fuzzy Colour Constancy and Automatic Parallel Colour Contrast Rule Extraction

Kon Zakharov, Antonija Mitrovic and Lucy Johnston: Intelligent Tutoring Systems Respecting Human Nature

Rachel Patel, Beryl Plimmer, John Grundy and Ross Ihaka: Exploring Better Techniques for Diagram Recognition

Sven Bittner: Increasing the Expressiveness of Subscriptions and Advertisements in Distributed Content-Based Publish/Subscribe Systems

Nancy Milik, Antonija Mitrovic and Michael Grimley: Fitting spatial ability and free-formed questions into Intelligent Tutoring Systems development

Kathryn Hempstalk: The Great Keyboard Debate: QWERTY versus Dvorak

Sam Bartels: Wireless Sensor Networks for Agricultural Applications

Mayank Keshariya and Ray Hunt: Optimised Transparent and Automated Handoff between WLAN and WWAN

Alastair Nisbet: An Improved Encryption Key Management System for IEEE 802.16 Mesh Mode Security Using Simulation

Natalia Nehring: Building in Web application security at the requirements stage: A tool for visualizing and evaluating security trade-offs

David Friggens: Model Checking Nonblocking Data Structures

Judy Bowen and Steve Reeves: Formal Methods and Refinement for User-Centred Design Artefacts

Andrew Gin, Nilufar Baghaei and Ray Hunt: Performance of Evolving IEEE 802.11 Security Architectures

Full Papers with Poster at the Conference

Alphabetically by first author's last name

Jennifer Ferreira, James Noble and Robert Biddle: Interaction Designers on eXtreme Programming Teams: Two Case Studies from the Real World

Syed Faisal Hasan: Congestion Control for Data-limited Flows

Doris Jung: Collaborative Profiles for Event Notification in Healthcare: Lessons Learned from Healthcare Staff Interviews

Christof Lutteroth: A Comparison of XML and Relational Database Technology

Gilbert Notoatmodjo: Designing Ethical Investigation on Password and Online Account Management Strategies

Andrea Schweer and Annika Hinze: Augmented Memory for Conference Attendees

Christian Seifert, Ian Welch and Peter Komisarczuk: HoneyC – The Low-Interaction Client Honeypot

Jonathan Teutenberg: Accent Modification: Approaches and Issues

Yifan Zhang and Peter Komisarczuk: Improving the Stability of Transmission in Mobile Ad Hoc Networks

Short Papers with Poster at the Conference

Alphabetically by first author's last name

Isaac Jonathan Freeman: Ad Hoc Visual Documentation

Reynaldo T. Giganto: Extracting Use Case Specifications for Sequence and Class Diagrams Generation

Vishal Jain, Nikola Kasabov, Paulo C. M. Gottgtroy, Lubica Benuskova and Frances Joseph: Brain Gene Ontology: Tool to facilitate Education and Research in Neuroinformatics area

Guy K. Kloss, Napoleon H. Reyes, and Ken A. Hawick: Integrative Architecture for Concurrent Artificial Intelligence in Robotic Systems

Stefan Mutter and Bernhard Pfahringer: A Discriminative Approach to Structured Biological Data

Ben Palmer: Verifying Privacy Preserving Auctions

Yuan Tian, Keith Unsworth, and Alan McKinnon: Simulation for LEGO MINDSTORMS™ Robotics

Anna Wingkvist and Jason Alexander: Was it Pod Worthy? A Preparatory Plan for Evaluating Podcasting in Higher Education

Short Papers

Alphabetically by first author's last name

Tobias Bethlehem: 3–4 Heap Workspace Operations

Yanbo Deng, Clare Churcher, Walt Abell and John McCallum: Using Web services to manage data from a variety of client applications

Yulong Gu: Knowledge Management Literature Review

Sergio Hernandez: Population-based Monte Carlo

Alireza Izaddoost: Accelerate process of tracing back DoS/DDoS attack

Abhilasha Keshariya: Proposed Model for Security Enhancement by Algorithm Permutations

Ian McDonald: Linux Kernel Development

Philip McLeod: Tartini: The Music Analysis Tool

Topical Indexing with WordNet

Olena Medelyan

Department of Computer Science,
The University of Waikato, Hamilton, New Zealand
olena@cs.waikato.ac.nz

Abstract. We describe how conventional automatic keyphrase indexing with domain-specific thesauri can be adapted to indexing with the lexical database WordNet. The goal of indexing is to determine the main topics of a document. WordNet organizes words into synonymous groups called synsets that represent single concepts. The proposed algorithm first maps document phrases onto WordNet synsets and then determines the most significant by exploring their statistical and semantic properties. To evaluate the algorithm we compare keyphrases assigned manually by human indexers with terms assigned automatically by the original thesaurus-based keyphrase indexing algorithm and by the one adapted for indexing with WordNet.

Keywords: keyphrase indexing, WordNet

1 Introduction

In a well organized document collection, full text documents—such as academic publications, electronic and paper books, blog articles and websites—are accompanied by a set of several phrases that reflect their main topics. These phrases, known as keyphrases, subject headings or content tags, are assigned manually either by authors of the documents, or by professional human indexers. This so called topical indexing provides a natural way of storing and accessing digital collections. However, manual indexing requires expensive resources and is practically impossible with the constantly growing number of electronic documents. The goal of this PhD project is to develop an automatic indexing algorithm that assigns keyphrase sets of equal or greater quality compared to manually assigned ones.

In topical indexing, keyphrases can either be freely chosen or driven from a thesaurus. Thesauri organize lexical knowledge of a human language by defining a set of terms used in a specific domain and semantic relations between them (e.g. if two terms are synonyms). Compared to free indexing, controlled indexing with a thesaurus increases the indexing consistency over the document collection. Thesauri can also serve as browsing tools to access documents through the semantic hierarchy of terms. However, thesauri have to be created and maintained manually for each specific domain and language, which is an expensive and time-consuming process.

WordNet¹ developed at Princeton University, is the largest lexical database of English language [1]. It covers a general domain with over 160,000 English words. It has been extensively used as a replacement for domain-specific thesauri in various natural language applications and is explored in this paper as a controlled vocabulary for keyphrase indexing.

The structure of WordNet has significant differences to conventional domain-specific thesauri and its usage for topical indexing is not straightforward. We first discuss how terms and semantic relations between them are organized in both resources and then propose a method for adapting WordNet for the keyphrase indexing task. Given a keyphrase extraction algorithm designed for indexing with a domain-specific thesaurus, we describe its re-implementation for indexing with terms and synsets from WordNet. In a small scale experiment we evaluate the performance of this new algorithm by comparing its keyphrases to those assigned by 6 professional human indexers on a set with 10 documents. These initial results provide an answer to the main question of this brief study within the PhD project: Is WordNet a useful alternative for domain-specific thesauri in the task of topical indexing?

2 Domain-specific Thesauri vs. WordNet

The main reason why domain-specific thesauri are used for keyphrase indexing is that they increase the indexing consistency over the entire document collection, and therefore the indexing quality. If two very similar documents are indexed consistently, i.e. with the same terms, user can efficiently locate them with a single keyphrase search. A thesaurus defines a controlled vocabulary with allowed terms, called *descriptors*, and a number of semantically equivalent *non-descriptors*, linked to them by the USE (and inversely the USED-FOR) relations. For example, documents about “laptops” and “notebooks” could be indexed with the same descriptor “portable computers”, but they are accessible with any of these terms as long as they are listed in the vocabulary.

WordNet defines over 110,000 unique nouns that can be used as index terms. However, here semantic equivalence or *synonymy* is represented differently from thesauri. All nouns are organized into *synsets* (approximately 80,000), which are sets of words defining distinct concepts. Each synset represents a particular meaning of one word or several synonymous words, and each word may appear in several synsets. For example, the word “pupil” appears in the following synsets:

- Synset 1. student, pupil, educatee (*a learner who is enrolled in an educational institution*)
- Synset 2. pupil (*contractile aperture in the iris of the eye*)
- Synset 3. schoolchild, school-age child, pupil (*a young person attending school (up through senior high school)*)

Each synset contains a *gloss*, a short description of its meaning in brackets. In domain-specific thesauri, the number of different meanings for each term is significantly lower than in WordNet. Ambiguity is resolved by adding scope notes,

¹ <http://wordnet.princeton.edu/>

that similar to glosses define the specific usage of a descriptor and forward the user to another descriptor if necessary. E.g. descriptor “epidermis” only refers to the epidermis of plants; for the epidermis of animals the descriptor “skin” is used instead. Both thesauri and WordNet define a dense structure of semantic relations between terms. However, in WordNet relations inter-connect the whole synsets, with the exception of synonymy, which is an intra-synset relation as described above. Beside synonymy, both resources define hierarchical relations between generic and specific terms. In thesauri, they are represented as inverse broader (BT) and narrower (NT) links (e.g. “apple” →_(BT) “fruit”), in WordNet with *hypernym* (broader) or *hyponym* (narrower) links, e.g. synset 3 for the word “pupil” (“schoolchild”) has a direct hypernym “young person” and two hyponyms: a “boarder”, a pupil who lives at school during term time, and “schoolboy”. All other types of semantic relatedness between terms in a thesaurus are expressed by the bi-directional links called related (RT), e.g. “eye” →_(RT) “vision”. WordNet differentiates further types of semantic relatedness, such as “part-whole”, membership, substance, coordinates or “sister” relations, antonyms and domain terms.

While for each concept a thesaurus defines a preferred term, and zero or more alternative terms, concepts in WordNet are represented as a set of equal synset members. WordNet is a general resource and has higher term ambiguity than a domain-specific thesaurus. These two main differences present a challenge for using WordNet as a controlled vocabulary for keyphrase indexing.

3 Indexing with WordNet

Keyphrase indexing with a domain-specific thesaurus is the process of selecting thesaurus terms that represent document’s main topics. In WordNet, the smallest meaningful units are not terms, but synsets, representing distinct concepts. Therefore, keyphrase indexing with WordNet means identifying the main synsets in a document. This process can be designed in two stages:

1. Candidate selection, where all WordNet synsets mentioned in the document are identified.
2. Filtering, where the most important synsets are chosen and presented in form of keyphrases to the user.

Selection of candidates from a domain-specific thesaurus means simple mapping of document’s words and phrases to thesaurus terms. The same approach in WordNet would produce too many mappings due to the frequent ambiguity of words. At the same time, WordNet possesses more semantic information about terms than a thesaurus, which can improve the filtering performance of the indexing algorithm.

3.1 Candidate Selection

Mapping of document’s words onto synsets in WordNet is not straightforward, because almost every word appears in several synsets. Ideally, the meaning of each word in the document needs to be disambiguated; i.e. the algorithm should choose

only one synset per word, one that represents the concept relevant to this document. Word sense disambiguation algorithms usually take advantage of semantic relations [2]. For example, if “pupil” (cf. Section 2) needs to be disambiguated, the algorithm looks for semantically related synsets in its vicinity. The term “boarding school” would indicate that sense 3 is correct, while “vision” is an indicator for the sense 2.

In many cases disambiguation is undecidable, because no semantically related words can be found, or several mappings are possible. The latter is the result of the fine granularity of WordNet’s synsets. Even very close meanings, e.g. synsets 1 and 3 for the word “pupil”, are separated into distinct concepts. For natural language applications this granularity is unnecessary and presents additional difficulties for the algorithms. They are not accurate enough to be used in real-world applications. The best performing supervised algorithms have an accuracy of about 75% [2], but require manually sense-tagged corpora.

3.2 Filtering

Determining the most significant among the candidate synsets can be realized with techniques from conventional keyphrase indexing.

Turney [3] is using a number of parameters for each candidate *n-gram*² appearing in the document. A genetic learning algorithm determines optimal parameter settings using training data and the top ranked candidate n-grams are selected as keyphrases. Marko et al. [4] suggest computing the probability of a candidate term being a keyphrase as a product of conditional probabilities of its morpheme trigrams in the documents to which this candidate term was assigned manually. Common are fixed weighting schemes based on parameters such as frequency and position of a phrase in the document [e.g. 5].

Witten et al. [6], Hulth [7] and Medelyan and Witten [8] use machine learning techniques, where each candidate phrase is an instance, and its probability of being a keyphrase is computed by using a learning scheme that integrates different features, i.e. syntactic and semantic properties of a phrase, such as TF×IDF³, position of the first occurrence in the document and part-of-speech sequence. Given a training data, where the distribution of the feature values for positive and negative examples is known, a model is computed that is then applied for indexing of unseen documents. The keyphrase extraction algorithm KEA that performs controlled indexing in this way [8] outperforms free indexing approaches such as [3 and 6] and requires significantly less training data than used in [4]. Therefore, it has been adapted for indexing with WordNet concepts as described in the following section.

KEA is designed for domain-specific thesauri, where no word-sense disambiguation is required. Instead of including the complex disambiguation in the candidate selection stage, as described in Section 3.1, we consider all possible synset

² Here, an *n-gram* is a phrase appearing in a text, consisting of *n* consecutive words. Usually, all *n-grams* extracted with *n*=[1,5] are considered.

³ $\text{TF}\times\text{IDF} = \frac{\text{freq}(P, D)}{\text{size}(D)} \times -\log_2 \frac{\text{df}(P)}{N}$, where freq(*P,D*) is the number of times *P* occurs in *D*, size(*D*) is the number of words in *D*, df(*P*) is the number of documents containing *P* in the global corpus, and *N* is the size of the global corpus. This metric compares the use of a phrase in a particular document with its general use.

mappings and rely on the indirect disambiguation in the filtering stage. We assume that irrelevant word senses will have lower probability values compared to relevant synsets and will be thus eliminated from the results.

4 Indexing with WordNet Synsets in KEA

The latest KEA version 4.1⁴ was augmented with the Java WordNet Library (JWNL)⁵ to enable efficient access to the WordNet database. In the candidate selection step KEA extracts all possible n-grams from a document and sends them to JWNL to check whether they are listed as nouns in the WordNet index. If yes, synsets containing these n-grams are included into the candidate set. A synset may become a candidate through one or more n-grams that are words constituting this synset. And each n-gram may result in inclusion of one or more synsets. Each synset has an occurrence count, which is the sum of frequencies of its n-grams. The next step is to identify a subset containing the most significant synsets for each document.

In order to build a model, a training set—where keyphrases were assigned manually from a domain-specific thesaurus—is used. Since many of these keyphrases are specific terms that are not listed in WordNet, we allow candidate synsets to be positive instances even if their terms only partially match manually selected keyphrases. For example the following Agrovoc⁶ keyphrases were assigned to a sample document: “salinity”, “irrigation”, “drainage”, “salinity control”. While the first three were found in WordNet, the last keyphrase was mapped to “salinity” and “control”. Once candidate terms are identified, we calculate their feature values.

The original KEA [8] uses four features: the TF×IDF score, the position of the first occurrence of a phrase expressed as the percentage of the document preceding it, the node degree and the length of a phrase. The length feature is specific to the thesaurus and the training set used in KEA: it has been observed that indexers prefer two-word keyphrases, although the majority of terms in the thesaurus were single words. We are unable to make the same assumption about indexing with WordNet and therefore ignore this feature. The other three features can be directly applied to indexing with WordNet. For example, the node degree represents how many synsets related to a given synset appear in the document. This feature is based on the observation that if a document describes a particular topic area then it covers most of the concepts related to this area. Therefore, candidate synsets with a high node degree are more likely to be significant.

The algorithm was enhanced by a new feature called *synset coverage*. Given a synset, how many words representing this synset appear in the document? The feature represents the proportion of these words compared to synset’s overall size in words. For example, if a term “cd” was matched to two synsets {candle, candela, cd, standard candle} and {compact disk, compact disc, CD} and a term “compact disk” appears in the same document, the probability of the latter synset will be increased because 2/3 of its representative words are covered, compared to only 1/4 of the former synset.

⁴ <http://www.nzdl.org/Kea/>

⁵ <http://jwordnet.sourceforge.net/>

⁶ An agricultural thesaurus, available on <http://www.fao.org/agrovoc/>

Each candidate synset in the training set is marked as a positive or a negative example depending on whether they were identified among the manually assigned keyphrases or not. This binary feature is the class used by the machine-learning scheme. The scheme then generates a model that predicts the class using the values of the other four features. KEA uses the Naive Bayes technique, which learns two sets of numeric weights from the discretized feature values, one set applying to positive instances (“appears in the manual keyphrase set”) and the other to negative ones (“does not appear in the keyphrase set”). To assign WordNet synsets to new documents, KEA determines candidate synsets and their feature values, and then applies the model built during training. The model determines the overall probability that each candidate is highly related to the document.

The user selects the number of synsets (or terms) to be presented in the result set. The result set may contain either top ranked synsets (the set of synset words and the gloss), or keyphrases, which are the most frequent terms in the top ranked synsets. If a keyphrase appears in more than one synset, more synsets are analyzed till the pre-defined number of output keyphrases is reached.

5 Evaluation

Since there is no training data available where documents are indexed manually with synsets from WordNet, we used agricultural documents from the FAO’s repository⁷ indexed with terms from the Agrovoc. Positive examples were obtained by full and partial mapping of manual Agrovoc keyphrases to WordNet terms (cf. Section 4).

After training on 30 documents, KEA’s WordNet keyphrases were evaluated on 10 documents indexed each by six professional human indexers. Table 1 presents Agrovoc keyphrases assigned by at least two out of six indexers (column “Human Indexers”), Agrovoc keyphrases extracted automatically by the original KEA algorithm (“KEA-4.1”), and WordNet keyphrases extracted with the adapted algorithm described in Section 4 (“KEA-WordNet”). The positive observation is that most WordNet keyphrases either match manually assigned Agrovoc terms, or are their direct synonyms (e.g. “towns” and “cities” in Document 3), or terms from the same topic area (e.g. “calorie”, “undernourishment” and “fat” in Document 1). At the same time, KEA-WordNet column contains more general terms than chosen by humans or KEA-4.1 (e.g. “consumer”, “method”, “policy”).

Table 2 shows indexing consistency in the group of professional indexers and consistency of both KEA versions with these indexers. The consistency was computed over WordNet synsets, after words and phrases in each keyphrase set were matched to WordNet terms with the strategy described in Section 4.

KEA-4.1, which uses the same controlled vocabulary as the indexers, achieved high consistency with humans: only 10% lower than the consistency among indexers. KEA-WordNet has a much lower consistency with humans: about a half of what they have achieved.

⁷ <http://www.fao.org/documents/>

Document 1. The growing global obesity problem: Some policy options to address it.		
Human Indexers	KEA-4.1	KEA-WordNet
{overweight}	{overweight}	{obesity, overweight}
{taxes}	{taxes}	{tax}
{food consumption, food intake}	{food consumption}	{consumer, intake}
{feeding habits, diet}	{body weight}	{body weight, body}
{prices, price policies}	{controlled prices, price fixing}	
{fiscal policies, nutrition policies}	{policies}	{policy}
{nutritional requirements}		{nutrition}
{developing countries}	{developing countries}	
	{saturated fats}	{calorie, fat}
		{undernourishment}

Document 2. Overview of techniques for reducing bird predation at aquaculture facilities.		
Human Indexers	KEA-4.1	KEA-WordNet
{aquaculture}	{aquaculture}	{aquaculture}
{fishery production, fish culture}	{fishing operations}	
{fencing, noise, scares}	{fencing, noise, scares, ropes}	{fence, barrier, noise}
{predatory birds, noxious birds}	{birds, predators}	{bird, predation}
{bird control, control methods}		{frightening, method}
{damage}	{damage}	
		{contents}
		{modification}

Document 3. Feeding Asian cities: Food production and processing issues.		
Human Indexers	KEA-4.1	KEA-WordNet
{food policies}	{food policies, food consumption}	{policy}
{food supply}	{food supply}	
{towns}	{towns}	{city, land}
{urbanization, urban areas, rural urban relations}	{urbanization, urban areas}	{urbanization, urban area}
{urban agriculture}	{urban agriculture}	
{food production}	{new products}	{production, products}
{agricultural policies, agricultural sector, suburban agriculture}		
{asia}		{asian, asian country, vietnam}
{state intervention}		{intervention}

Table 1. Examples for keyphrase sets assigned manually and automatically from a domain-specific thesaurus and automatically from WordNet.

	Synsets per Document	Consistency
Indexers vs. Indexers	17	58
KEA-4.1 vs. Indexers	14	48
KEA-WordNet vs. Indexers	18	30

Table 2. Consistency among human indexers and consistency of two indexing algorithms with human indexers in assigning keyphrases from WordNet.

6 Conclusions

Topical indexing with WordNet differs from indexing with a domain-specific thesaurus. While conventional thesauri define a set of controlled terms that ensure consistent indexing and allow assigning domain-specific keyphrases, WordNet's structure is not designed for indexing purposes and its terminology is more general.

Due to the unique organizational structure of WordNet, indexing algorithms designed for domain-specific thesauri need re-implementation. We have proposed an indexing method that does not require additional resources such as word sense disambiguation and performs well in assigning general topics to documents.

WordNet covers only a fraction of specific terms covered by a thesaurus like Agrovoc, but its short and rather general terms are not necessarily worse keyphrases. This general indexing might be useful for other purposes than conventional indexing of domain-specific document collections. Keyphrases assigned from WordNet are similar to search queries in general search engines, which are mostly general single words. WordNet could be a suitable vocabulary for general keyphrase indexing of websites from a mixed domain, as an addition to full-text search. In a domain-specific document collection indexing with terms from WordNet would make sense in cases when laypersons without the knowledge of the domain-specific terminology need to access the documents. WordNet is also a useful alternative for indexing in domains for which domain-specific thesauri are unavailable.

The presented results for indexing with WordNet can be improved as the algorithm has not been fully adjusted to WordNet's properties to take the maximum advantage out of its rich semantic resources. Beside the word sense disambiguation component, which would be desirable in the candidate selection stage, the algorithm would also benefit from WordNet specific filtering features. Training and testing data, where each document would be manually indexed with WordNet synsets instead of Agrovoc terms, would be necessary for the further development of the indexing algorithm.

References

1. Fellbaum, C. *Wordnet: An Electronic Lexical Database*. Cambridge, Massachusetts: MIT Press (1998)
2. Mihalcea, R. and Edmonds, P., eds. In: Proc. of Senseval-3: 3rd Intern. Workshop on the Evaluation of Systems for the Semantic Analysis of Text, Barcelona, Spain (2004)
3. Turney, P.: Learning to extract keyphrases from text. Technical report, NRCC (1999)
4. Markó, K., Hahn, U., Schulz, S., Daumke, P., and Nohama, P.: Interlingual Indexing across Different Languages. In: Proc. of RIAO (2004) 82-99
5. Barker, K. and Cornacchia N.: Using noun phrase heads to extract document keyphrases. In: Proc. of the 13th Canadian Conf. on Artificial Intelligence (2000)
6. Witten, I. H., G. W. Paynter, E. Frank, C. Gutwin, and Nevill-Manning, C.G.: Kea: Practical automatic keyphrase extraction. In: Proc. of the 4th ACM Conf. on Digit. Libr. (1999)
7. Hulth, A. Combining Machine Learning and Natural Language Processing for Automatic Keyword Extraction. Ph.D. thesis, Computer and Systems Sc., Stockholm Univ. (2004)
8. Medelyan O., and Witten, I. H.: Thesaurus Based Automatic Keyphrase Indexing. In: Proc. of the Joint Conf. on Digit. Libr. (2006)

Investigating the Effectiveness of Problem Templates on Learning in Intelligent Tutoring Systems

Moffat Mathews and Antonija Mitrović

Intelligent Computer Tutoring Group, University of Canterbury, New Zealand
{moffat, tanja}@cosc.canterbury.ac.nz

Abstract. Intelligent Tutoring Systems (ITSs) provide an ideal environment for coached learning. A goal in ITS development is to maximise effective learning which provides the motivation for this research. This paper proposes the notion of *problem templates* (PTs): mental constructs used by experts to retrieve large amounts of domain-specific data to solve a problem. This research aims to examine the validity of such a construct and investigate its role in regards to effective learning within ITSs. After extensive background research, an evaluation study was performed at the University of Canterbury in which PTs were created in Structured Query Language (SQL) and were used to model students, select problems, and provide customised feedback in the experimental version of SQL-Tutor, an Intelligent Tutoring System. The control group used the original version of SQL-Tutor where pedagogical (problem selection and feedback) and modelling decisions were based on constraints. Prior research, real-life examples, and preliminary results show that such a construct could exist; furthermore, it could be used to help students attain high levels of expertise within a domain. Students using the template based ITS showed high levels of learning within short periods of time. The author suggests further evaluation studies to investigate the extent and detail of its effect on learning.

1. Introduction

Although learning is of great importance in our society, our education systems are fraught with problems. Some researchers even claim that there is a crisis in the education field and that education as a whole is on a decline [1]. Irrespective of their position on the state or trend of education today, researchers seem to agree on how little we know about the complex processes and interaction between teaching techniques, the content, the learning process, and the learner [2]. High student-to-teacher ratios, inadequate resources, and education models that view the student and content as static entities (forcing each student to trace exactly the same path through the curriculum) lead to less than optimal learning environments. In an attempt to tailor the course work to a large group of students, the teacher is compelled to aim the course content and set the speed of progress at the ‘average’ student level, thereby providing the ideal level for only a small percentage of students. With high student numbers, pupil interaction is low. The teaching process turns into one of knowledge transfer, where declarative knowledge is transferred from the teacher to the passive

student. The role of practice with adequate feedback is denigrated. Deep learning however, requires the student to play an active role, involving high participation and interaction between the student and the teacher. Practice is an essential part of skill acquisition [3].

Psychological studies show that students could learn and retain more information from discovery than from direct instruction [4]. However, it has also been noted that experts could benefit from learning through exploration, whereas novices could get overwhelmed and lose direction with unguided exploration [5]. This implies that learning should be continually monitored and customised to the knowledge level and expertise of the student; providing more guidance, direction, and feedback for the novice and allowing greater freedom for exploration and choice as expertise increases. This level of monitoring and customising is only possible in either one-to-one human tutoring or with Intelligent Tutoring Systems (ITSs), the latter being more practical.

1.1 Motivation for the Research

ITSs are computer-based, interactive, adaptive, learning environments that allow students to practice domain-specific skills whilst receiving explicit feedback to their solutions. Each student's knowledge of various portions of the domain is recorded in a student model. From this, a knowledge level is continually calculated and updated. Both the model and the knowledge level are used to customise and create each student's path through the course material. ITSs shift the focus from knowledge transfer to knowledge communication [5]; the student is no longer a passive recipient of knowledge. Instead, they are actively involved in the learning process that has been adapted to their current knowledge and capabilities. ITSs provide a good solution to the problems mentioned in the previous paragraphs.

Although ITSs have been used successfully in a number of domains achieving high learning rates within short periods [6-8], the learning rates are still not as high as one-to-one human tutoring [9]. A main goal of ITS development is to increase the effectiveness of learning within the ITS. It is this goal that provides the impetus for this research. Two main methods of achieving this goal exist. First, modules within the ITS can be enhanced to increase effectiveness. Second, conjectures regarding various aspects of learning can be made and evaluated. This research attempts both, with greater emphasis on the latter.

In line with this goal, this research consists of two main parts. In the first part, we propose the idea of *Problem Templates* (PTs) and examine the validity of such a notion by researching theories of expertise and memory [see Section 2]. PTs are high-level mental constructs that we believe experts use to hold large amounts of data that are retrievable as a single chunk. In the second part, we explore if physical representations of PTs can be created and implemented within an ITS, if pedagogical decisions within an ITS can be based on them, and if students' knowledge can be modelled using them [see Section 3]. An evaluation study was conducted to investigate the effect of PTs on learning; a brief description is given in Section 4. This is followed by the conclusions drawn from this research and future work in Section 5. This paper covers part of the research conducted for a BSc Honours Project.

2. Background

Experts are able to memorise and handle a significantly greater amount of domain information than non-experts [10]. This phenomenon, first reported by De Groot [11], has been termed the *expertise effect in memory recall*, and has since been observed in a number of domains [12]. Till recently the *chunking theory* [13] was used to explain this effect. As an extension to the chunking theory, the template theory of memory (TT) [14] suggests that higher level chunks called *templates* are stored in Long-Term Memory (LTM) with *pointers* to them in Short-Term Memory (STM). Templates contain slots that have either information or pointers to other templates stored in them. This recursive nature of templates allows the expert to categorise templates hierarchically as a tree structure, enabling the quick retrieval of vast amounts of information from a single pointer held in STM.

Other theories exist and a comparison is given in [15]. Some common themes emerge from these theories that are relevant to this research. First, domain knowledge, not cognitive ability, defines expertise. Second, patterns and schemas, where meaningfully related items are grouped together, are used to organise domain information. Third, experts store information regarding problem states in recursively contained chunks (or templates); the chunk size being proportional to expertise. The information contained in these templates allows the expert to recognise the problem state, instantly recall solution strategies, and enable prediction and planning. Fourth, although learning domain knowledge takes time, encoding knowledge into the template slots occurs rapidly. Experts build templates with practice over time. Finally, templates can be created deliberately or implicitly.

2.1 Introducing Problem Templates

Problem templates are an extension to memory templates proposed in TT. These mental constructs are created and organised by experts and contain within their slots, information to *recognise particular problem states* and *common strategies to solve* the problem (i.e. take the user from the current state to an intended solution state). In addition, they could also contain a list of *tools* or related information (or pointers to related templates) required to implement the chosen solution strategy. PTs are not only domain-specific, but are also generally specific to problem types. Because of PTs, experts have access to large amounts of specific information regarding the problem state and possible solution strategies, thereby displaying the expertise trait of seemingly effortlessly implementing solutions. The following paragraphs discuss certain traits of experts as described in research, and use them to justify the validity of PTs.

A review of research into expertise traits in [16] highlights common themes that support the existence of PTs. First, expertise is domain-specific. PTs are also domain-specific. In fact, they are specific to problem types. Second, experts use heuristics to solve problems. As expertise increases, experts use less deduction based thinking and more pattern-matching strategies. PTs utilise pattern matching techniques to recognise the problem state and employ common solution strategies. Third, experts are able to

quickly retrieve large amounts of domain-specific information. They are also able to alter strategies quickly to account for varying or exceptional circumstances. The idea of PTs allows a single pointer held in memory to retrieve the whole hierarchy of associated PTs from LTM. As the expert is dealing with lightweight pointers instead of data, it is very easy to alter their strategy for varying circumstances, by retrieving another pointer (i.e. traversing another branch of the tree). Fourth, experts use a *divide and conquer* approach to break complex problems into smaller ones and apply solution strategies for each sub-problem. With PTs, related templates are linked together by pointers. Each template carries one or more strategies to solve that problem state (or sub-problem). The process of retrieving and implementing these strategies to solve the overall problem automatically uses the divide and conquer approach.

Experts are able to proceed by forward reasoning whereas novices reason backwards from a goal [3]. This enables experts to anticipate future moves or problem states and plan ahead. We believe that PTs explain this phenomenon. As described earlier, the expert has a hierarchically organised, tree-like structure containing PTs. The inner nodes are generally lightweight containing pointers to other nodes, while the outer leaf nodes contain data. Given this tree, the expert is able to progress by strategically planning ahead. The novice however, has to envision the final goal and trace a path back to their current problem state. Because novices are primarily dealing with data rather than pointers, their cognitive load is also very high, increasing the time and effort required to solve the problem. This also restricts the novice's ability to plan.

In certain domains such as driving and aviation, the proportion of unnecessary, high-risk behaviour that causes safety concerns is inversely proportional to experience. PTs go a long way to explain this behaviour. As described in the previous paragraph, experts are able to plan ahead while novices are unable to predict future problem states that could result as a consequence of their current actions. This inability to envision or predict future consequences elicits high-risk behaviour. Programmes such as driver education are created to increase experience and add expert wisdom by teaching consequences; or in our view, impart template information in the form of coached, expert training.

We believe that examples of experts using PTs can be found in many other domains. All these PTs allow the user to *recognise the problem state, employ common solution strategies*, and optionally contain lists of *tools* or *associated* information (or pointers to related PTs). For instance, in practical driving instruction, students are taught PTs (hill-starts, three-point turns, parallel parking, uncontrolled intersections, etc.) instead of precise technical details such as velocity, torque, engine RPM, gear ratios, etc. Grandmasters in chess use PTs to recognise problem states, predict many moves ahead, plan, and implement relevant strategies. Similarly, golf experts recognise problem states (e.g. the sand bunker, green, fairway, etc), select appropriate tools (e.g. sand wedge, putter, driver - irons/woods), and implement correct strategies (e.g. wedge shot, putt, drive). Design patterns have transformed the way students are taught in the field of Object-Oriented design. These are common strategies compiled by experts over many years that are applicable to common problem states. In team sports, collaborative PTs such as game-plays (pattern plays) are created with the purpose of quickly recognising and solving problem states collaboratively.

3. Creating and Implementing Templates in an ITS

We extended SQL-Tutor, a constraint-based tutor that teaches database querying, to include PTs. For more information on SQL-Tutor and Constraint-based modelling, please refer to [8, 17]. SQL-Tutor contains 278 problems with associated ideal solutions, written by experts since 1996. To create PTs in Structured Query Language (SQL), we first analysed and categorised these solutions using features such as level of nesting, clauses used, number of relations and associated attributes, types of attributes, aggregate functions, etc. This gave us 38 groups containing semantically equivalent solutions. In the next step, all relation and attribute names were replaced by generic variables (e.g. <rel1> to represent a relation) to form generic statements (one for each group). These statements now represented common solution strategies labelled as ideal solutions by experts for all problems in SQL-Tutor. To complete the formation of templates, the statements were sequentially numbered (template ID), and a feedback message attached to each. This feedback message was shown to the student on violation of the template. Two examples of templates (PT 1 and PT 3) are shown in Table 1. The templates consist of the template ID, the list of problems belonging to the template, the generic statement representing the SQL goal statement, and the feedback message.

Table 1: Representation of SQL PT1 and PT3

(1 (1 26 59 132 135 164 168 151 199 235) ("SELECT * FROM table" "Retrieve all attributes of one table"))
(3 (152 237 255 260) ("SELECT DISTINCT attribute(s) FROM table" "You want the details without duplicates (DISTINCT) of the attribute(s) of a table"))

The 38 templates were then categorised into eight template groups; the reasons for this are as follows. The original version of SQL-Tutor was used as the control version in the evaluation study. In that version, prior to selecting a problem, students must choose an SQL clause from which they would want a problem. At the equivalent point in the experimental version (the template based version), students would select from templates. However, this meant that the number of items presented (6 versus 38) would not only be dissimilar in the amount of choice, but also in what can be contained in STM at one time for adequate processing (approximately 7 ± 2 chunks). Template groups provided a solution to these problems.

Each problem in SQL-Tutor is assigned a difficulty level by the problem author, and ranges from one to nine; nine being the most difficult. Problem difficulty levels were compared within each template group. The hypothesis was that since templates signified a particular type of solution, they should also have the same difficulty levels; this was confirmed. The average template group difficulty level was used for certain pedagogical decisions.

In the control version, the student models are represented as overlays on top of constraints. A similar method was used for the experimental version, but with

templates instead of constraints. Constraint histories were still recorded for both versions for comparative analysis, although they were used in pedagogical decisions only for the control group.

Problem selection was done using the difficulty levels described above, the student's knowledge scores, and the student's knowledge level. The student's knowledge score and level reflect their mastery of particular concepts and the overall domain respectively. The knowledge scores and levels are continually updated as the student progresses through the content.

SQL-Tutor provides up to six levels of feedback (simple feedback, error flag, hint, partial solution, list all errors, and full solution) on submission of the student's solution. In addition to these messages, the experimental group also received template-specific feedback messages on feedback levels greater than error flag. The feedback messages also contained the generic template statement in text form.

4. Evaluation Study

An evaluation study was conducted at the University of Canterbury in May 2006, with volunteers from an undergraduate database course. Of the 68 students in the course, 48 logged into SQL-Tutor and completed a pre-test. Equally comparable pre and post-tests were administered online; the pre-test on first login, and the post-test on logins after a specified date (5th June 2006) nearing the end of the study. Students were randomly allocated to the control group (23) or the experimental group (25). They were able to use SQL-Tutor at any time. Student models comprising constraint histories and logs containing all student actions and ITS decisions were recorded.

Both the pre and post-tests contained four problems, each worth one mark. For the pre-test, the control group scored a mean of 0.5 ($sd = 0.07$) while the experimental group scored a mean of 0.42 ($sd = 0.06$). There was no significant difference between the pre-test results ensuring that both groups belonged to the same population and were comparable. Unfortunately, only 16 students completed the post-test, out of which six did not attempt any problems between the pre and post-tests. Their results were excluded from the analysis. Analysis of the remaining ten students is presented in Table 2. There was no significant difference between the groups. However, the number of students in the matched condition is too low for adequate comparisons.

Table 2: Analysis of matched pre and post-tests

<i>Group</i>	<i>No. of participants</i>	<i>Pre-test</i>	<i>Post-test</i>
Experimental	6	38% (0.25)	62.5% (0.2)
Control	4	42% (0.30)	62.5% (0.12)

Learning curves were plotted for both groups [see Figure 1]. Learning curves show whether the concepts taught are being learned; if they are, a power curve should result. For more information on learning curves see [18, 19]. The power curve equation for the control group was $y=0.0995x^{-0.384}$, and for the experimental group it was $y=0.1044x^{-0.4079}$. Both graphs have a good fit to the power curve (0.92 for the

control, 0.89 for the experimental). This indicates that not only did both groups learn domain-specific knowledge at a relatively high rate, but that they did it equally well.

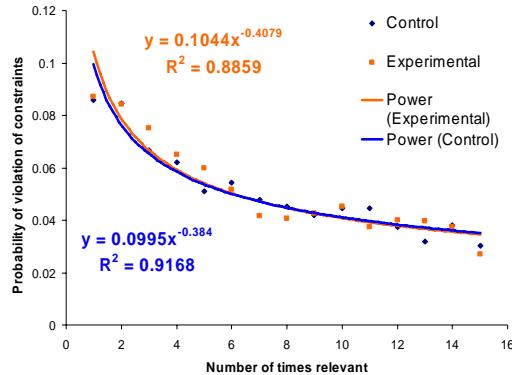


Figure 1: Learning curves for both groups

Some restrictions to this research exist. First, the time of evaluation was relatively short due to time constraints. Generally, under non-ITS instruction (e.g. sports), students take much longer periods of time to compile (create and organise) PTs. Compiling templates however, takes them from being intermediate users to mastery. Second, PTs taught by human tutors are generally compiled by combining the knowledge of many experts and refined over a period of years. Third, although PTs can be learned implicitly and automatically, they are generally taught explicitly. In this study, students did not undergo formal teaching of PTs. In spite of these restrictions, the experimental group learned domain-specific knowledge at an equally high rate, and attempted and solved the same number of problems in shorter periods than their counterparts in the control group.

5. Conclusion and Future Work

This research introduced the idea of problem templates as mental constructs that experts use to quickly retrieve large amounts of domain-specific information and solve problems seemingly effortlessly. The main goals of the research were to explore the validity of such a construct, investigate if physical representations of PTs could be created and implemented in an ITS, explore if student modelling and pedagogical decisions (such as problem selection and feedback) can be made based on them, and examine their effects on learning within the ITS. A review of prior research of memory and expertise, and exploration into examples of expertise traits in other domains provides justification for the existence of such a construct. Furthermore, physical representations of templates were created and implemented in SQL-Tutor. Pedagogical decisions and student modelling were also carried out using PTs. Results from an evaluation study show that high levels of learning can be achieved using PTs in ITSs.

Proposed future work in this area include an evaluation study conducted over a longer period of time to collate sufficient data, exploring the ability of template creation (including their implementation and evaluation) in other domains, and conducting scientific studies to comprehend the processes used by human tutors and students when using PTs.

References

1. Bennett, F., Computers as Tutors: Solving the Crisis in Education. 1999. Faben Inc.
2. Prawat, R.S., Teachers' Beliefs about Teaching and Learning: A Constructivist Perspective. *American Journal of Education*, 1992. 100(3): p. 354-395.
3. Ericsson, K.A. and N. Charness, Expert Performance; Its Structure and Acquisition. *American Psychologist*, 1994. 49(8): p. 725-747.
4. Anderson, J.R., Rules of the Mind. 1993, Hillsdale, NJ: Lawrence Erlbaum Associates.
5. Mitrović, A., SQL-Tutor: A Preliminary Report. 1997, Computer Science & Software Engineering Department, University of Canterbury: Christchurch. p. 1-11.
6. Corbett, A. Cognitive Computer Tutors: Solving the Two- Sigma Problem. in User Modeling 2001: 8th International Conference, UM 2001, LNAI 2109. 2001. Sonthofen, Germany: Springer-Verlag Berlin Heidelberg.
7. VanLehn, K., et al., The Andes Physics Tutoring System: Five Years of Evaluations, in Artificial Intelligence in Education - Supporting Learning Through Intelligent and Socially Informed Technology, C.-K. Looi, et al., Editors. 2005, IOS Press: Washington, D.C. p. 678-685.
8. Mitrović, A. and S. Ohlsson, Evaluation of a Constraint-Based Tutor for a Database Language. *International Journal of Artificial Intelligence in Education*, 1999. 10: p. 238-256.
9. Bloom, B.S., The 2 Sigma Problem: The Search for Method of Group Instruction as Effective as One-to-One Tutoring. *Educational Researcher*, 1984. 13(6): p. 4-16.
10. Gobet, F. and A.J. Waters, The Role of Constraints in Expert Memory. *Journal of Experimental Psychology*, 2003. 29(6): p. 1082-1094.
11. de Groot, A.D., Thought and Choice in Chess. 1978, The Hague, Netherlands: Mouton Publishers.
12. Gobet, F. and G. Clarkson, Chunks in Expert Memory: Evidence for the Magical Number Four ... Or is it Two? *Memory*, 2004. 12(6): p. 732-747.
13. Chase, W.G. and H.A. Simon, Perception in Chess. *Cognitive Psychology*, 1973. 4: p. 55-81.
14. Gobet, F. and H.A. Simon, Templates in Chess Memory: A Mechanism for Recalling Several Boards. *Cognitive Psychology*, 1996. 31: p. 1-40.
15. Gobet, F., Expert Memory: A Comparison of Four Theories. *Cognition*, 1998. 66(2): p. 115-152.
16. Shanteau, J., The Psychology of Experts: An Alternative View, in Expertise and Decision Support, G. Wright and F. Bolger, Editors. 1992, Plenum Press: NY. p. 11-23.
17. Mitrović, A., An Intelligent SQL Tutor on the Web. *International Journal of Artificial Intelligence in Education*, 2003. 13(2-4): p. 173-197.
18. Martin, B., et al. On Using Learning Curves to Evaluate ITS. in The 12th international conference on Artificial Intelligence in Education. 2005. Amsterdam.
19. Martin, B. and A. Mitrović. Evaluating Intelligent Education Systems with Learning Curves. in The Workshop on Evaluation at Adaptive Hypermedia 2004. 2004. Eindhoven, The Netherlands.

Turning the page of an electronic book

Veronica Liesaputra

Department of Computer Science, University of Waikato, Hamilton, New Zealand
vl6@cs.waikato.ac.nz

Abstract. Page turning is an important, yet invisible, part of paper-based document navigation. However, this affordance is not easily reclaimed in a digital setting. The interaction often becomes interruptive, rather than an unselfconscious act. Providing a literal representation of page turning may help this navigation to be seamlessly integrated into the flow of readers regular activities, and people can transfer their paper-based document navigation skills to the digital world. This paper presents an overview of three page turning techniques for electronic books that are sufficiently realistic, scalable and computable in real time. A summary of the main features of each technique will be presented at the end of this paper.

1 Introduction

Crestani et al. define an electronic book as a digital book that not only captures the affordances of a physical book, but also transcends the limitations of its paper counterpart [1]. In many systems, the look and feel of an electronic book imitates the look and feel of a physical book, and the visual features of the original document are preserved by presenting the page images instead of the raw text. Readers often can annotate, bookmark, highlight and randomly access articles. Some of the functionality that the electronic text bestows such as searching, content editing, hyperlinking and zooming, are also included into the system.

Although digital technology offers many functionalities that can go beyond paper, in many systems, a reader cannot interact with an electronic book in the same way as a physical book. For example, turning a page of a book is often an anticipatory act. To ensure reading continuity between the text at the end of a page and at the beginning of the next page, readers often lift a page before it was read, and so sometimes pages are turned prematurely. In many electronic books, a reader has to click the next button to view the next page. Users briefly lose contact with the text, and it removes the opportunity to subtly look ahead the content of the page behind the current page.

The British Library’s “Turning the Pages” animation is an attempt to preserve the physicality of page turning itself [2]. Visitors can metaphorically pick up a page and turn it. To accomplish this, photographs were taken at intermediate points during each page turn, and stored in a system constructed using Macromedia Director [3]. When a visitor turns a page, the corresponding image is displayed. Although the simulation is compelling, it is very labour intensive and not scalable for handling books with large number of pages.

The 3D Book Visualiser developed at the University of Waikato generates a model of a physical book and tries to simulate how the book looks in nature [3,

4]. A spline function is used to create a curved appearance of the top pages of an open book. Page turning is animated by using a mass-spring structure. Due to the computational overhead, the system pre-computes the appearance of the page when it is turned from the top, middle or bottom point of the page's right edge through a pre-defined path. By doing so, the size of the book and the turning path cannot be varied.

The main goal of my research is to evaluate whether simulated three dimensional physical representations of documents have significant advantages over the conventional digital representations—scrolls and concertinas. In the scroll format, documents are represented as a single continuous page with vertical and horizontal scroll bars. Readers roll up the top of the page and the lower contents move up. This format is exemplified by the Teletype roll and the web page. The concertina format represents the document as a single linear sequence of pages segmented based on the paper size. Concertinas is developed in the form of fanfold papers, Adobe Acrobat Reader and Word Print Preview. Readers navigate either by using scroll bars, or by clicking the next and the previous buttons. In this project, documents will be presented as three-dimensional physical representations of a book. The appearance and handling of a simulated book will resemble the appearance and handling of a real book. Readers can turn single or multiple pages, and rotate or move the book.

The performance of each representation will be compared according to three criteria: efficiency, effectiveness and reading experience. A system is efficient if users require a small amount of effort to perform their tasks and a small amount of time to learn how to use the system. The effectiveness of a document's format is examined by looking at how many tasks are completed accurately and how many errors produced. While reading experience is determined by how pleasant and satisfying the interface is to the users.

According to the Merriam-Webster dictionary, a book is a set of paper bound together into a volume [5]. A book simulation involves simulating both the paper and the binding. During the first six months of this project, various techniques that can be used to simulate the appearance and handling of the pages have been investigated and implemented. However, in this paper we will only describe three available page turning mechanisms—cylindrical, conical and peeling—that enable users to turn a page of the electronic book in the same way as a physical book. The simulation looks sufficiently realistic so it does not deviate from the book metaphor, it is computable in real time and is scalable to handle books that have high page counts. Because of the space constraint, we will only consider the case when a user picks the bottom corner of the page and turns it from the right to left side of the book. Adjusting the algorithm for general case is very simple. We also assume that initially the page lies flat on the XY plane.

The page turning implementation presented in this paper is written in Java and JOGL (a Java based OpenGL toolkit) is used to perform the rendering. To turn a page from the bottom corner, readers need to click the bottom corner of the page and drag it to the left. When the mouse button is released before the turned page reaches the midpoint of the book, the page springs back to its original position. From the midpoint onwards, the turned page will continue turning until it reaches its final position.

2 Cylindrical model

In this model, the page turning mechanism is a combination of a deformation and a rotation process. The page is deformed by wrapping the page around an imaginary cylinder whose sides touch the XY plane along the Y axis. Different curling effects can be accomplished by changing the radius of the cylinder. The deformed page is then rotated around the book spine from one side of the book to the other. The rotation angle (ϕ) increases from 0° to 180° as the turn progresses.

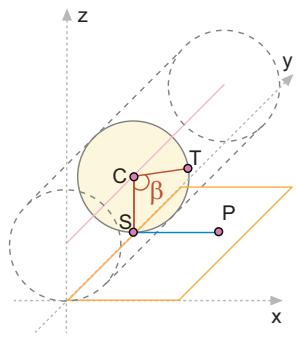


Fig. 1. Mapping a point to a cylinder

At the beginning of a page turn, the radius of the cylinder is very large and the page lies flat on the plane. We then gradually reduce r until the midpoint is reached (or $\phi = 90^\circ$). This causes the page to curl around the cylinder and the right edge of the page is lifted up from the plane. Finally, r increases from the midpoint onwards, making the page uncurl.

Internally, the page surface is a mesh of $m \times n$ number of points. As the page is turned, we need to calculate where each point maps to on the cylinder and then rotate it by an angle of ϕ around the book spine. Given any point $\mathbf{P} = (P_x, P_y, 0)$ on the page, we want to map the arc SP to the cylinder, where $\mathbf{S} = (0, P_y, 0)$. If point $\mathbf{T} = (T_x, T_y, T_z)$ is where \mathbf{P} maps to on the cylinder, the circular cross section of the cylinder that is parallel to the base of the cylinder and goes through \mathbf{T} will coincide with the Y axis at \mathbf{S} , as shown in Figure 1. This means that length of the line SP is equal to the length of the arc ST . If the angle between point \mathbf{S} , the centre of the cross section and point \mathbf{T} is β , point \mathbf{T} can be calculated by rotating \mathbf{S} around the line ($X = 0, Z = r$) with an angle of β .

The values of r and β are the same for each point in a column, and so we are only required to calculate where each point on the first row of the mesh maps to on the cylinder and change the value of T_y as we go down the row. The deformation computation is in order of $O(n)$.

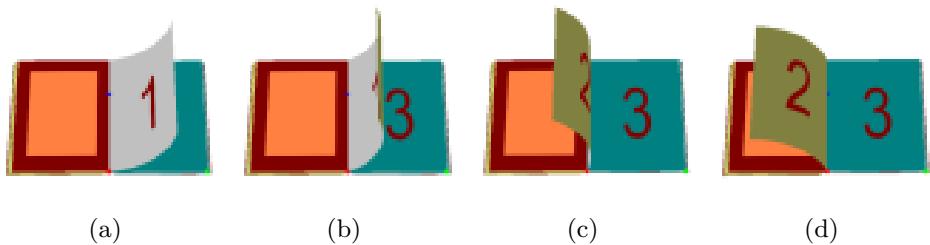


Fig. 2. Page rotated at (a) 45 (b) 75 (c) 105 and (d) 135 degrees

Figure 2 shows the appearance of the turned page. In this model, the right edge of the page is lifted ahead of the plane, making the page appear rather rigid. This model is suitable to define how the page looks when the turning path is parallel to the X axis. It has been used by 3D book visualiser to simulate the appearance of the page when a wedge of pages is being turned [4].

3 Conical model

Readers seldom turn a page with a path that is parallel to the X axis. They generally lift the corner of the page and turn it diagonally upward until it reaches the page midpoint, then diagonally downward from the midpoint onwards. Thus, only the grabbed corner should be lifted ahead of the mesh. To achieve this, Card et al. use a conical model instead of a cylindrical model [6]. The page turning mechanism of this model is similar to the cylindrical model except that it uses a cone instead of a cylinder. The shape of the cone is defined by its angle θ and its apex location $A = (0, A_y, 0)$. Different page curves can be obtained by varying these values.

Initially, $\theta = 90^\circ$ and the turned page appears flat on the plane. As the page is turned, we gradually reduce the value of θ and move the cone apex down the Y axis. The lower right corner of the page will be lifted up ahead of the rest of the page. Once the rotation angle ϕ is greater than 90° , the values of θ and A_y are gradually increased. The page curve flattens as ϕ heads toward 180° .

Due to the nature of the cone, for a point $P = (P_x, P_y, 0)$ on the mesh, we want to map the arc SP . $S = (0, S_y, 0)$ is the point of intersection between the Y axis and the circle whose radius is equal to the length of the line AP , as shown in Figure 3. Let us denote $T = (T_x, T_y, T_z)$ as the point where P maps to on the cone. Similar to the cylindrical model, a circular cross section that is parallel to the base of the cone and goes through point T will also go through S , and so the length of the arc SP will also be equal to the length of the arc ST . Given the angle β between point T , the centre of the cross section and point S , T can be obtained by first rotating S around the line ($X = 0, Z = r$) with an angle of β to $S' = (S'_x, S'_y, S'_z)$, followed by a rotation around the line ($Y = S_y, Z = 0$) by an angle of θ . The radius of the circle (r) at each cross section is varied, and so the deformed page is defined by computing where each point of the mesh maps onto the cone. The complexity of the algorithm is in the order of $O(n^2)$.

Although the curling effect of the turned page (as shown in Figure 2 and Figure 4) looks sufficiently realistic, in both the cylindrical and conical models, readers do not have any control on how the page is turned. Users are only able to specify

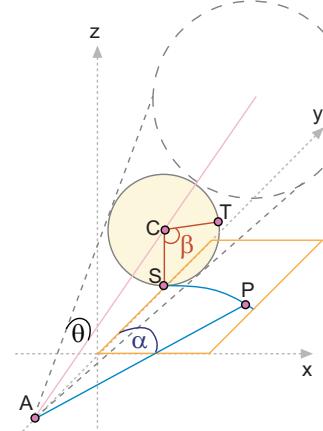


Fig. 3. Wrapping a page around a cone

which side of the book the page should be turned to. A reader cannot state how high the corner of the page should be lifted up from the plane or where the corner of the page should move to.

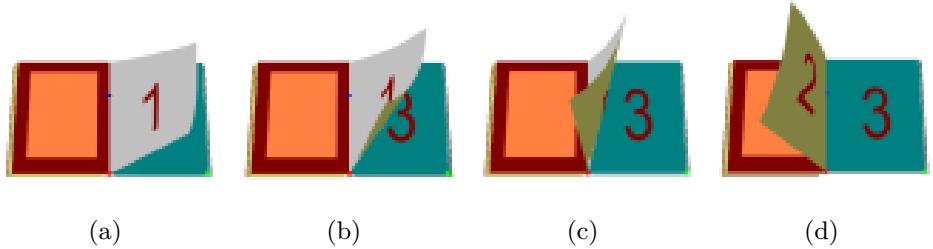


Fig. 4. Page rotated at (a) 45 (b) 75 (c) 105 and (d) 135 degrees

4 Peeling

As shown in Figure 5, this technique computes in real time how the page would look when turned if a page was creased flat in two dimensions, and adds visual details to simulate a three-dimensional bend instead of a sharp crease. This method was first proposed by Beaudouin-Lafon to handle overlapping windows [7], and it is now widely used to simulate a page turning effect of a two-dimensional book constructed using Macromedia Flash [8]. Unlike the cylindrical and conical model, readers have the flexibility to define path of the page as it turns. By implementing it in a three-dimensional environment, a user also has the ability to state how high the corner should be lifted upwards.

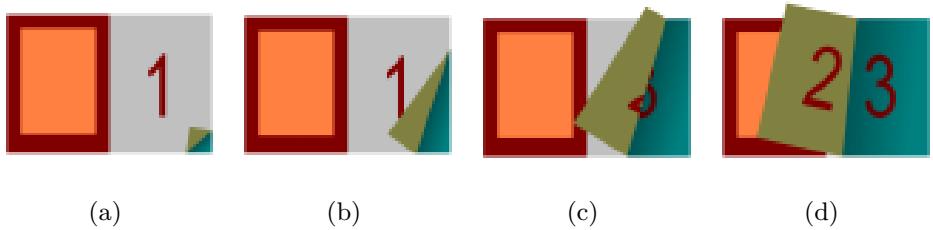


Fig. 5. Page turning using the peeling effect

This technique involves partitioning the page into three sections: the visible part of the page being turned, the part of the other side of the page that the turn has made visible and the part of the page underneath that has been revealed. These regions are shaded differently in Figure 6. The initial position of the page is the

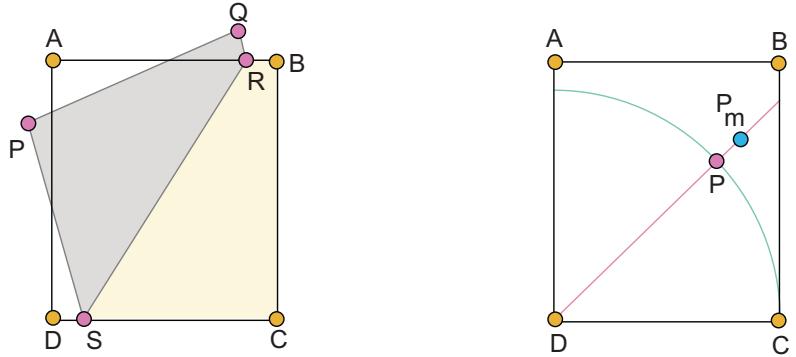


Fig. 6. The position of the creased page

Fig. 7. Mapped position of the mouse

rectangle $ABCD$, and the act of turning has moved the lower right corner C to position P . The region made visible by the turn is the creased polygon $PQRS$. While the area revealed, defined by the polygon $SRBC$, is a reflection of the creased polygon. The location of points Q , R and S are calculated based on the location of point P .

The page-turning point P cannot be moved freely: the paper imposes a physical constraint on where it can be, for the distance PS cannot exceed the length of CD . This constrains P to lie within the circle shown in Figure 7. This means that the reader cannot move the lower left corner of the page outside the circle shown in Figure 7, because that would cause the page to be torn from the book. In our implementation, if the mouse moves outside the circle (say to position P_m) it is mapped to a point P on the arc of the circle.

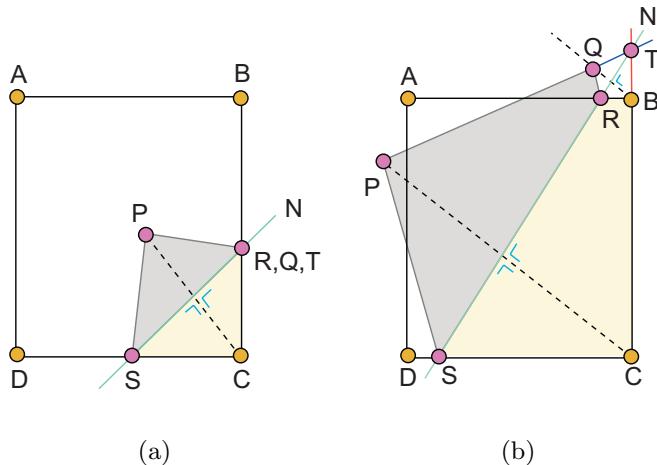


Fig. 8. The appearance of the page: (a) when R intersects BC (b) when R intersects AB .

To calculate the creased section of the page, first we compute the crease line N that is the perpendicular bisector of the line PC . Points S and T are the intersection points of the crease line with the bottom and the right edges of the paper. If T is located below the point B , as shown in Figure 8(a), the position of points Q and R is equal to T , and the creased polygon is defined by the triangle PRS . Figure 8(b) showed the appearance of the creased section when T is located above B . Point R is where the line N intersects the top edge AB . Point Q is obtained by drawing a line from corner B parallel to CP , and noting where it intersects the line PT —effectively ensuring that the crease line is a perpendicular bisector of line QB as well as of PC . The creased section is the quadrilateral $PQRS$.

Although this peeling effect is designed to be used in two dimensional environment, the bending appearance of the page can be simulated by adding a Z value to each of the point, displayed in Figure 9, and by utilizing a simple spline function to make the bend look curvaceous instead of like a sharp crease. Like a page, the creased area is now smaller than the polygon $SRBC$ to compensate for the translation in the Z direction. Without the spline function, the complexity of this technique is constant. With the spline function it is in the order of $O(n^2)$.

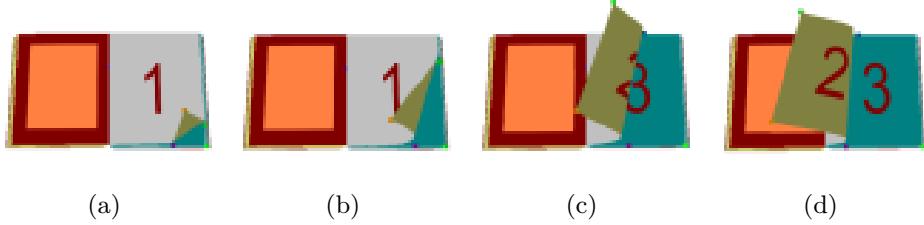


Fig. 9. A page being turned to various points

5 Conclusion

Page turning is a common and important navigation feature of a physical book. In many electronic book systems this interaction is simplified by clicking a navigation button. Though clicking a button is efficient, readers briefly lose contact with the text and the navigation becomes interruptive. Recently, many research projects propose different page turning mechanisms with various degrees of realism and interactivity in an attempt to preserve the physicality of the page turning. In this paper we have presented and implemented three page turning techniques—cylindrical, conical, and peeling—that are sufficiently realistic, scalable and computable in real time. The implemented page turner is still in prototype form and requires more work to make the simulation appear more realistic and intuitive for the users. The current form is sufficient for highlighting the main features of each technique.

The turned page wrapped in a cylinder looks rigid, and it is suitable to simulate the appearance of a page when it is turned parallel to the X axis. In three-dimensional environment, this model uses the least computational resources. The conical model is suitable to simulate how the page looks when it is turned diagonally. By combining these models, we can achieve a more realistic page turning where a reader can either turn the page horizontally or diagonally. Because the turning path in cylindrical and conical model is pre-defined, the peeling technique is the most interactive model. Users can state the turning path of the page and how high the page is being lifted upwards. Its computational resources are dependent on the spline function used to create the curvaceous appearance of the page. Without the use of the spline function, it uses less computational resources than the cylindrical model. This peeling technique can be used to simulate the page turning for both two and three dimensional digital books. Using a 9×9 page mesh, the page turning speed for each model is about 60 frames per second. For a comparison, PAL (a color encoding system used by New Zealand's broadcast television systems) displays pictures at a rate of 24 frames per second [9].

Even though we have achieved an adequate mechanism to allow people to turn the page of electronic books in the same way as paper books, it is still a question whether they gain significant advantages by turning the pages in this manner over the conventional way—scrolling or clicking navigational buttons. In the future we plan to conduct some usability studies to evaluate the effectiveness of the page turning navigation over the conventional navigation.

Acknowledgements

We gratefully acknowledge funding from Google.

References

1. Crestani, F., Landoni, M., Melucci, M.: Appearance and functionality of electronic books. *International Journal on Digital Libraries* (2005) 192–209
2. Library, B.: Turning the pages, <http://www.bl.uk/onlinegallery/ttp/ttpbooks.html> (2006)
3. Chu, Y.C., Witten, I.H., Lobb, R., Bainbridge, D.: How to turn the page. In Proc. of the 3rd ACM/IEEE-CS joint conference on Digital Libraries (2003) 186–188
4. Chu, Y.C., Bainbridge, D., Jones, M., Witten, I.H.: Realistic books: A bizarre homage to an obsolete medium? In Proc. of the 4th ACM/IEEE-CS joint conference on Digital Libraries (2004) 78–86
5. Dictionary, M.W.O.: Browse, <http://www.m-w.com/dictionary/browsing> (2006)
6. Card, S.K., Hong, L., Chen, J.D.: Turning pages of 3d electronic books. 2006 IEEE Symposium on 3D User Interfaces (2006) 159–165
7. Beaudouin-Lafon, M.: Novel interaction techniques for overlapping windows. In Proc. of the 14th ACM symposium on User interface software and technology (2001) 153–154
8. Bhangal, S.: The page turn effect in flash mx, <http://www.oreillynet.com/pub/a/javascript/2004/09/03/flashhacks.html> (2004)
9. Union, I.T.: Recommendation itu-r bt.470-6: Conventional television systems (1998)

Towards a General Model of Supporting Explanations for Constraint-based Tutoring Systems

Amali Weerasinghe, Antonija Mitrovic and Brent Martin

Intelligent Computer Tutoring Group
University of Canterbury New Zealand
{acw51, tanja, brent}@cosc.canterbury.ac.nz

Abstract. We present a project with the goal of developing a general model of explanation support, which could be used in both well- and ill-defined instructional tasks. We have previously studied how students interacted with an existing intelligent tutoring system teaching database design, while getting additional help from a human tutor through a chat interface. Analysis of interactions provided by the human tutors indicates that they have helped the students to improve their understanding of database design. We then developed an explanation model, which we present in this paper.

1 Introduction

Intelligent tutoring systems (ITS) are computer-based educational programs that help students to learn by providing adaptive pedagogical assistance. Several empirical studies have shown significant learning gains when interacting with an ITS [4,6]. However, some studies indicate that students acquire shallow knowledge even in the most effective ITSs [1]. As a result students gain sufficient knowledge to obtain a passing grade on tests, but have difficulty in transferring knowledge into novel situations. Therefore cognitive scientists and educationists have been interested in finding methods that can be used to overcome the shallow learning problem. Self-explanation is described as an “*activity of explaining to one-self in an attempt to make sense of new information, either presented in a text or in some other medium*”[2], and has been shown to facilitate the acquisition of deep knowledge by students.

The long-term goal of this PhD project is to develop a general model of explanation which will provide adaptive support to learners across domains. The main objective of this model is to assist students to develop their self-explanation skills while being prompted to explain their mistakes to the tutor. Since we previously implemented explanation support for the database design tutor [8], the initial work on this project started with the same tutor. As the first step, we conducted an observational study [7] focusing on how students interacted with EER-Tutor [6], while getting additional help from a human tutor through a chat interface. From the results of this study, we developed a prototype of the explanation model, which we present in this paper. This model addresses three basic decisions: when to prompt for explanations, what to explain and how to obtain explanations from learners.

The background section reviews Intelligent Tutoring Systems, and Constraint-based Modelling (CBM), a popular student modelling technique used to develop ITSs. The explanation model is presented in Section 3 followed by directions for future work.

2. Background

We discuss the different components of a typical ITS in the following section. The remaining sections focus on constraint-based modeling and self-explanation, as they are important aspects of this research.

2.1 Intelligent tutoring systems

The architecture of a typical ITS consists of four main components: a *domain module*, a *student modeller*, a *pedagogical module* and an *interface*. The student interacts with the ITS through the interface. Depending on the implementation, the system waits for a student's request to evaluate his/her solution or provides immediate feedback after tracing a student's behaviour. Evaluating a solution may result in a number of actions within the system. The student modeller compares the student's solution against the system's solution using the domain module. The student modeller then updates the student model to reflect the student's new knowledge. The pedagogical module then provides feedback or selects a new problem based on the student's performance.

2.2 Constraint-based modelling

Constraint based modelling introduced by Ohlsson, is based on his theory of learning from performance errors [6]. The key assumption in CBM is that the diagnostic information is in the problem state at which the student arrives and not in the sequence of his/her actions. This assumption is supported by the fact that a correct solution cannot exist for a problem that traverses a problem state violating fundamental ideas or concepts of the domain.

The unit of knowledge in CBM is called a state constraint. Each constraint is an ordered pair $\langle C_r, C_s \rangle$, where C_r is the relevance condition and C_s is the satisfaction condition. The first specifies when this piece of declarative knowledge is relevant, and the second describes the state whereby the piece of knowledge has been correctly applied. In other words,

IF \langle relevance condition \rangle is true
THEN \langle satisfaction condition \rangle will also be true

Initially, the student solution is matched against C_r . The satisfaction condition is evaluated only if the student solution satisfies C_r ; otherwise the constraint is ignored. The constraint is considered satisfied if the student solution satisfies the C_s , else the constraint is violated.

CBM is computationally simple because student modelling is reduced to pattern matching [6]. During the evaluation of a problem state, all relevance patterns are matched against the problem state. In a case where the problem state matches the relevance pattern, it is then checked against the satisfaction condition. If the satisfaction condition is not met, then the constraint is violated, which indicates an error. Constraint based tutors have been developed for several domains such as database querying, database modeling and data normalization [6].

2.3 Tutoring systems facilitating SE

Self-explanation has been facilitated in several ITSs to enhance learning. These systems are discussed in this section.

Self-explanation with SE-Coach. SE-Coach supports self-explanation by prompting students to explain solved examples [3]. It is implemented within ANDES, a tutoring system that teaches Newtonian Physics. The first level of scaffolding in the SE-Coach's interface is provided by a masking mechanism that covers different parts of the example with grey boxes, each corresponding to a unit of information. When the student moves the mouse over a box, it disappears, revealing the content underneath. The second level of scaffolding produces specific prompts to self-explain. Students are prompted to self-explain only when the tutor decides it is beneficial. To determine when to intervene, SE-Coach relies on a probabilistic student model, that monitors how well the student understands the domain by capturing both implicit self-explanations and self-explanations generated through the interface. The results of the empirical evaluation of SE-Coach reveal that the structured scaffolding of self-explanation can be more beneficial in early learning stages.

PACT Geometry Tutor. Aleven and Koedinger investigated self-explanation in the PACT Geometry Tutor [1]. The students in the experimental group were expected to provide correct explanations for solution steps by citing definitions and theorems used. A glossary of knowledge in the form of definitions and theorems was provided in order to help students to explain the solution steps. The study revealed that explaining reasoning steps results in improved problem solving skills.

NORMIT. NORMIT [6] is designed for university level students and provides a problem-solving environment for data normalization. Students are expected to self-explain while solving problems. In contrast to other ITSs that support self-explanation, NORMIT requires an explanation when an action is performed for the first time. For the subsequent actions of the same type, explanation is required only if the action is performed incorrectly. Similar to other systems, NORMIT supports self-explanation by prompting the student to explain by selecting one of the offered options. An evaluation study indicated that the students' performance increased when interacting with NORMIT.

These systems use different approaches to facilitate self-explanation, depending on the domain and the target student group. However, all the models that have been used

are domain specific. The goal of our research is to develop a general model of explanation that can be used across domains which facilitate self-explanation.

3. Prototype of the Explanation Model

The explanation model will be used to decide when to prompt for explanations, what to explain and how to obtain explanations from learners. The model consists of three parts: error hierarchy, tutorial dialogues and rules for adapting them. Each component is now described in turn. Error hierarchy and the dialogues are used to determine timing and content of the explanations respectively. Learners will be able to provide explanations by selecting the correct one from a list provided by the tutor.

3.1 Error Hierarchy

The domain model of constraint-based tutors is represented as a set of constraints [6]. Violations of constraints indicate mistakes in students' solutions. In previous work, we developed a hierarchy of students' errors in the Entity-Relationship (ER) domain [8], which categorizes errors as being syntactic or semantic in nature. A high-level view of the hierarchy is given in Figure 1, with nodes ordered from basic domain principles to more complicated ones. Violated constraints for each type of error are represented as leaves of the hierarchy. Syntax errors are simple, each requiring only one feedback message to be given to the student; for that reason, every syntactic error corresponds to a particular constraint being violated. For example, constraint 8 is violated when the student creates an attribute which does not belong to an entity/relationship type. The hierarchy for semantic errors is deeper, with error types further divided into sub-errors.

The ER error hierarchy was developed for that particular domain, so we were interested whether it can be reused in other domains. With that goal, we tried to fit the errors from a different domain, ER-to-relational mapping, into this structure. This domain involves mapping an ER schema to a relational schema using the 7-step mapping algorithm [5]. The task is well-defined, due to the deterministic algorithm used. However, both domains (ER modelling and ER-to-relational mapping) involve mapping as the major activity. Due to this similarity, we decided to explore additional domains of different nature, such as data normalization and fraction addition. Data normalization is the process of refining a relational database schema in order to ensure that all relations are of high quality.

During this investigation, we identified situations when it was not enough to present a single feedback message for some violated syntax constraints – instead, a dialogue was required. Therefore, we modified the structure of the error hierarchy to divide all error types into two main categories: *Basic Syntax Errors* and *Errors dealing with the main problem-solving activity*. Under the new node *Basic Syntax errors*, we included simple syntax errors, such as checking whether the student has filled the required fields, the components used to fill the required fields are valid etc. Hence it is sufficient to discuss such errors using a single message. The other category requires a dialogue to be conducted.

Another observation was that different clusters of syntax errors were needed in these newly examined domains, as opposed to the flat structure of the syntax error sub-hierarchy in Figure 1. For example, in data normalization, several constraints check whether the student is using valid attribute names in different steps of the algorithm, all of which can be categorized into a single node (*Check validity of attributes*), specified as a child node of *Basic Syntax Rules*.

Another refinement required was to combine the two nodes *Connecting an attribute to an incorrect construct* and *Errors dealing with cardinalities and participation*, which deal with associations between solution components, into a new

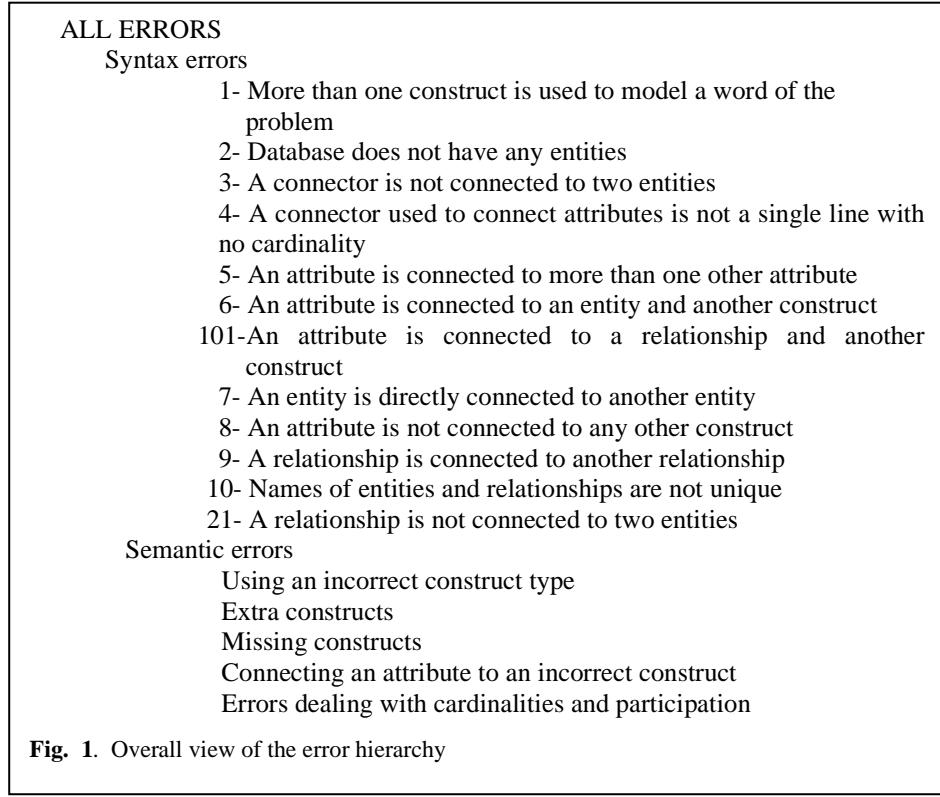


Fig. 1. Overall view of the error hierarchy

parent node *Associations*. This new node will have different domain-specific child nodes.

The last refinement is based on an observation from the previous study [7]: some students seem to be reacting to feedback on errors by making suggested changes without reflecting on other changes that need to be carried out. In ER modelling, if a regular entity with a key attribute is changed to a weak entity, then a partial key should be specified instead of the key attribute. This behaviour may lead to frustration due to the number of attempts that the student has to go through to arrive at the correct solution. A new node *Failure to complete related changes* was added to the existing error hierarchy, which reminds the student to check whether other changes

are necessary. In such cases, the student will be prompted to reflect on other related changes before submitting the solution.

The highest level of the refined error hierarchy has two nodes: *Basic syntax errors* and *Errors dealing with the main problem solving activity*. The second node is now further categorized into five nodes: (i) *Using an incorrect construct type* (ii) *Extra constructs* (iii) *Missing constructs* (iv) *Associations* and (v) *Failure to complete related changes*. The subsequent levels deal with domain-specific concepts. We also tested the refined error hierarchy in the domain of fraction addition. Even though this domain is very simple, it is quite different from the domains that we have investigated previously. All the error types in the fractions domain could be specified using the error hierarchy. The common feature in all these tasks is that the syntactic and semantic accuracy of a solution can be completely evaluated by the components of the solution and its associations. However, there are exceptions. For instance, in reading and comprehension, where learners are asked to answer questions based on a paragraph, the accuracy of an answer cannot be evaluated by checking only for the correct words according to the grammatical rules. We also need to understand the implicit semantic meaning of the sentence. Therefore, our error hierarchy is not useful in such cases. In summary, we have been able to use this hierarchy in four different types of tasks: thus we believe it would be sufficiently general to be used for different types of instructional tasks only when the solution can be completely evaluated by the components of the solution and its associations.

3.2 Tutorial Dialogues

In our model, explanation is facilitated through tutorial dialogues. For each error type (i.e. each leaf node in the hierarchy), we designed a dialogue consisting of four stages. In the first stage, the dialogue informs the student about the concept that s/he is having difficulty with. The purpose of the second stage is to assist the student in understanding why the performed action is incorrect. The third stage prompts the student to specify how to correct the mistake. In the fourth stage, the student can review the domain concept learned.

An example tutorial dialogue is given in Figure 2. Initially, the system identifies the domain concept the student has problems with, and asks the student to explain it (*EERTutor1*). A list of possible answers is provided from which the correct one needs to be selected. If the student fails to provide the correct answer (*Student1*), s/he will be asked a more specific question that provides a further opportunity to understand the fundamental principle that is violated (*EERTutor2*). However, if s/he fails to correct the mistake even after going through a series of detailed questions, as the last resort the tutor will provide an explanation on how to correct the mistake together with a brief description about the fundamental principle that needs to be learnt (*EERTutor4-5*). The dialogues use simple questions (*EERTutor1*), fill-in-a-blank (*EERTutor7*), or true-false questions, to motivate the student to explain. When a certain mistake is repeated, the model informs the student of its observations (“*You seem to be having some difficulty with regular entities.*” in *EERTutor1*), thereby providing an opportunity to reflect on his/her domain knowledge. As all dialogues facilitate explanation by pointing out errors (*EERTutor3*), students are given

opportunities to reflect on their problem solving procedure, which is another important meta-cognitive skill. Although the prompts are domain-specific, the structure of the dialogues is domain-independent. We are currently investigating the applicability of the dialogue structure to various domains.

3.3 Rules for adapting dialogues

These rules enable individualization of the dialogues. For each student, the rules decide on the entry point into the dialogue, and/or the timing of the dialogue. Currently there are eight rules and they are based on the observations from the study [7]. For example, rule 4, dealing with customizing the entry point to the dialogue, is initiated when the same error is made in the last n attempts. In that case, a dialogue corresponding to the mistake is initiated, but the dialogue starts from the problem-independent question (*EERTutor1*). If the error was made less than n attempts, then the dialogue will start from the error within the current context. Hence the dialogues are adapted to the student's knowledge and explanations.

EERTutor1: You seem to be having some difficulty with regular entities. Let's look at regular entities in detail. Can you tell me the general rule to decide whether something is a regular entity?

Student1: *I don't know*

EERTutor2: Ok! Let's try a simple thing. What is used to uniquely identify instances of a regular entity?

Student2: *The key attribute*

EERTutor3: Now tell me what is unique about *CHAPTER* regular entity?

Student3: *I'm not sure*

EERTutor4: Let me help you a bit. Check carefully whether any of the attributes of *Chapter* can uniquely identify the instances of *Chapter*?

{After sometime} Please tell me whether you managed to find any attributes.

Student4: *No*

EERTutor5: Fine. It doesn't have any attributes that can uniquely identify instances of *Chapter*. So now can you tell me why modeling *Chapter* as a regular entity is incorrect?

Student5 : *Because it doesn't have a key attribute.*

EERTutor6: Great! Now tell me the best way to model *CHAPTER*?

Student6: *A weak entity*

EERTutor7: Well Done! Now select the correct pair of words to fill in the blanks so that you'll remember what you learnt.

A is used to uniquely identify instances of a entity.

Student7: *key attribute ,regular*

EERTutor8: Excellent, Go ahead and make the changes. Submit the solution when you're done.

Fig. 2. A sample tutorial dialogue

Rule 1 (dealing with timing of dialogues) checks whether the student made any attempts at the current problem, and has been inactive for a specified period of time (such as 1.5 minutes, the time period we observed in the study [7]). This rule will initiate an evaluation of the student's solution even though it has not been submitted yet, and start a dialogue to discuss the most suitable error (depending on the error hierarchy and the student solution). Individualization of the chosen dialogue will be

based on the rule 4 discussed above. As these rules do not depend on domain specific details to individualise dialogues, the rules can be used across domains.

4. Future Work

The explanation model will be implemented in different domains to evaluate the effectiveness of the model. We have chosen ER modelling as the ill-defined task, and data normalization as the well-defined. ER modelling is ill-defined because the final result, the ER model, can be defined in abstract terms, but there is no algorithm to find it. Data normalization is well-defined due to the deterministic algorithm used. EER-Tutor and NORMIT which helps students to learn ER modelling and data normalization respectively are two existing tutoring systems developed by our research group at University of Canterbury [6]. The explanation model will be incorporated into these systems to facilitate self-explanation. The effectiveness of the enhanced systems will later be evaluated in authentic class room environments.

References

1. Aleven, V., Koedinger, K. R., Cross, K.: Tutoring Answer Explanation Fosters Learning with Understanding. In: Lajoie, S., Vivet, M.(eds.): Proc. AIED 1999, IOS Press, 1999, 199-206
2. Chi, M. T. H.: Self-explaining Expository Texts: The dual processes of generating inferences and repairing mental models. Advances in Instructional Psychology, 2000, 161-238
3. Conati, C., VanLehn, K.: Toward Computer-Based Support of Meta-Cognitive Skills: a Computational Framework to Coach Self-Explanation. Artificial Intelligence in Education, 11, 2000, 389-415
4. Koedinger, K. R., Anderson, J. R., Hadley, W. H., Mark, M. A.: Intelligent tutoring goes to school in the big city. International Journal of Artificial Intelligence in Education, 8 (1), 1997, 30-43
5. Milik, N., Marshall, M., Mitrovic, A.: Teaching Logical Database Design in ERM-Tutor. In: M. Ikeda & K. Ashley (eds.): Proc. 8th Int. Conf. on Intelligent Tutoring Systems, 2006, 707-709
6. Mitrovic, A., Suraweera, P., Martin, B., Weerasinghe, A.: DB-suite: Experiences with Three Intelligent, Web-based Database Tutors. Interactive Learning Research, 15(4), 2004, 409-432
7. Weerasinghe, A., Mitrovic, A.: Individualizing Self-Explanation Support for Ill-Defined Tasks in Constraint-based Tutors, In: Aleven, V., Ashley, K., Lynch, C. and Pinkwart, N. (eds.):Workshop on ITS for Ill-defined domains at ITS2006, 2006, 56-64
8. Weerasinghe, A., Mitrovic, A.: Facilitating Deep Learning through Self-Explanation in an Open-ended Domain. Knowledge-based and Intelligent Engineering Systems, 10(1), 2006, 3-19

A Framework for Supporting the Application of Qualitative Spatiotemporal Reasoning

Carl P. L. Schultz¹, Robert Amor¹, Hans W. Guesgen²

¹ Department of Computer Science, The University of Auckland
csch050@ec.auckland.ac.nz
trebor@cs.auckland.ac.nz

² Institute of Information Sciences and Technology, Massey University
h.w.guesgen@massey.ac.nz

Abstract. Numerical approaches for representing and reasoning about information are ineffective when data is imprecise or uncertain. People on the other hand cope very effectively with vague information in daily life, for example when using spatial or temporal information. This has motivated the field of qualitative spatiotemporal reasoning (QSTR), which focuses on coarse, qualitative distinctions between spatial and temporal entities and relations. A substantial body of work has emerged from the QSTR community, however, serious difficulties prevent a uniform and general qualitative treatment of data representing space and time. Without unifying principles there is no basis for comparing the various QSTR approaches, and it is not always clear when and how QSTR should be applied. These issues must be addressed before QSTR can be properly integrated into standard software tools and practices. In this paper the first author's PhD programme is outlined, covering (a) the research aim of developing a framework for supporting the design and implementation of QSTR solutions, and (b) the research approach, which is based around the analysis of case studies, two of which are discussed.

Keywords: Artificial Intelligence, qualitative spatiotemporal reasoning, case study.

1 Introduction

Computers and software systems rely on numerical methods for representing and processing information, which work very effectively when data is certain and precise. However, uncertainty and imprecision are inherent properties of data that we gather from the physical world, and when probability distributions are unavailable or the numerical precision is not satisfactory, quantitative analysis methods break down. On the other hand people have a remarkable capacity to reason about and operate in the continuously changing physical world, considering that the information we have is necessarily vague and uncertain. In particular, people cope very effectively with everyday phenomena without resorting to detailed numerical analysis of a system or situation [1]. For example if I stay in the New Zealand sun for a short amount of time during summer, I will likely get sunburnt. I am not confident as to the exact number

of minutes it might take, and I have no information about the ultraviolet dosage required to cause damage. Despite having only very limited information, it is enough for me to know how to enjoy the summer without getting hurt! This approach is called qualitative reasoning (QR) [2], where the aim is to make the smallest number of distinctions between objects and relationships in order to complete a task in a given domain [3].

To explain general qualitative reasoning for physical systems consider the task of brewing a cup of coffee. It is enough for me to know that (a) the stove is hot, (b) the water in the coffee pot is at room temperature, and (c) if I place the coffee pot on the stove, the water will heat up and eventually boil. It is not necessary to use tools that provide numerical temperature readings, nor is any attempt made at solving numerical differential equations that model water as it heats up. Instead, we define a set of qualitative values that describe the interesting or relevant possible water temperatures: *room temperature, hot, boiling*. Qualitative relationships are used: *in, on*. Finally qualitative functions describe the relationship between variables in the system: *Stove-temperature influences water-temperature*. It is then possible to answer questions like: “What conditions must be satisfied to brew a cup of coffee? What temperature should I set the stove to, so that the water will boil?”.

More specialised qualitative approaches have focused on reasoning about time, resulting in a subfield called qualitative temporal reasoning, designed to manage coarse grained causality, action, and change in a software system. A notable and highly influential example is Allen’s elegant and efficient interval calculus [4], in which a set of thirteen atomic relations between time intervals is defined, a subset of which is shown in Figure 1. A composition table is provided which gives the possible temporal relations between the intervals t_1 and t_3 given relations for (t_1, t_2) and relations for (t_2, t_3) , along with an algorithm for reasoning about networks of relations. For example, if:

- I brush my teeth (t_1) *before* I have breakfast (t_2), and
- I have breakfast (t_2) *before* I leave for school (t_3), then
- Brushing my teeth (t_1) must also come *before* leaving for school (t_3).

A natural progression from qualitative temporal reasoning is to consider qualitative spatial reasoning (QSR) [1, 5, 6], where relationships between objects and regions are coarsely defined and reasoned about, concerning topology, shape, orientation, and distance. QSR can be used to answer questions like: “Are there any cafés *near* the university *in* Downtown?”. However, serious doubts have been raised regarding the possibility of a systematic, unified approach to QSR known as the poverty conjecture [7], stating the belief that no qualitative description of space exists that can be used to solve tasks in a variety of domains without problem specific metrical information. Despite this, significant progress has been made in a number of subfields, for example, Region Connection Calculus (RCC) [8] is a system used to reason about the topological relationships between regions, and in a similar fashion to Allen’s interval calculus, defines a set of qualitative spatial relationships that can exist between region pairs. Figure 1 illustrates a subset of these relations. Composition tables are provided, along with algorithms to reason about networks of region relationships.

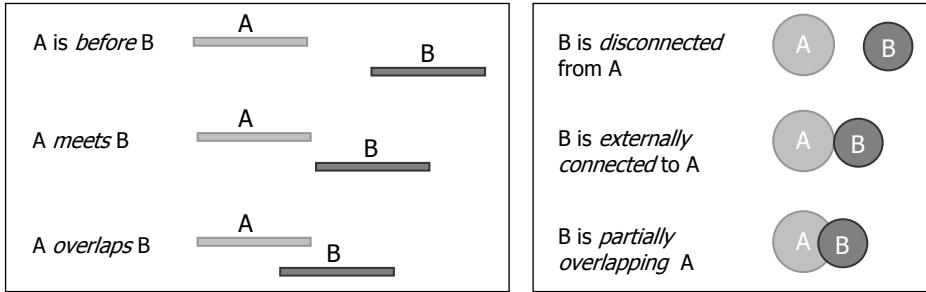


Fig. 1. Subset of the temporal relationships defined in Allen’s interval calculus [4] (left), where A and B are time intervals. Subset of the region relationships defined in Region Connection Calculus [9] (right), where A and B are regions.

Methods for qualitative reasoning about time or space are collectively known as qualitative spatiotemporal reasoning (QSTR). While a substantial body of theoretical work exists in QSTR, along with a host of industrial applications, a central problem is the lack of a unified framework that provides a standard for the various formalisms and techniques [3]. For example, QSTR formalisms have been developed that work at different granularities, addressing different aspects of a problem, and it is not clear how the various approaches relate to one another, thus making it difficult for researchers to exchange and compare results [3]. The fundamental problem is that the lack of principles and approaches for integrating a QSTR solution with standard software systems [3, 7]. In some cases a qualitative approach will greatly assist in solving a problem. In other cases it may: fail to reveal any insights; simply not apply to a domain; have no impact; or even complicate the problem. The first author’s PhD research project addresses this issue, with the overall aim of developing a framework for systematising the application of QSTR methods. The framework will be based on problem and QSTR method classification schemes, classification scheme relationships, metrics for quantifying aspects of applied QSTR, and best-practice software architecture design strategies.

2 Supporting QSTR Software Development

The overall aim of the first author’s PhD research is to support the development QSTR in software. The result will be a framework that acts as a practical guide for applying QSTR, aimed at software developers who are assumed to have little or no experience with the qualitative reasoning literature. Three main aspects will be addressed:

1. Making clear which qualitative technique is the most appropriate for a given type of problem
2. Establishing best-practice design methods in terms of software architecture
3. Quantifying the advantages, limitations and drawbacks of the proposed qualitative method, and, where possible, providing a means for measuring the potential benefits.

2.1 Objectives and Methodology for Developing the Framework

The tasks that are being undertaken towards the development of the proposed framework are (i) producing classification schemes for structuring the problem domain and the QSTR method domain, (ii) determining the associations between the two domains, (iii) developing metrics for assessing QSTR approaches, and (iv) establishing the most appropriate design strategies for applying qualitative methods in terms of software architecture.

In order to explore the possibilities of applied QSTR to determine the association between QSTR techniques and problems, a number of case studies are being undertaken, along with the analysis of other successful QSTR implementations. Conclusions drawn relating to the appropriate application and implementation of QSTR will direct the development of the proposed framework.

Classification schemes will be developed for classing QSTR methods and the problems that they can apply to. This is a necessary part for identifying which problems or tasks in general can benefit from a qualitative approach, and will be based on the common, salient characteristics shared across many similar problems and QSTR approaches. The schemes will specify which problems can be addressed by qualitative methods and will assist a person who is interested in exploring a qualitative solution. Furthermore, this will provide a platform for other qualitative spatiotemporal reasoning researchers to compare novel methods to existing ones. The sources of the data used for developing the classification schemes are the case studies that have been undertaken (discussed in Section 3), reviewing QSTR applications in the literature, and reviewing artificial intelligence problem solving literature.

Relationships between the attributes defined in the problem and the qualitative method classifications will be determined to provide a system for associating the two domains. Relationships will be identified by considering the trends in existing qualitative applications, by reviewing qualitative formalisms, and by conducting a deeper analysis of the way in which data associated with a problem is manipulated by the qualitative approaches.

Metrics will be developed for analysing the underlying qualitative formalisms in order to determine the most suitable approach for a given task, and to verify its applicability to the problem. The effectiveness of a qualitative approach must be quantified in terms of the problem being solved so that different qualitative methods can be systematically compared. For example, important factors are the degree to which a problem has been solved and the cost incurred for applying the solution.

Integrating qualitative methods into a task environment requires a clear understanding of the software components that must exist, and how the components must interact. Without information on the best practices for software architectural design, a developer who is applying a qualitative approach may produce software that is inefficient or even faulty. Providing this information will decrease the software design and development time, and will ensure that reliable and consistent implementation results are achieved.

3 Case Studies

The application of QSTR covers a wide range of disciplines apart from physical systems, including education, economics, and ecological and social sciences. To help classify the various problems that can benefit from a QSTR approach, five application-based case studies are being performed covering project management, robotics, astronomy, geographic information systems, and construction IT. The intention is to encounter, first hand, the issues that are raised when attempting to implement the proposed QSTR approaches. Case study analysis is conducted by referring to the current draft classification schemes, which are primarily based on more general artificial intelligence problem solving literature. From these case studies and other literature review-based work patterns are being identified and used to refine the classification schemes and the problem and QSTR method domain associations. In the following sections, two of the five studies are presented.

3.1 Case Study: Qualitative Query Support for GIS

Modern Geographic Information Systems (GIS) commonly provide powerful tools for manipulating, viewing and querying geographic information, allowing the isolation and informative presentation of relevant spatial features from typically large volumes of data. An effective querying system must provide flexibility, to appropriately capture a user's desired search criteria, and usability, so that the system is appropriately accessible. Despite this, standard GIS querying capabilities are often very limited, (particularly many publicly accessible web-based GIS), or require a user to have knowledge in specialised areas such as Structured Query Language (SQL) or set theory. By relying on numerical analysis techniques, GIS struggle with uncertain and imprecise information. As people communicate about spatial concepts using qualitative information it is desirable that a querying system support the use of such information and uncertainty. This application area raises issues regarding human-computer interaction (HCI), reasoning given uncertain and imprecise spatial criteria, and the management of large amounts of data for qualitative spatiotemporal reasoning.

We have developed a system called TreeSap GIS [10, 11] that explores the use of QSTR, and demonstrates its applicability towards more sophisticated, yet widely accessible, qualitative query support, as illustrated in Figure 2. TreeSap is capable of reasoning about qualitative distances (e.g. near, far, and so on) through transportation networks such as roads and bus services, and qualitative straight line distances. TreeSap can also position new features on a map that are defined solely in terms of the qualitative relationships that they must have with other features. For example, the requirements for a day care centre are that it must be moderately near residential areas via the road network so that it is accessible, and far away from motorways for noise reasons. TreeSap can attempt to find a suitable location for the day care centre, and give the user feedback on the degree to which the criteria have been met.

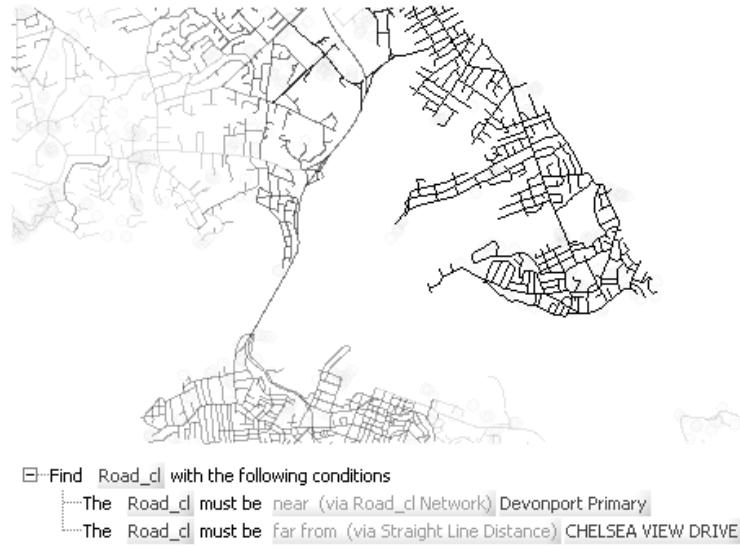


Fig. 2. Screenshot of the transparency method (top) used to visualise results of a qualitative query (bottom). The qualitative terms used to specify criteria (“very near”, etc.) capture the concept of vagueness and are accessible to non-experts in GIS.

3.2 Case Study: QSTR for Subjective Lighting Criteria in Architecture

The discipline of architecture is concerned with more than simply meeting practical criteria, such as: Can the building support the required load? Does the noise level, temperature, or airflow meet the appropriate health and safety standards? Architecture involves the study of how to direct a person’s perception of their environment, for example, to evoke a mood, or to convey an abstract concept. This involves managing contradictory requirements that are often difficult to resolve through purely numerical analysis; an example of this is the subjective impression, or atmosphere of a space that can be evoked by lighting.

In such cases, numerical approaches for representing and reasoning about lighting related information are not satisfactory. For example, the level of detail at which processing is being performed is often inappropriate, particularly for early stages of design. Issues regarding usability are raised as an architect, for example, must manually determine whether the desired aesthetic and functional requirements from a lighting configuration are being met, having been given lists of numerical data that can involve a mixture of units (resulting from numerical simulation of a designed model). Thus, issues raised in this application area include the human-computer interaction issue of managing subjectivity, reasoning given vague information, and integrating various vague pieces of information (e.g. “dim lighting, with sharp

shadows and striking highlights can evoke a dramatic and sophisticated atmosphere") into a reasoning framework.

Qualitative spatial reasoning is required to determine the relationship between light sources and surfaces in a room, such as identifying whether a light beam will arrive at a surface or if it is obstructed, qualitatively determining the degree of light uniformity across a surface, and so on. For example in certain cases the conical light beam shape can be reduced to a straight line which simplifies the occlusion test.

A software system is being developed that uses a QSTR engine for analysing a lighting installation and reporting on the subjective impressions that will be evoked. A mockup of the proposed system interface is illustrated in Figure 3.

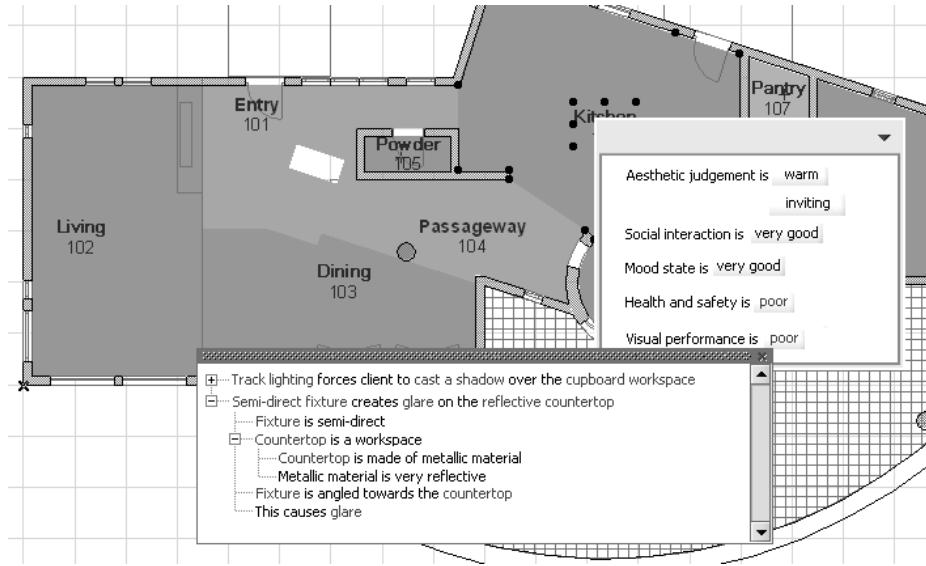


Fig. 3. Mockup screenshot of the interface used to analyse the subjective impact of a physical lighting configuration.

4 Conclusions

Qualitative spatiotemporal reasoning (QSTR) is a field of artificial intelligence motivated by the way that people handle vague and uncertain information about spatial and temporal phenomena in daily life. It addresses a number of limitations that arise when a system relies entirely on numerical methods for representing and processing data. A number of successful techniques and formalisms have emerged over the last 30 years, however a lack of design and implementation support, along with questions surrounding applicability, are hindering the field's ability to broaden its scope of application. This PhD research is focused on providing a framework that will tie the various aspects of QSTR together by identifying (a) important

characteristics of the problems being tackled with QSTR (b) important characteristics of the QSTR approaches being applied, and (c) the relevant interactions between problems and QSTR solutions. The framework will act as a practical guide for developers who are assumed to be unfamiliar with QSTR literature, in particular (i) making clear which qualitative technique is the most appropriate for a given type of problem, (ii) establishing best-practice design methods in terms of software architecture, and (iii) provide metrics to assess the overall solution quality by quantifying the advantages, limitations, and drawbacks of the proposed qualitative method. Development of the framework is currently being driven by five case studies, each involving the application of a QSTR method to a problem in a particular domain. Two studies were discussed: qualitative query support for GIS and QSTR engine for managing subjective lighting criteria in construction IT.

Acknowledgments. This work has been funded by the Bright Future Top Achiever Doctoral Scholarship (Tertiary Education Commission, New Zealand).

References

1. Hernandez, D.: Qualitative Representation of Spatial Knowledge. Lecture Notes in Computer Science, Vol. 804. Springer-Verlag, Germany (1991)
2. Forbus, K.D.: Qualitative Reasoning. In: Tucker, A.B. (ed): The Computer Science and Engineering Handbook, CRC Press (1996) 715-733
3. Bredeweg, B., Struss, P.: Current Topics in Qualitative Reasoning. In: AI Magazine, Vol. 24(4). AAAI (2003) 13-16
4. Allen, J.F.: Maintaining Knowledge about Temporal Intervals. In: Communications of the ACM, Vol. 26(11). ACM Press, New York (1983) 832-843
5. Hernandez, D., Jungert, E. (eds): Special Section on Qualitative Spatial Reasoning (Guest Editors' Introduction). In: Journal of Visual Languages and Computing, Vol. 9(1). Academic Press (1998) 1-3
6. Cohn, A., Hazarika, S.M.: Qualitative Spatial Representation and Reasoning: An Overview. In: Fundamenta Informaticae, Vol. 46(1-2). (2001) 1-29
7. Forbus, K.D.: Qualitative spatial reasoning: Framework and Frontiers. In: Glasgow, J., Narayanan, N., and Chandrasekaran, B. (eds): Diagrammatic Reasoning: Cognitive and Computational Perspectives. MIT Press (1995) 183-202
8. Cohn, A.G., Bennett, B., Gooday, J.M., Gotts, N.: RCC: A Calculus for Region Based Qualitative Spatial Reasoning. In: Geoinformatica, Vol. 1 (1997) 275-316
9. Cohn, A.G., Bennett, B., Gooday, J.M., Gotts, N.: Representing and Reasoning With Qualitative Spatial Relations About Regions. In: Stock, O. (ed): Temporal and Spatial Reasoning, Kluwer (1997) 97-134
10. Schultz, C.P.L., Clephane, T.R., Guesgen, H.W., Amor, R.: Utilization of Qualitative Spatial Reasoning in Geographic Information Systems. In: Riedl, A., Kainz, W., Elmes, G.A. (eds): 6th International Symposium on Spatial Data Handling: Progress in Spatial Data Handling, Vol. 12. Springer-Verlag, Berlin Heidelberg (2006) 27-42
11. Schultz, C.P.L., Guesgen, H.W., Amor, R.: Computer-Human Interaction Issues When Integrating Qualitative Spatial Reasoning Into Geographic Information Systems. In: 7th International Conference ACM SIGCHI-NZ: Design Centered HCI. ACM Press (2006) 43-51

A Constraint-Based Intelligent Tutoring System for the Java Programming Language

Jay Holland, Antonija Mitrovic

Intelligent Computer Tutoring Group,
Computer Science Department, University of Canterbury
Private Bag 4800, Christchurch, New Zealand
E-mail: {jah130,tanja}@cosc.canterbury.ac.nz

Abstract. This paper presents the design and implementation of Java-ITS, a constraint-based intelligent tutoring system for teaching the Java programming language. In order to learn programming, a student must acquire new cognitive skills, which when coupled with having to also learn the syntax of a particular programming language (necessary to apply a practical context to this skill), can make the process overwhelming. Even if a student can understand programming at a micro-level, to be a better programmer they must be aware of the overall design and context of a program, a useful skill that is often an after-thought. The goal of our project is to make the process of gaining programming skill both accessible through smoothing the learning curve, and relevant (from a practical perspective), such that transfer problems are reduced.

1 Introduction

Acquisition of computer programming skill is a core component of the Computer Science curriculum, a fact reflected in the many first-year tertiary prescriptions that require a student to undertake some kind of programming course. There are many aspects to programming theory, such as program control-flow and scope, and this variety can make it difficult for students already lacking a suitable information technology background [1]. It is generally accepted that the best way to introduce these ideas is through the teaching of a specific language. The Java programming language provides an appropriate introductory programming syllabus. Due to its low-level abstractions and system-independent nature, the student is able to concentrate more on the general programming concepts rather than system idiosyncrasies.

Although programming courses tend to have material taught in lectures, most of the learning reinforcement takes place in laboratories, where practical tasks are carried out. An increasingly popular and effective way of improving student learning is through Intelligent Tutoring Systems (ITSs), which enhance learning by providing feedback personalised to a student. These have been shown to be effective for many different disciplines and areas, including mathematics [2], physics [3] and database design [4]. The Java-ITS system, which is part of a master's of science project, is our attempt to teach the Java language to students, through a tutor that utilizes the constraint-based modelling (CBM) methodology [5]. Section 2 presents related work, fol-

lowed by the discussion of architecture and design decisions behind the system in Section 3. We conclude the paper by presenting future work in the final section.

2 Related Work

2.1 Intelligent Tutoring Systems

Personal tutoring is one of the most effective ways of enhancing learning. Due to growing populations and the complexities of some fields, personal tutors are not always readily available, whereas computers are becoming more and more commonplace. From the early days of computing Computer-Aided Instruction (CAI) has been a prominent field for research into how to achieve the same effectiveness as a personal human tutor. The first systems were primitive in terms of how they reacted to the students' behaviour; there was little or no adaptation to the student's progress, and they generally just followed a linear script. This changed with the advent of ITSs, which calculated the proficiency of students in various concepts related to the field of the tutor, and used this information to personalise the tutoring. This skill-tracking is known as *student modelling*. An interdisciplinary field, ITS theory draws from psychology and education as well as computer science, as we try to model and understand the cognitive process.

2.2 Constraint-Based Modeling

CBM handles student modelling by representing all domain knowledge in the form of state constraints. Each constraint is an ordered pair made up of a relevance condition and a satisfaction condition. For a given solution any relevant constraints (i.e. constraints whose relevance conditions are met by the students solution) must be satisfied to have the solution be evaluated as correct. Any constraint violations indicate an error in the solution; this in turn indicates the student has an incorrect understanding of the domain knowledge of each violated constraint.

2.3 Intelligent Programming Tutors

Very few ITSs teach general programming skills through free-form coding. Several ITSs focus on a single skill and tailor the interface to the particular skill; e.g Kumar's set of C++ and Java tutors [6], which teach, each in a separate system, expression evaluation, for loops, and C++ pointers, amongst other topics.

In terms of programming as a ‘coding’ activity, one of the most popular ITSs has been the Carnegie-Mellon LISP tutor [7]; it was also one of the first programming ITSs. A model-tracing tutor, it has provided a good starting point for other programming-tutor research. An evaluation of the system showed that its effectiveness approached that of a human tutor; on average, students covered the entire course curriculum in 15 hours, which was only 3.6 hours worse than the average time taken for

the students to complete the material with a human tutor (11.4), and 11.5 hours better than learning without either (26.5). Covering the material in a classroom setting takes over 40 hours.

The Java Intelligent Tutoring System (JITS) [8] is another Java tutor to allow free-form coding, albeit with no design section. The system is built around the core of the “intent-recognition algorithm” [9]. Several strategies are implemented to attempt to predict what the student was intending to accomplish with the code. One such strategy, the “Syntax Error Correction Strategy”, finds unrecognisable tokens in a student’s submission, and reverses possible error transformations such as the replacement of a symbol by another symbol, or the insertion of an extraneous symbol, to attempt to correct to code; if a more meaningful code chunk is obtained, the system assumes that a syntax error is present, and that the corrected code chunk represents the true intent of the student. JITS will then initiate a dialogue with the student, which includes questions about sections of the code; for example, the system may ask ‘I see ‘intt’. Do you mean the keyword ‘int’?’ The dialogue continues until the code is completely correct.

3 The Java-ITS System

Java-ITS is an intelligent tutoring system where students can form solutions to various Java programming problems, and receive feedback on their solutions. The curriculum the system supports is a subset of the Java programming language; as the complete Java domain is vast, incorporating all the domain knowledge into a single tutor, while not impossible, would require an immense amount of effort to validate that everything was correctly implemented. By working with a subset, we are able to still give an appropriate learning experience, yet maintain validity of the concepts taught. If there is a need to extend the tutor in the future to cover more of the domain, more concepts can be incrementally added and validated by extending the domain model and problem set. As the target audience of the tutor is novice programmers, the curriculum begins with the most elementary of concepts and follows a typical tertiary course progression. By working through all the problems in the system, the student should gain a good understanding of all programming concepts up to and including loops.

3.1 Architecture

The system architecture of Java-ITS adheres closely to the architecture of other constraint-based tutors, as illustrated in Figure 1. All information and interaction is presented to the student through a web interface, which can be viewed in any mainstream web-browser. The session manager handles any requests from the web-server. It works as a hub, and interacts with most other parts of the system at some point. Pedagogical decisions and operations take place in the pedagogical module (PM). This receives the interactions (via the web-server and the session-manager) from the student, such as problem selection and solution submission.

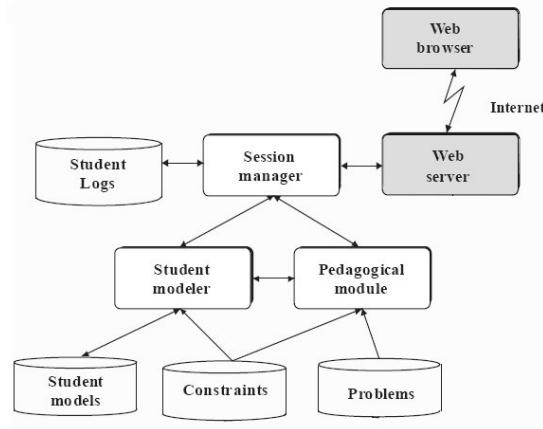


Figure 1: The Architecture of Java-ITS

The constraint store is the knowledge base of the system, and contains all the constraints that make up the domain model. The constraints can be divided into three categories: ones that examine syntactic properties, ones that examine semantic properties of a solution, and ones that examine style – although there are many ways to write a program, it is better to encourage good practice. Each constraint contains three parts: a relevance condition, a satisfaction condition, and a feedback message. The relevance and satisfaction conditions are examined during evaluation, and the feedback message is shown to the student if that constraint is violated.

In terms of the problem set, with many tutors the goal of each problem within the tutor will be similar, with a general template being used to generate further problems in the same goal set. With programming, the area is so broad that different skills are required by programming solutions to problems with vastly different outcomes that are hard to generalise; therefore, in a programming tutor, it is difficult to keep to just one form without restricting learning. To solve this issue in Java-ITS, the tutor's problems have been broken into groups, with each group containing problems of a certain type. For example, here is an example of a problem of an ‘iteration’ type:

"Bob has two cats, Fluffy and Whiskers. Fluffy needs to be given milk every day, but Whiskers only needs to be given milk every second day. Complete the following method, `feedCats`, such that it iterates over given number of days, and milk is only given to each cat on the appropriate days."

Each problem is presented with its own *context*. The context is a code fragment that frames a problem, and is displayed inside the solution workspace, so that the student has existing properties to work with. For example, the context could be a ‘for’ loop beginning and end, or a method outline (signature and braces). Often there will be variables and other methods that the student can reference in their own code, such as arguments to methods; in fact, often these variables will be mentioned directly in the problem text itself, and therefore the student will be expected to use them in some way. The context for the previous `feedCats` problem is as follows:

```

Cat whiskers;
Cat fluffy;
public void feedCats (int days) {
}

```

Each problem has a corresponding ideal-solution, which is used by the semantic constraints during evaluation to semantically validate the student's solution. It describes an abstract version of what is required from the submission, i.e. rather than explicitly specifying what design concepts and code fragments should occur in the submitted solution, it only notes the general requirements of the given problem that must manifest in the solution for the problem's tasks to be considered satisfied, such as "the solution must return this variable" or "must loop up to this value". It is essentially a formal specification of the problem statement.

The Student Modeler (SM) is responsible for maintaining the student models. It receives the list of violated and satisfied constraints from the solution evaluation, and appends this information to a student's model. The SM can also provide details relating to a model, such as how well a student knows a particular constraint; this information will be used by the pedagogical module to make problem-selection decisions.

Via the web-based interface, the student is able to perform all necessary interactions. These include operations such as problem selection, problem solving, and submitting a solution. The problem solving interface, illustrated in Figure 2, is where the student will spend most of their time. The screen is split into several panes, with four main panes being related to tutoring activities. The top pane presents the problem text to the student, while the large middle pane is the solution workspace and allows the student to form a solution to the given problem. The bottom pane contains components (tiles) to be used in solution formation, and the rightmost pane presents feedback on a student's solution.

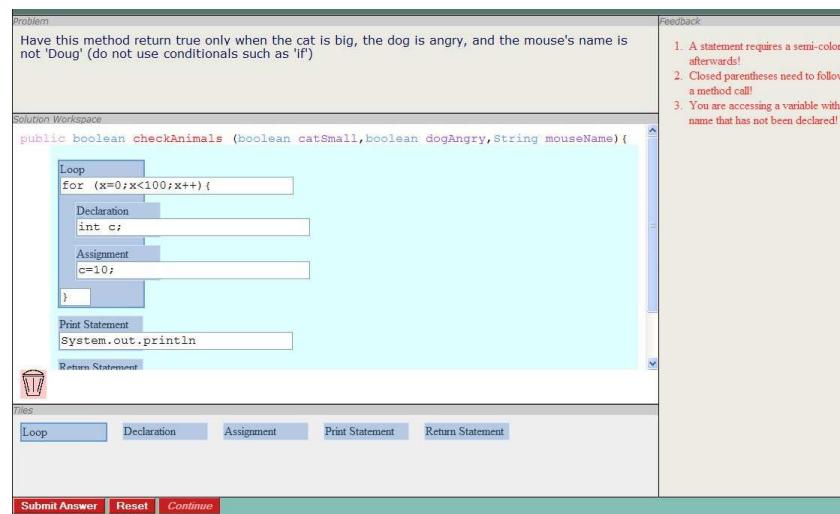


Figure 2: Problem Solving Interface (Coding Stage)

3.2 Problem-Solving Stages

The interface (and the general layout of the task) has been designed such that the student is better able to handle the complexity of a program, by first presenting the student with generic programming constructs in the form of tiles. Problem solving is done in two phases: the student firstly designs the program in terms of generalised solutions using tiles, followed by a coding stage, where he/she enters actual code fragments into the tiles.

A typical tutoring session progresses as such: The student selects a problem, and they are presented with the problem-solving interface, set to the design stage, populated with any information necessary to the task (problem text, context, solution tiles). The student then designs the solution using the tiles, and submits the solution when he/she believes they have a correct solution, or is unable to continue. The system then returns feedback to the student, revealing any errors that may exist in the solution. The student can use this feedback to try and correct the solution, and submit again. This submit/feedback loop will continue until the student correctly forms a solution, at which point the feedback will indicate that the solution is indeed correct, and the student can then move on to the coding stage. A similar process as with the design stage then takes place, albeit with code fragments instead. Once the student submits a correct solution to the coding stage, then they can move on to the next problem.

The design stage is characterised by the use of *tiles*. Each tile, housed in the tile pane below the solution workspace, represents a different abstract programming construct, such as a loop or a variable declaration. Tiles can be categorised into two groups: *statement* tiles, which are equivalent to a line of code in Java, and *block* tiles, which also act as containers for other tiles (a method is an example of a block tile).

To design solutions, students drag-and-drop tiles from the tile pane and place them inside the solution workspace input areas, and possibly inside other tiles. Using the tile pane is a ‘cloning’ action - the tiles will not disappear from the tile pane once used, and the student is free to make as many copies as needed. Tiles in the solution workspace can be deleted if necessary, and once placed inside the solution, they can still be moved to other parts of the solution.

The coding stage is built upon the solution the student generated previously in the design stage. Upon entering the coding stage, the student is presented with the tile-based solution developed in the design stage, but text-entry boxes have now been inserted inside each tile (two for block tiles), and the tiles themselves are now immovable (the ability to add new tiles to the solution has also been taken away). The student must now complete these text entry boxes by entering Java code. The result of this is that the solution, once completed, will resemble a real-life code listing.

The two-stage principle is a product of the main goals of the system. The first goal was to provide accessible tutoring - as programming is a complex activity, the system would benefit the student's learning by providing a way for the student to handle that complexity, therefore it is mandatory for the student to design the solution first using tiles. By making the design task explicit, the student will have to consider the structure. The second goal was to reduce the transfer problems between the tutoring system and a real-life coding situation; therefore the system was designed such that the student would be at some stage entering code themselves, similar to using a text editor during normal programming tasks. There is evidence that suggests that although stu-

dents tend to retain more information for a purely text-entering approach, they enjoy using the system less than symbolic approaches [10]. By combining symbolic and partial free-form text, we hope to receive the benefits of both approaches. Having the symbolic approach first reduces the memory load during the coding stage, and also smoothes out the learning curve.

The two-stage approach also aids the server-side solution evaluation; by reducing the ambiguity of the student's completed solution, the amount of reasoning required by the system is reduced, which in turn simplifies the system implementation. If we were to evaluate a solution composed of only free-form text, then we would run into the same problems as a compiler would; compiler messages are often misleading due to fact that common errors include missing semicolons or braces, meaning it can be difficult to tell where a method really ends, or the boundaries of a statement. Also, the intent of the student is not always clear inside free-form text. If a student were to create a line of code that was completely syntactically incorrect, we may not be able to tell if it they were trying to write a conditional or simply make a method call; in situations like this feedback would be limited and general, and possibly inaccurate. By forcing the student to enter their code inside the tiles, we are able to decipher the intent of a statement by the tile type, and can provide more specific feedback.

3.3 Solution Evaluation

The student modeller module handles solution evaluation. Once a submission is received from the student, all constraints are evaluated over the solution in two stages. The relevance conditions are initially evaluated to deduce which constraints are relevant to the current problem and/or solution. The ones that are considered relevant then have their satisfaction conditions evaluated – any relevant constraints not passing this stage will be considered violated, which means the student has not learnt the domain concept the constraint represents, otherwise they will be considered satisfied, indicating the student does understand the concept.

The process is executed by propagating the solution through a constraint network, which is loosely based on a Rete network, in order to optimise the potentially intensive procedure. Each node in the network references part or all of a condition, and specifies which constraints the condition came from, such that the system knows which constraints to apply to the result of evaluating the node's condition.

4 Future Work

Java-ITS has not yet been evaluated as to its effectiveness as a tutoring tool. A full evaluation is planned for early 2007, to be taken with an introductory Java programming class.

Although supporting the entire Java domain is beyond the immediate scope of this project, the system can be incrementally improved through developing more constraints, therefore extending the domain coverage. In addition, the problem set can also be increased, by either working inside a template to develop more complex problems (for example increasing the number of clauses), or developing new templates

that focus on different problem goals. If any new design concepts are introduced through new constraints and problems, then new tiles will need to be introduced into the interface. More research must be made into ‘good practice’ coding style in order to generate more style constraints.

The system can also be extended to support problem and feedback-level selection. Currently, the student manually selects the problem they wish to work on, but the system can be adapted to provide suggestions or making problems mandatory, depending on how well a student understands a concept; if a student is unfamiliar with the loop construct, we can suggest problems which deal lightly with loops at first, then move them on to more complex problems. With feedback-level selection, we can give varying degrees of hints, to allow the student to think more about the problem themselves before receiving more specific feedback.

References

1. Pillay, N.: Developing Intelligent Programming Tutors for Novice Programmers. Inroads - the SIGCSE Bulletin **35** (2003) 78-82
2. Singley, M.K., Anderson, J.R., Gevins, J.S., Hoffman, D.: The algebra word problem tutor. Artificial Intelligence and Education (1989) 267-275
3. Gertner, A., VanLehn, K.: Andes: A Coached Problem Solving Environment for Physics. In: VanLehn, K. (ed.): Intelligent Tutoring Systems: 5th International Conference. Berlin: Springer (2000) 131-142
4. Suraweera, P., Mitrovic, A.: An Intelligent Tutoring System for Entity Relationship Modeling. Int. J. Artificial Intelligence in Education **14** (2004) 375-417
5. Ohlsson, S.: Constraint-based Student Modeling. Student Modeling: the Key to Individualized Knowledge-based Instruction. Springer-Verlag (1994) 167-189
6. Kumar, A.: Web-based tutors for learning programming in C++/Java. Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science education (2004) 266-266
7. Anderson, J.R., Reiser, B.J.: The LISP tutor. Byte **10** (1985) 159-175
8. Sykes, E.R., Franek, F.: An Intelligent Tutoring System Prototype for Learning to Program Java. 3rd IEEE International Conference on Advanced Learning Technologies, Athens, Greece (2003) 485-486
9. Sykes, E.R., Franek, F.: Inside the Java Intelligent Tutoring System Prototype: Parsing Student Code Submissions with Intent Recognition. IASTED International Conference on Computers and Advanced Technology in Education, Innsbruck, Austria (2004) 613-618
10. Corbett, A.T., Anderson, J.R., Fincham, J.M.: Menu selection vs. typing: effects on learning in an intelligent programming tutor. International Conference of the Learning Sciences, Evanston, IL (1991) 107-112

Computational Model of Plan Competition in the Prefrontal Cortex

Gregory A. Caza

Department of Computer Science, University of Otago, PO Box 56, Dunedin, New Zealand
gcaza@cs.otago.ac.nz

Abstract. Human behaviour consists of more than simple, conditioned responses to stimuli in the environment. Actions are influenced by intentions because plans are consciously formulated to reach certain goals. Each stimulus may also be associated with more than one plan or goal. I am developing a neural network model of how plans compete in the brain to influence behaviour.

Keywords: neural network computational model cortex frontal Hopfield plan competition cognitive

1 Introduction

In the human brain, the **prefrontal cortex (PFC)** is believed to be an important component of cognitive control [1], cases where behaviour is guided by intentions or conscious goals. According to this theory, the PFC is used to respond to stimuli in a task-relevant manner, potentially overriding habitual tendencies in order to choose a response that better achieves the current cognitive goal.

For example, a healthy adult human can quite easily learn to hit a sequence of keys, “A-B-C-D”, whenever a light is turned on. Such behaviour can become so practised that it requires little or no conscious thought. However, say the instructions evolved and an alternate sequence of keys, “E-F-G-H” was required if the light was flashing. A conscious intention, or cognitive plan, must overcome the habitual response, “A-B-C-D”, in the exception cases.

There is evidence that switching plans in such a way is more difficult for patients with damage to the frontal part of the brain [2]. Thus, it is theorized that the PFC maintains an active representation of the current cognitive goal and uses that plan to influence behaviour [3].

2 Background and Project Goals

This plan competition model is one part of the thesis for a Master’s degree in Cognitive Science. The full scope of my research is an investigation into language acquisition in infants. In short, my proposal is that an infant must switch from an action-observation plan to a learning “plan” in the appropriate situations.

The plan competition is one small, yet vital, component of a larger computational network that will model situations where an infant learns the meaning of new words. There is no space here to elaborate on matters of linguistics or developmental psychology, so this paper will focus on the project to model plan competition.

The goal of this subproject is to create a neural network to model an element of the PFC that controls competition between cognitive plans. This type of stimulus-response control is very important because more than one plan—and hence, more than one response—can be associated with a given stimulus.

For example, when you see your computer, you can check your email *or* you can play a game. Both might involve grabbing the mouse as the first step in the plan, but the sequence of behaviour eventually diverges. The choice of which action to take is clearly not an automatic process, or the uncontrolled response would never change (e.g., always check email and never do anything else.) Similarly, a human with a fully-functional PFC must be able to actually make a decision to avoid being frozen into inaction when confronted with the computer as a stimulus.

The *Plan Competition Network* being developed is based on a combination of pre-existing neural network principles. Implementation decisions were motivated by behavioural and neurological evidence in the literature. A brief summary of the research that shaped the model is presented below.

2.1 Models of Serial Behaviour

Contention scheduling is a term used to refer to “routine selection between routine actions” [4]. Contention scheduling networks have been used to model serially-ordered behaviour. One model of contention scheduling, which Houghton and Hartley dubbed **competitive queuing**, appears in Figure 1.

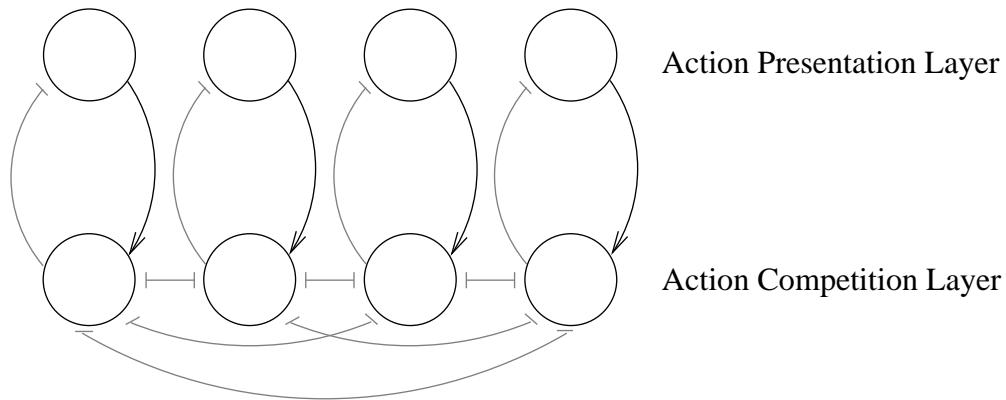


Fig. 1. A ‘competitive queuing’ model that demonstrates contention scheduling [5]. Arrow-headed connections are excitatory; other connections are inhibitory.

In the model shown, the presentation layer represents the action schemata, consisting of tasks to be completed serially. The competition layer acts as a competitive filter. The lateral inhibition in the lower layer enforces a ‘winner-take-all’ decision, with one neuron becoming activated while simultaneously inhibiting its competitors. Finally, this selected response will inhibit itself through the connections back to the presentation layer. Thus, if a gradient of activation is provided as input, the strongest response will be chosen. But, after that response inhibits itself, the next strongest will rise to the top, and so on.

This sort of competitive queuing model is useful for simulating the execution order of short, ballistic actions such as typing [5]; there is good neurological evidence [6] that it is an accurate representation of what occurs in the monkey PFC during fast drawing of geometric shapes. Such models are too simple, however, to represent action sequences that occur over long or unpredictable courses of time. Also, the discrete units in the competition layer can only represent isolated actions and not multi-step, alternative sequences or plans.

2.2 Hopfield Model

The **Hopfield Net** [7] is a well-established neural network model of associative memory. The basic Hopfield Net consists of fully-connected neurons, with no neuron connected back to itself. All connections are two-way and the weight for each connection is symmetric. Connection weights begin at zero, and will fluctuate in strength as the network learns. Connections will be either excitatory or inhibitory, depending on the whether the weight is positive or negative; it is possible for a weight to change sign multiple times during training. Figure 2 shows an example of a small Hopfield Net.

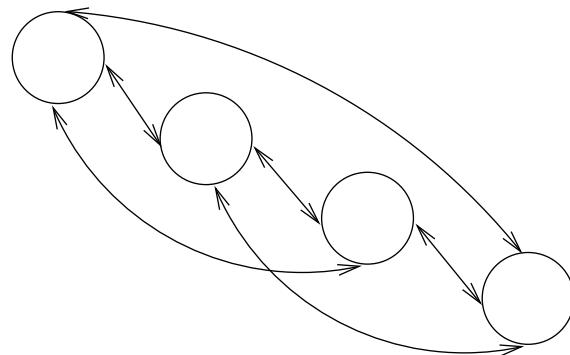


Fig. 2. Hopfield Net [7]. All connections are shown as excitatory for the purposes of illustration, but may become inhibitory during training.

Each node has an external input. During training, this input value is fed into an activation function which is used to set the output value of the node;

connection weights are adjusted accordingly. During testing, each output is ‘soft-clamped’, or initialized, to the value of its external input. All nodes are then settled, asynchronously and in random order, until the output of each node reflects the activated value of the sum of its inputs.

The network becomes, essentially, a pattern completer; it is attracted to, and tends to converge on, a stable state that consists of a group of associated neurons from the training data. For example, say the 4-node network in the diagram was trained to two patterns: [node0 node1] and [node2 node3]. After training, if the external input for node0 was activated in isolation, then the network would settle into the completed pattern: [node0 node1].

2.3 Backward Inhibition

One valuable concept to borrow from competitive queuing is the idea of inhibition back to the presentation layer. There must be a mechanism for clearing the last plan from memory. The phenomenon of self-inhibition in visual attention is well-established [8]; it takes a person longer to look back to a location recently attended to.

There is also evidence of similar inhibition in task-switching, meaning the act of shifting back to a recently abandoned plan also suffers from increased reaction times. There is behavioural evidence [2] of this **backward inhibition**, and neurological evidence [9] suggests it involves activity in the PFC. The ‘switch costs’ required to move between cognitive task sets have even been modelled computationally [10].

3 The Plan Competition Model

I have developed a multi-layer neural network to model plan competition in the PFC. The model consists of a *Plan Presentation Layer*, a *Plan Competition Layer*, and a *Plan Termination Gate*. Each of these elements is briefly described below. Figure 3 shows a simplified version, using a reduced number of nodes.

3.1 Design

The Plan Competition Layer is a Hopfield Net, trained to attract to a stable state of associated neurons that represent the winning plan. It is connected to another layer that simulates possible plans, based on current stimuli. Each neuron in this Plan Presentation Layer has a strong forward connection to a neuron in the competition layer, acting as the external input would in a standard Hopfield Net.

This model evolves the backward inhibition property of contention scheduling. Each competition neuron also has an inhibitory connection back to the presentation neuron that is providing it with input. These backward connections are gated by the Plan Termination Gate, meaning it controls the inhibition and only ‘turns it on’ under certain circumstances. (There is neurobiological evidence

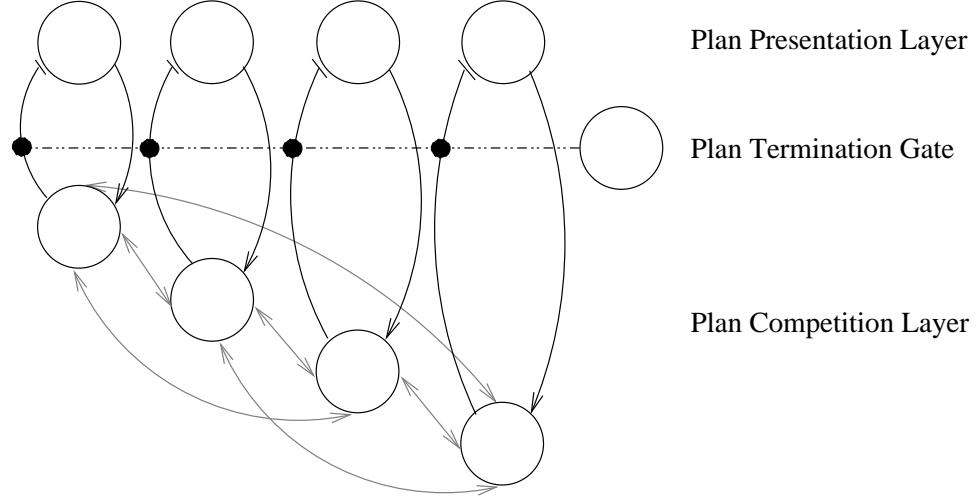


Fig. 3. A simplified representation of the Plan Competition Network. Black connections are of fixed weight, while grey connections vary in strength. The dotted line represents a gate on the inhibitory connections.

for gating systems in the PFC [3].) It is hypothesized that this gating is used to terminate the current plan and make way for subsequent plans. The simplest reason the current plan could be terminated is because it has been successfully completed. However, there also needs to be cognitive ‘escape plan’ in case the current goal is unachievable or abandoned for other reasons.

The Plan Competition Network is very similar to the competitive queuing models presented above. One main difference is that the competition layer, represented here by the Hopfield Net, is not one of simple lateral inhibition with an individual neuron ‘winning’ over all competitors. Rather, it is a ‘winning-assemble-takes-all’ system, in which a *group* of associated neurons will become active and inhibit their competitors. In that sense, this network has more in common with later models of contention scheduling that predict the selection of competing action schemata [11].

The other main difference is that the return inhibition to the presentation layer is gated, allowing the plan (i.e., the input) to remain in active memory until the sequence has been completed. A simple, time-delayed backward inhibition is insufficient because of the variability in the amount of time it takes to complete a plan.

3.2 Training Parameters

During training, a pattern is presented at the external inputs and propagated forward from the presentation layer to the competition layer. Because all of the flexible connections are symmetric, learning is performed on a by-connection basis, according to the following formula:

$$\Delta w_{ij} = \epsilon y'_i y'_j \lambda . \quad (1)$$

The change in weight, w , for the connection between neuron i and neuron j is proportional to the product of the *training* output (see below), y'_n , of each neuron. Here, ϵ represents the learning constant which Hopfield [7] set to $\frac{1}{N}$, where N is the total number of neurons. And, λ is a modification of **Oja's Rule** [12], which keeps the weights from growing to infinite magnitude. In this case, λ is the difference between 1 and the absolute value of the weight's current strength. The net effect is to make the values asymptotic as they approach unit magnitude.

Substituting the expanded terms into 1 yields the following full equation:

$$\Delta w_{ij} = \frac{x_i x_j (1 - |w|)}{N} . \quad (2)$$

A scaled training output is used so that neurons that are off at the same time will be associated by having their mutual connections strengthened. The training output is related to the neuron output, y_n , by the following formula:

$$y'_n = 2y_n - 1 . \quad (3)$$

Thus, for neurons with discrete activation, the training output 3 is simply:

$$y'_n = \begin{cases} 1 & \text{if } y_n == 1 \\ -1 & \text{otherwise} \end{cases} . \quad (4)$$

Experiments to date have been performed using various sizes of training data, as well as smaller values of the learning parameter, ϵ . It is not yet known if these equations will require modification under a network where the neurons have a continuous activation function.

3.3 Results

The ideal network will have at least the following characteristics, after training:

1. If there is no input, no neurons will be activated. This case represents a 'no current plan' situation.
2. If the current input matches a plan exactly, that plan will win the competition. In the competition layer, the associated neurons will be activated and all others will be off.
3. If the current input is the subset of a plan, the full plan will be activated.
4. If the current input presents multiple plans at equal strength, **one** of the plans will win.
5. If multiple plans are presented as an input gradient, the strongest plan will win.
6. After gating, the last winning plan will be deactivated.
7. If multiple plans are presented and the inhibition is gated, the next strongest plan will take over.

The first three goals have already been achieved, as have the last two. For multiple plan situations, the network is currently unpredictable. Most of the time, one plan will win. However, because settling occurs asynchronously in a random order, occasionally a mixture of the two plans will result. The greater the difference in the relative strengths of the two plans, the more likely a clear winner will be chosen. Further research should serve to optimize the behaviour of the Plan Competition network in multiple plan cases.

Initial research consisted of neurons with a discrete, step-based activation function. All neurons in the network are currently moving to a continuous activation function. Further experimentation is planned to determine the most appropriate function.

4 Future Work

There are a few future steps to be taken, in addition to those already mentioned. Different approaches [13–15] are currently being investigated to improve the performance and neurological plausibility of the Hopfield layer of the network.

Another layer will be added to the network to represent stimulus-response interaction with the surrounding environment. The goal will be to train the network to gate the inhibition at appropriate times. Thus, clearing the current plan from the PFC will become *learned* behaviour. Eventually, the Plan Competition Network will be integrated with a word-learning module in order to achieve the full goals of the thesis.

References

1. Miller, E.K., Cohen, J.D.: An Integrative Theory of Prefrontal Cortex Function. *Annual Review of Neuroscience* **24** (2001) 167–202
2. Mayr, U., Keele, S.W.: Changing Internal Constraints on Action: The Role of Backward Inhibition. *Journal of Experimental Psychology* **129**(1) (2000) 4–26
3. Braver, T.S., Cohen, J.D.: On the Control of Control: The Role of Dopamine in Regulating Prefrontal Function and Working Memory. In: *Control of Cognitive Processes: Attention and Performance. Volume XVIII.*, Cambridge, MA, USA, MIT Press (2000) 713–737
4. Shallice, T.: From Neuropsychology to Mental Structure. Cambridge University Press, Cambridge, MA, USA (1988)
5. Houghton, G., Hartley, T.: Parallel Models of Serial Behaviour: Lashley Revisited. *Psyche* **2**(25) (1995) 1–25
6. Averbeck, B.B., Chafee, M.V., Crowe, D.A., Georgopoulos, A.P.: Parallel processing of serial movements in prefrontal cortex. *Proceedings of the National Academy of Sciences of the United States of America* **99**(20) (October 2002) 13172–13177
7. Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America* **79** (1982) 2554–2558
8. Dorris, M.C., Klein, R.M., Everling, S., Munoz, D.P.: Contribution of the Primate Superior Colliculus to Inhibition of Return. *Journal of Cognitive Neuroscience* **14**(8) (2002) 1256–1263

9. Dreher, J.C., Berman, K.F.: Fractionating the neural substrate of cognitive control processes. *Proceedings of the National Academy of Sciences of the United States of America* **99**(22) (2002) 14595–14600
10. Gilbert, S.J., Shallice, T.: Task Switching: A PDP Model. *Cognitive Psychology* **44** (2002) 297–337
11. Cooper, R., Shallice, T.: Contention Scheduling and the Control of Routine Activities. *Cognitive Neuropsychology* **17**(4) (2000) 297–338
12. Hertz, J., Krogh, A., Palmer, R.G.: Introduction to the theory of neural computation. Addison-Wesley, Redwood City, California, USA (1991)
13. Johansson, C., Rehn, M., Lansner, A.: Attractor neural networks with patchy connectivity. *Neurocomputing* **69** (2006) 627–633
14. Zeng, X., Martinez, T.: A New Relaxation Procedure in the Hopfield Network for Solving Optimization Problems. *Neural Processing Letters* **10** (1999) 211–222
15. Toulouse, G., Dehaene, S., Changeux, J.P.: Spin Glass Model of Learning by Selection. *Proceedings of the National Academy of Sciences of the United States of America* **83** (March 1986) 1695–1698

Computing Semantic Relatedness using Wikipedia Link Structure

David Milne

Department of Computer Science,
The University of Waikato, Hamilton, New Zealand
dnk2@cs.waikato.ac.nz

Abstract. This paper describes a new technique for obtaining measures of semantic relatedness. Like other recent approaches, it uses Wikipedia to provide a vast amount of structured world knowledge about the terms of interest. Our system, the Wikipedia Link Vector Model or WLVM, is unique in that it does so using only the hyperlink structure of Wikipedia rather than its full textual content. To evaluate the algorithm we use a large, widely used test set of manually defined measures of semantic relatedness as our bench-mark. This allows direct comparison of our system with other similar techniques.

Keywords: Wikipedia, Data Mining, Semantic Relatedness

1 Introduction

How related are “love” and “sex”? This is a delicate question. Any answer is bound to be subjective—and revealing. But what if we were to consult a dispassionate, objective computer? According to the techniques described in this paper, the answer would be a clinical 67%.

Making such judgments about the semantic relatedness of different terms is a routine yet deceptively complex task. To perform it, we draw not only on our attitudes and personal background, but also on an immense amount of background knowledge about the concepts that these terms represent. Any attempt to compute semantic relatedness automatically must do the same. Many techniques use statistical analysis of large corpora to provide this context. Others use hand-crafted lexical structures such as taxonomies and thesauri. In either case it is the background knowledge that is the limiting factor; for the former approach it is unstructured and imprecise, and for the later it is limited in scope and scalability.

These limitations are the motivations behind several new techniques which infer semantic relatedness from the structure and content of Wikipedia. With over a million articles and thousands of contributors, this massive online repository of knowledge is easily the largest, fastest growing encyclopedia in existence. With its extensive network of cross-references, portals and categories it also contains a huge amount of explicitly defined semantics. This rare combination of scale and structure makes Wikipedia an attractive resource for this work and for other NLP applications.

This paper describes a new technique, the Wikipedia Link Vector Model (or WLVM), which calculates semantic relatedness between terms using the links found within their corresponding Wikipedia articles. Unlike similar techniques, it is able to provide relatively accurate measures using only the link structure and titles of articles, rather than their textual content. Before delving into the details of this approach, we first describe its context in terms of the larger work in which it exists, and the other systems to which it can be compared. This is followed by a description of the algorithm, and its evaluation using a well known data set of manual judgments of semantic relatedness. The paper concludes with a discussion of the strengths and weaknesses of the approach, and directions for possible improvement.

2 Context of the research

The research described in this paper is motivated by the larger goal of improving the way in which we locate information. Whenever we seek out new knowledge—whenever we turn to the ubiquitous search engines—there is a fundamental paradox that must be grappled with: how can one describe the unknown? This is because a query is not simply a statement of intent as is commonly thought. Instead it is an excerpt, a few words or phrases, from within a relevant document. To form an effective query, one must predict not only what information this relevant document contains, but also the terms by which this is expressed. In short, one must already know a great deal of what is being sought, in order to find it.

What knowledge seekers need—at least those who are not clairvoyant—is a bridge between what they can describe and the information they seek; between their query terms and the topics and terminology of the documents available. One possible bridge is a thesaurus; a map of semantic relations between words and phrases. Knowledge seekers who cannot identify the effective terms for their query could use a thesaurus that covers the terminology of both documents and potential queries, and describes relations to bridge between them. Seekers who cannot form a specific query at all could use a sensibly organized thesaurus that exposes the topics available and allows them to be explored.

Current use of thesauri for retrieval is limited. They are extremely expensive to produce, and thus are only available for a small portion of document sets. The research which forms the context of this paper aims to address this by developing a framework by which thesauri can be produced cheaply for any document collection. Ideally, these thesauri should be as accurate, relevant, and browsable as their expensive, manually defined counterparts. This is an ambitious goal. Existing techniques for building thesauri automatically are attractive for their low cost and their ability to match the content of documents exactly, but are woefully inferior in terms of accuracy and conciseness. There are many other techniques available, such as query log mining and web link analysis, but their use is typically limited to behind-the-scenes processing due to their inaccuracy.

The most recent—and perhaps the most promising—development for this research is the emergence of internet communities such as Wikipedia and del.icio.us which can be exploited directly for the task of organizing information. These offer terms and

relations defined by human intelligence (as opposed to statistical or lexical approximations), constant maintenance, coverage of swiftly changing domains, and reflection of contemporary language and interests. All this is achieved through the exploitation of existing public efforts, without the cost associated with traditional thesauri.

Identifying the semantic relatedness of terms and concepts within Wikipedia is an important step in extracting sensible, browsable thesauri from it. Previous work on this problem has shown that most of the relations described by Wikipedia's structure—the hierarchical relations between categories and the interlinking of articles—cannot be directly mapped to traditional thesauri [1]. Many direct links are irrelevant and need to be discarded, while other additional relations need to be inferred from chains of links. An accurate measure of semantic relatedness would be a valuable guide in either case, and would be extremely useful for disambiguating raw text or entries in other thesauri to concepts in Wikipedia.

3 Related Work

Measures of semantic relatedness are used extensively in natural language processing, in such applications as word sense disambiguation, text summarization, and information extraction. The most direct method for evaluating these measures is to compare them with judgments made manually. The largest, most widely used test set for this purpose is the WordSimilarity-353 collection [2]. This contains 353 word pairs for which at least 13 manual judgments of similarity (on a scale of 0-10) are specified. Despite the subjective nature of such judgments, agreement between them was relatively high; the average correlation between an individual participant's judgments and those of the whole group was 0.79 according to Spearman rank-order correlation [3].

Table 1: Performance of semantic relatedness measures

Measure	Correlation with manual judgments	Reference
WordNet	0.33–0.35	[6]
Roget's Thesaurus	0.55	[5]
LSA	0.56	[4]
WikiRelate!	0.19 – 0.48	[7]
ESA	0.75	[8]

As described previously, the central point of difference between the various techniques for obtaining semantic relatedness measures is their source of background knowledge. Corpus based approaches obtain this from performing statistical analysis of large untagged document collections. The most successful of these is Latent Semantic Analysis (LSA) whose measures, as shown in Table 1, have a 0.56 correlation with manual judgments [4]. Other techniques make use of structured

thesauri such as Roget [5] and Wordnet [6], but suffer reduced accuracy and can only provide judgments for a limited number of terms.

Strube and Ponzetto [7] were the first to compute measures of semantic relatedness using Wikipedia. Their approach, known as WikiRelate!, took familiar techniques that had previously been applied to WordNet and modified them to take advantage of the data found within Wikipedia. For example, their path-based measure was adapted to make use of Wikipedia's structure of categories rather than WordNet's relations between synsets, and their text overlap-based measures were based on the text found in Wikipedia's articles, rather than WordNet's glosses. These combined measures provide a level of accuracy that is comparable to those derived from WordNet.

Gabrilovich and Markovitch [8] achieve extremely accurate results with a technique that is somewhat reminiscent of the vector space model widely used in information retrieval. Instead of comparing vectors of term weights to evaluate the similarity between queries and documents, they compare weighted vectors of the Wikipedia articles related to a particular term or portion of text. The weights of these articles—the strength of their association with the input text—are calculated using a centroid based document classifier. The result is a measure that approaches the accuracy of manual judgments. As well as offering much improved accuracy over WikiRelate!, ESA offers the ability to provide relatedness measures for any length of text; there is no restriction that the input be matched to an article title.

4 The Wikipedia Link Vector Model

The Wikipedia Link Vector Model (WLVM) extracts semantic relatedness measures for term pairs from the hyperlink structure of Wikipedia. To do so it must first identify the articles that might discuss the terms of interest. The most direct method for doing so is to obtain the article whose title matches the term directly, but this is complicated by the tendency for terms to have multiple meanings. Figure 1 illustrates the example of *plane*, an ambiguous term that might refer to a theoretical surface of infinite area and zero depth, or a tool for flattening wooden surfaces. In such cases the

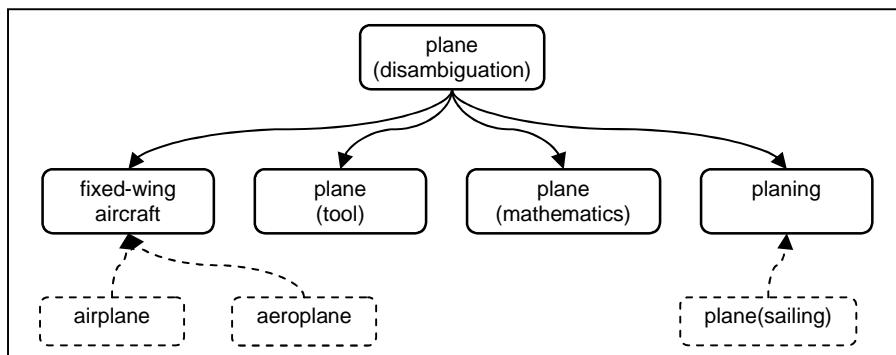


Figure 1: Candidate Wikipedia articles for the term *plane*

conventional solution for Wikipedia authors is to encase additional disambiguation information within parenthesis: the articles become *plane(mathematics)* and *plane(tool)* respectively. These brief scope notes can simply be stripped out when looking for relevant articles so that multiple items are returned for ambiguous terms. Wikipedia's contributors do not always follow this convention, however; *plane* can also refer to *planing*; the act of skimming over the surface of water, or to a *fixed-wing aircraft*. Sometimes this can be resolved by inspecting redirects; there exists a pseudo-article named *plane(sailing)* which redirects to the article *planing*. Other times one must consult disambiguation pages which list the various articles that a term might refer to; *fixed-wing aircraft*'s only redirects are *airplane* and *aeroplane*, but it is listed under the disambiguation page for *plane* as the most common sense of the term. Thus the full procedure used to obtain all Wikipedia articles relating to a term is to first list all pages whose titles (sans scope notes) match the term, and then process this so that:

- Articles are used directly.
- Redirects are followed so that their corresponding articles are used.
- Disambiguation pages are processed so that every article that they link to is used.

The next step in obtaining a similarity measure between two terms is to judge the similarity between their representative articles identified in the previous step. In this approach, the semantic similarity of two Wikipedia articles is defined by the angle between the vectors of the links found within them. This is similar to the vector space model used extensively within information retrieval to judge the similarity between documents and queries [9]. Rather than constructing vectors of term counts weighted by their probability of the term occurring (traditionally given by *tf-idf* measures), we build them using link counts weighted by the probability of each link occurring. This probability is defined by the total number of links to the target article over the total number of articles. Thus if t is the total number of articles within Wikipedia, then the weighted value w for the link $a \rightarrow b$ is:

$$w(a \rightarrow b) = |a \rightarrow b| \times \log \left(\sum_{x=1}^t \frac{t}{|x \rightarrow b|} \right) \quad (1)$$

In other words, the weight of a link within a source document is the number of times the source document contains that link (generally 0 or 1), multiplied by the inverse probability of any link to the target document. Thus links are considered less significant for judging the similarity between articles if many other articles also link to the same target; the fact that two articles both link to *science* is much less significant than if they both link to a specific topic such as *atmospheric thermodynamics*. With these weights defined for all n links $\{l_i / i=1..n\}$ found within a pair of articles x and y , the vector for each article is given by:

$$\begin{aligned} x &= (w(x \rightarrow l_1), w(x \rightarrow l_2), \dots, w(x \rightarrow l_n)) \\ y &= (w(y \rightarrow l_1), w(y \rightarrow l_2), \dots, w(y \rightarrow l_n)) \end{aligned} \quad (2)$$

Our similarity measure for the articles is then given by the angle between their vectors. This ranges from 0° if the articles contain identical lists of links, to 90° if there is no overlap between them. With these angles calculated for all possible mappings between the relevant articles for our two terms, the actual similarity between the terms is judged to be the lowest angle found between any pair of relevant articles. This is where articles are finally disambiguated, so that only the two articles that are most closely related to each other are used to form the final measure of similarity. Thus, the actual articles used for final calculation will differ if one is judging between *plane* and *jet*, or *plane* and *bezier curve*.

5 Evaluation

Our natural ability as humans to disambiguate topics and judge their relatedness can be considered the gold standard against which our technique should be compared. The WordSimilarity-353 dataset described in Section 3 was used for this purpose, and allows direct comparison with similar techniques. To the best of our knowledge, this is also the largest publicly available test set of its kind.

As an open source project, the entire content of Wikipedia is easily obtainable for studies such as this. It is made available in the form of database dumps that are released sporadically, from several days to several weeks apart. The version used in this evaluation was released on June 3, 2006. At this point Wikipedia contained approximately 2GB of compressed plain text, which explodes to 40 GB of compressed data if its full revision history is considered. Our technique only requires the link structure and basic statistics for articles, which can be obtained separately as about 500 MB (compressed). From this we identified over a million articles, which constitute the various concepts for which semantic relatedness judgments are available. These are highly inter-linked; each article links to an average of 26 others. A further million redirects doubles the available terminology, but this shrinks when scope notes are discarded from titles, due to the ambiguity problem described in the previous section. Thus the final vocabulary for which semantic relatedness judgments were available is a little under two million distinct terms.

Access to this data is facilitated by the Wikipedia Miner Toolkit,¹ which was developed by the author to allow rapid exploration of Wikipedia's link structure. It wraps a database representation of the encyclopedia with convenience classes for querying and browsing its structure. When mined according to the process described in Section 4, the 437 distinct terms contained within our test data set related to almost 5,000 Wikipedia concepts. This once again highlights the high degree of ambiguity involved; each term relates to an average of 11 articles, with a maximum of 118 articles for the highly ambiguous term *Jackson*.

The next step for this technique is to judge the similarity between the Wikipedia articles identified for each term. In order to evaluate the accuracy of this separately, we manually identified the correct articles for each term pair. The automatically generated similarity measures between these manually selected articles correlate

¹ <http://sourceforge.net/projects/wikipedia-miner/>

highly with the manual judgments of the terms they represent (according to spearman rank-order correlation coefficient). The correlation of 0.72 shown in Table 2 can be compared directly to the results described in Table 1; it is only slightly lower than the most accurate method ESA, and is a significant improvement over the remainder. Thus the angle between vectors of normalized link counts is a good measure of the semantic relatedness of Wikipedia articles.

The final step is to identify the semantic relatedness of each term pair by selecting the two articles with the highest semantic relatedness (or lowest vector angle). Unfortunately, accuracy degrades significantly when the algorithm is asked to disambiguate articles automatically in this way. As shown in Table 2, correlation with manual judgments drops to 0.45. This is understandable; disambiguation is inherently difficult when terms in each pair can only be disambiguated against each other. In Wikipedia there is the vast number of correct senses for many of the terms, which complicates matters. For example, *Arafat* and *Jackson* are almost completely divergent according to manual judgments, but are made moderately related when *Jackson* is disambiguated to *Jesse Jackson*, an American politician criticized for his anti-Semitic remarks.

Table 2: Performance of the *WLVM* measure for semantic relatedness

article selection (disambiguation)	correlation with manual judgments
manual	0.72
automatic	0.45

There is room to improvement, however. Under WLVM, dissimilar links degrade weightings as much as shared links increase them. Large articles are typically multi-faceted and have a greater chance of containing dissimilar links, and are consequently unfairly penalized. If anything, this bias should be reversed; when disambiguating manually, one tends towards the more common, easily recognized sense of a term—which typically corresponds to the larger article.

To evaluate the effect of this bias, we tested another measure: the sum of shared link weights. This is identical to WLVM except that the weights of shared links are added together and those of dissimilar links are discarded. Table 3 shows the effect of this simple modification: the accuracy of the measure between individual articles degrades significantly, and yet disambiguation improves. Thus the sum of shared link weights is a better measure of the semantic relatedness of terms, but WLVM is a much more accurate measure of the semantic relatedness between articles.

Table 3: Performance of the *sum of shared link weights* measure

article selection (disambiguation)	correlation with manual judgments
manual	0.59
automatic	0.52

6 Discussion and Conclusions

In this paper we proposed and evaluated a novel approach to computing semantic relatedness of terms with the aid of Wikipedia. Our approach is most similar to Explicit Semantic Analysis (ESA) and WikiRelate!, which also exploit the content of Wikipedia for this purpose. The central point of difference is that our technique uses only the skeleton structure of Wikipedia rather than its entire content. To obtain measures from either of the other techniques, one must obtain and pre-process a vast amount of textual data. By comparison this technique merely requires one to download a database of pages and links that is far smaller and already indexed. Unfortunately this comes at the cost of accuracy; our approach falls well behind ESA and only outperforms some of the measures provided by WikiRelate!.

There is room for improvement, however. We have identified a distinct bias in WLVM towards smaller, more obscure articles, which greatly degrades its ability to disambiguate terms. One simple modification resolves this to improve accuracy beyond all measures provided by WikiRelate!. Unfortunately it degrades the measure significantly when one considers the relatedness of articles rather than terms. Future work will be centered on resolving WLVM's disambiguation bias while maintaining or improving it as measure of relatedness between Wikipedia articles.

There are many possibilities to explore regarding this. Particularly promising is the vast number of other links found in Wikipedia that our measure does not yet consider. For example, we observed that the categories that articles descend from and the articles that link to them correlate quite highly with semantic relatedness even when un-weighted. With such possibilities left to be explored, it seems likely that this comparatively accurate measure of semantic relatedness can be further improved until it reaches the same level as ESA, while bypassing the need to process Wikipedia's extensive textual content.

References

1. Milne, D., Medelyan, O. and Witten, I. H. Mining Domain-Specific Thesauri from Wikipedia: A case study. Proc. of WI 2006
2. Finkelstein, L., Gabrilovich, Y.M., Rivlin, E., Solan, Z., Wolfman, G. and Ruppin, E. Placing search in context: The concept revisited. ACM TOIS 20(1), 2002.
3. Spearman, C. The Proof and Measurement of Association between Two Things. The American Journal of Psychology 100(3/4), 1987
4. Deerwester, S., Dumais, S., Furnas, G., Landauer, T. and Harshman, R. Indexing by latent semantitic analysis. JASIS 41(6), 1990.
5. Jarmasz, M. (2003) Roget's thesaurus as a lexical resource for natural language processing. Unpublished Master's thesis, University of Ottawa, 2003.
6. Budanitsky, A. and Hirst, G. Evaluating wordnet-based measures of lexical semantic relatedness. Computational Linguistics, 32(1), 2006
7. Strube, M. and Ponzetto, S.P. WikiRelate! Computing Semantic Relatedness Using Wikipedia. Proc.of AAAI 2006
8. Gabrilovich, E. and Markovitch, S. Computing semantic relatedness of words and texts inWikipedia-derived semantic space. Accepted for IJCAI 2007
9. Salton, G. and Wong, A. and Yang, C.S. A vector space model for automatic indexing. Communications of the ACM 18(11), 1975

X3D Software Visualisation

Craig Anslow¹, Stuart Marshall¹, James Noble¹, and Robert Biddle²

¹ School of Mathematics, Statistics, and Computer Science,
Victoria University of Wellington, New Zealand

² Human Oriented Technology Laboratory, Carleton University, Canada
`{craig, stuart, kjx}@mcs.vuw.ac.nz`

Abstract. We have a software visualisation architecture that requires tools to develop visualisations from XML execution traces and integrate the visualisations into user's web environments. Most existing web software visualisation systems create 2D visualisations and if they do use 3D they are using technologies that are outdated, not designed for the web, and hard to extend. We are building a tool that transforms XML execution traces into X3D – the Web3D Consortium's open standard for web 3D graphics – web enabled visualisations and exploring how suitable X3D is for use in software visualisation. Our tool and visualisations will help developers to understand the structure and behaviour of software for reuse, maintenance, re-engineering, and reverse engineering.

Key words: Software Engineering, Software Visualisation, Graphics, X3D

1 Introduction

Software visualisation is the use of the crafts of typography, graphic design, animation, and cinematography with modern human-computer interaction and computer graphics technology to facilitate both the human understanding and effective use of computer software [1]. In a recent survey [2] based on questionnaires completed by 111 researchers from software maintenance, re-engineering and reverse engineering, 80% found software visualisation either absolutely necessary or important (but not critical) for their work.

We have a visualisation architecture [3] for deploying software components over the web. The design supports components in multiple languages and configurations (e.g. complete programs or just code fragments). Our architecture requires tools to develop and deliver visualisations from XML execution traces.

Our research group have created a tool that can produce 2D Scalable Vector Graphics (SVG) visualisations over the web from our execution traces [4]. Ware [5] demonstrates that displaying object oriented software in three dimensions instead of two can make it easier for users to understand the data. We want to create 3D software visualisations for software reuse, maintenance, re-engineering, and reverse engineering. We have decided to explore how suitable X3D [6] – the Web3D Consortium's open standard for web 3D graphics – is for creating visualisations from our XML execution traces and use in software visualisation.

Identifying the advantages and disadvantages of 3D visualisation techniques is beyond the scope of this paper. The rest of this paper is organised as follows. In section 2 we describe our tool for visualising execution traces in 3D. In section 3 we describe what X3D is. We then show our X3D software visualisations in section 4. In section 5 we discuss related work and conclude our ideas in section 6.

2 VARE3D

We are developing a web-based software visualisation tool called VARE3D, based on our Visualisation Architecture for REuse (VARE) [3]. VARE3D currently transforms XML execution traces into X3D visualisations, see Figure 1. Users make queries from a web browser. Users can test drive software components by specifying a sequence of method invocation and field access/modifications and then executing the sequence on a component. The output of a test drive is an XML execution trace. Users can transform XML execution traces using XSLT³ into X3D visualisations. Users can then display X3D visualisations in a web browser.

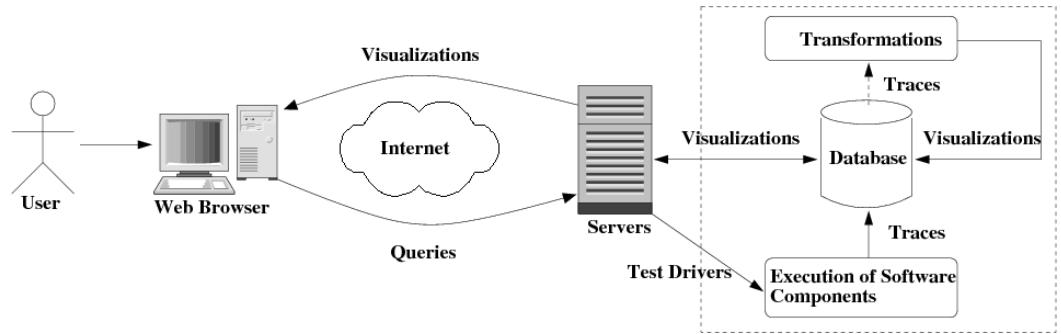


Fig. 1. VARE3D architecture.

The XML execution traces contain static and dynamic information of software components including the events that happened during the execution of a component. Separating the test driving and creation of visualisation steps allows users to test drive components and then create X3D visualisations in the future. Users can also view stored X3D visualisations without having to test drive remote software or transform XML execution traces.

³ Extensible Stylesheet Language Transformations (XSLT) is a language used for transforming XML documents into other XML documents.

3 What is X3D?

X3D [6] is an XML language for 3D content delivery on the web. X3D is the successor to the Virtual Reality Modeling Language (VRML97). X3D combines both geometry and runtime behaviour into a single XML file. X3D can be displayed in a native X3D browser or a web browser that has an X3D plug-in. X3D content can be created using purpose built X3D authoring tools, text editors, transformed from XML, and converted or exported from third party applications. X3D allows scripting in ECMAScript or Javascript.

The X3D runtime environment is the scene graph which is a directed, acyclic graph containing the objects represented as nodes and object relationships in the 3D world. Nodes within the scene graph can have descriptive fields and can contain one or more child nodes. X3D is organised as a set of components where each component is a set of related functionality. Profiles are built from components and are a standardised set of extensions to meet specific application needs.

Navigation in X3D is specified by the navigation info node which defines the navigation paradigms (walk, slide, examine, fly, and look-at), the speed at which a user can move, and the range of visibility within the 3D world. These options can also be dynamically changed in a web browser. A user can also navigate to predefined locations called viewpoints. A user can interact with nodes in X3D by clicking, dragging, or moving the mouse over them. Animation is controlled by time sensors, colour, coordinate, orientation, and position interpolators. Lighting components can be used to illuminate or hide objects depending upon their location and grouping. Collision and gravity can also be applied to give real world effects.

4 X3D Software Visualisations

We want to find out how effective X3D is for use in software visualisation. In particular we want to visualise the static structure and dynamic behaviour of software from execution traces. We now present a representative sample of software visualisation techniques including algorithm animation, source code related visualisations, and execution trace visualisations.

Algorithm Animation. Brown and Najork [1] identified several reasons for integrating 3D graphics into an algorithm animation system. Figure 2 shows how the third dimension can be used for expressing fundamental information about Dijkstra's Shortest Path Algorithm. The shortest path is the green edges (A-B and A-C-E-D-F), while explored edges are purple and unexplored edges are white. Columns are used to represent the cost of getting to each vertex. The length of a path is the sum of all the weights of the edges along a path.

The third dimension in this animation provides state information about the cost of vertices and weight of edges. Animation is used to show fundamental operations of the algorithm: lifting an edge represents addition, lowering a highlighted edge indicates the outcome of a comparison, and shortening a column shows assignment.

The columns and edges are implemented as indexed faced sets. A column has eight points while an edge has four points. When the cost to a vertex changes the top face of the column is increased or decreased in the z dimension by changing four coordinate points. When an edge changes only two coordinate points are changed. Coordinate points are moved using coordinate interpolators, while the colour of a column or an edge is controlled by colour interpolators. A time sensor is used to start and keep the animation going, which loops every 20 seconds. Start, stop, and pause buttons can also be added to give a user more control over the animation. Using indexed faced sets as opposed to boxes and cylinders allowed smoother transitions and required less calculation effort.

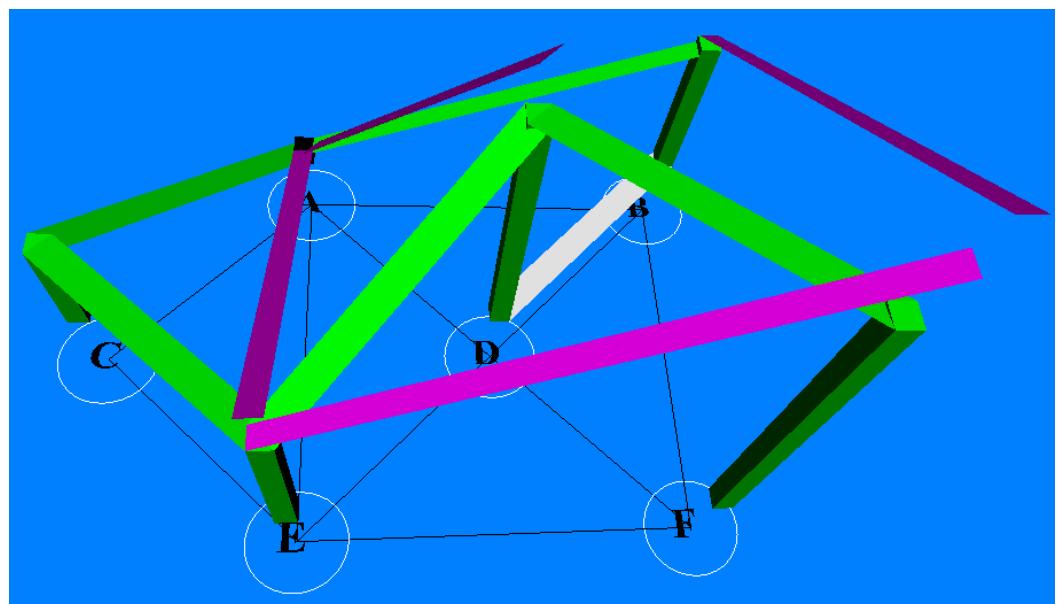


Fig. 2. Shortest Path Algorithm Animation

Source Code Related Visualisation. Figure 3 shows a class diagram of a C++ program represented as a node-link visualisation. The image has two displays, the left display shows the class diagram and the right display shows the original source code of the program which was used as the information for the visualisation. The source code is rendered in HTML. Classes are represented as spheres and the base class is represented as a cone. The two classes that inherit from the base class are coloured grey while the other classes are red. The cone is animated to change colour from blue to red to purple to green continuously every second. Larger purple cylinders represent inherited relationships from the base class while smaller white cylinders represent other class relationships.

A user can examine the visualisation by rotating the view which gives different view points of the class diagram to gain a greater understanding. A user can manipulate nodes by dragging them to show different parts of the visualisation. A user can click on a class which shines a light on the node in the visualisation and then some Javascript is executed that highlights the associated class declaration from the source code in the right display. In the visualisation the user has selected the class in the middle of the left display (grey sphere) which has highlighted the associated class declaration (highlighted yellow) in the source code (the Fox class).

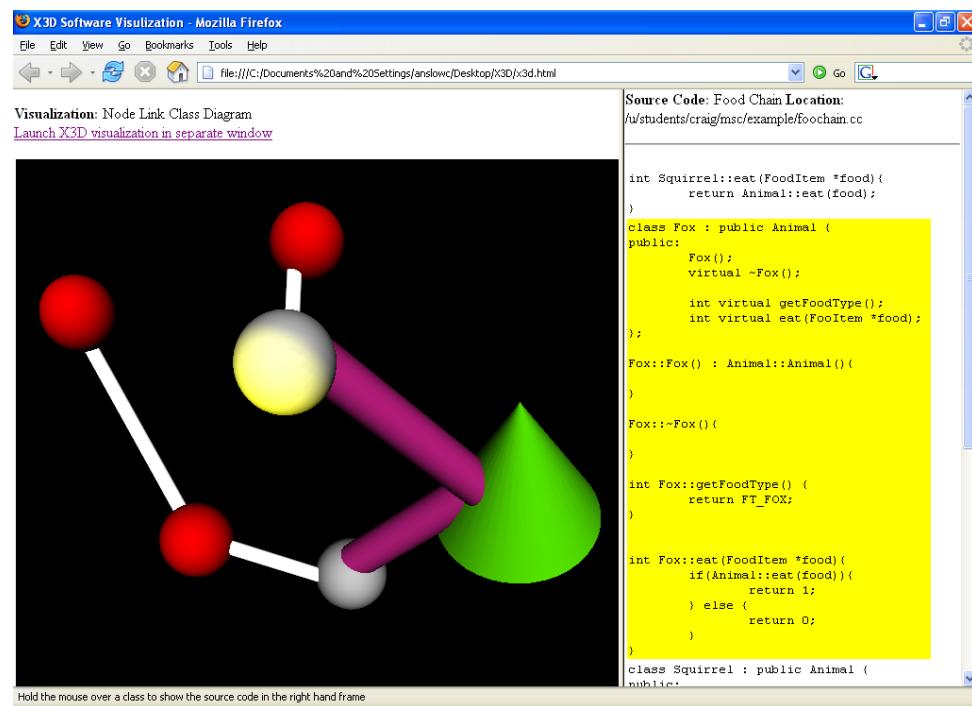


Fig. 3. Source code related visualisation.

Execution Trace Visualisations. Wiss et al. [7] found if it were possible to predict what structure the data will have for a visualisation then this will help determine what metaphor is most applicable for visualising large data structures. Wiss et al. [8] performed an empirical study on three 3D information visualisation designs. The results indicated that the subjects were significantly faster with the information landscape followed by the cam tree and then the information cube. We have customised VARE3D to allow a user to display alternate 3D information visualisation metaphors from the same execution trace. We next show our information landscape and information cube visualisations.

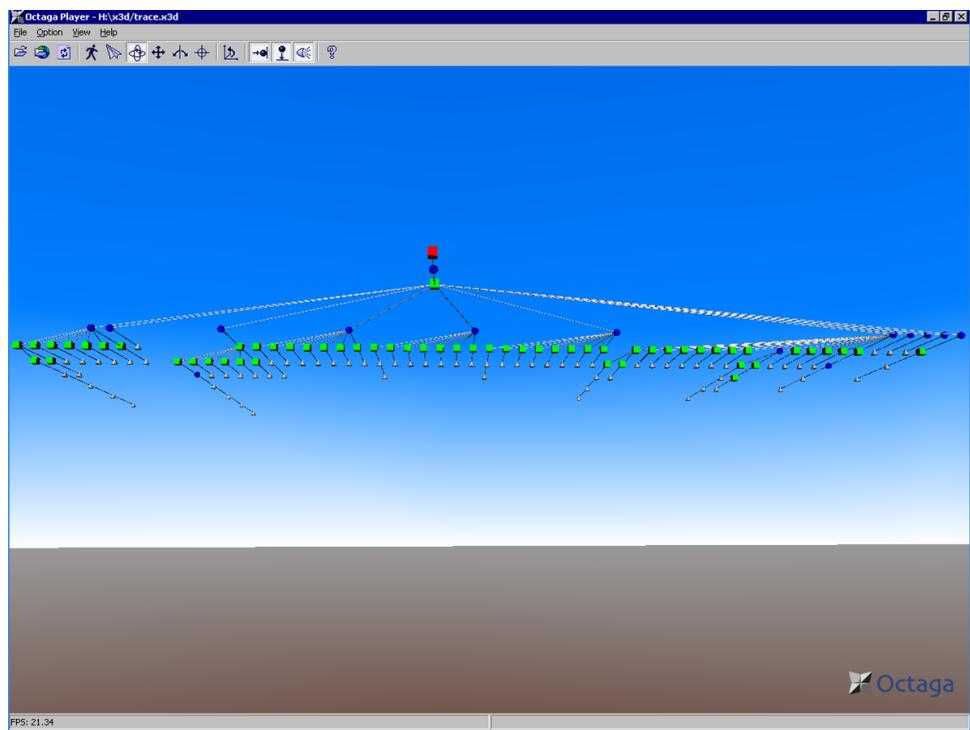


Fig. 4. Execution trace visualisation - information landscape.

Figure 4 shows the layout of all the events from an execution trace of a Java program as an information landscape. The information landscape is essentially 2 1/2 D rather than 3D. A red box represents the main class, blue spheres object creation, green boxes as method calls, white cones as method returns and end of the object. The image starts at the red box (the highest node in the image) and is animated from right to left.

Figure 5 shows the same information as Figure 4 but displayed as an information cube. The information starts with the main method outer red box, followed by the first object creation blue sphere at the top right of the cube then continuing along the links and down to each of the object creation blue spheres. The spheres are transparent and the events that an object executes are encompassed within the sphere.

Implementing the information landscape was more difficult than the information cube as it required more complex calculations for the links between nodes and the number of lines of code was greater. A linear layout approach was used however using formal graph layout algorithms should improve the creation and the actual visualisations. Finally transparency in the information cube was limited to only one level.

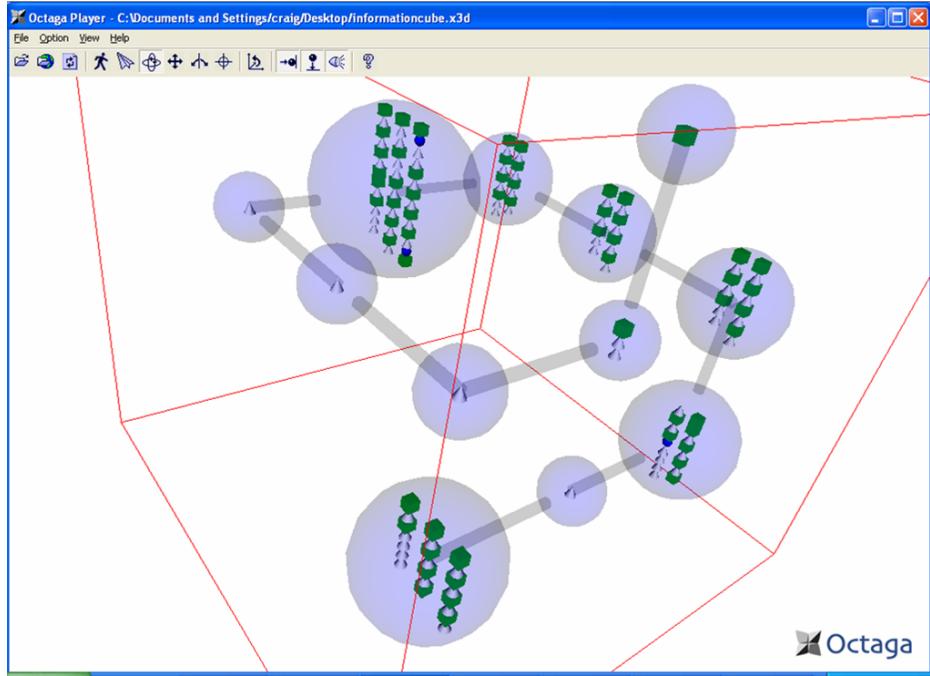


Fig. 5. Execution trace visualisation - information cube.

5 Related Work

We now discuss some related systems and tools that use 3D technologies for visualising software. Further information on software visualisation and program visualisation can be located elsewhere [1].

sv3D [9] is a framework for visualising source code and related attributes in 3D. sv3D is implemented using Qt for the user interface and Open Inventor for the rendering components. This tool only works as a stand-alone desktop tool and does not consider dynamic run-time information.

Churcher et al. [10] use VRML and XML for software visualisation. They visualise inheritance structures, class hierarchies, class cohesion, object-oriented metrics, and class clusters. Charters et al. [11] created a tool that also uses VRML and XML to display visualisations of software components as 3D cities. Feijjs and Jong [12] use VRML to visualise software architectures. However, none of these systems address dynamic information and both use the older VRML language.

UML visualisations have been explored using Java3D [13], VRML [14], and X3D [15]. However displaying UML in 3D does not scale well once there are many nodes in a world. Reading text in 3D once it has been rotated, repositioned rather than left to right, or partially obscured can make it hard for users to interpret what is being displayed.

6 Conclusion

We are building a tool that transforms our software XML execution traces into X3D visualisations. Our tool will help assist developers to understand the structure and behaviour of software for reuse, maintenance, re-engineering, and reverse engineering. We plan to implement other 3D information visualisation metaphors, create some more algorithm animations, provide visualisation filtering options, investigate the use of text in 3D, and integrate some test driving tools. We intend to conduct a comprehensive evaluation of X3D against some software visualisation frameworks and taxonomies to determine if X3D is applicable for use in software visualisation.

References

1. Stasko, J., Brown, M., Price, B.: *Software Visualization*. MIT Press (1998)
2. Koschke, R.: Software visualization for reverse engineering. In: Revised Lectures on Software Visualization, International Seminar. (2002)
3. Marshall, S., Jackson, K., Biddle, R., McGavin, M., Tempero, E., Duignan, M.: Visualising reusable software over the web. In: Proceedings of the Australasian Symposium on Information Visualisation. (2001)
4. Duignan, M., Biddle, R., Tempero, E.: Evaluating scalable vector graphics for use in software visualisation. In: Proceedings of the Australasian Symposium on Information Visualisation. (2003)
5. Ware, C.: *Information Visualization: Perception for Design*. Morgan Kaufmann (2004)
6. Web3D-Consortium: X3D specification (2004) <http://www.web3d.org/x3d/>.
7. Wiss, U., Carr, D., Jonsson, H.: Evaluating three-dimensional information visualisation designs: A case study of three designs. In: Proceedings of the International Conference on Information Visualisation. (1998)
8. Wiss, U., Carr, D.: An empirical study of task support in 3D information visualizations. In: Proceedings of the International Conference on Information Visualisation. (1999)
9. Marcus, A., Feng, L., Maletic, J.: 3D representations for software visualization. In: Proceedings of the ACM Symposium on Software Visualization. (2003)
10. Churcher, N., Keown, L., Irwin, W.: Virtual worlds for software visualisation. In: Proceedings of the Software Visualisation Workshop. (1999)
11. Charters, S., Knight, C., Thomas, N., Munro, M.: Visualisation for informed decision making; from code to components. In: Proceedings of the Conference on Software Engineering and Knowledge Engineering. (2002)
12. Feijs, L., de Jong, R.: 3D visualization of software architectures. Communications of the ACM (1998)
13. Dwyer, T.: Three dimensional UML using force directed layout. In: Proceedings of the Australasian Symposium on Information Visualisation. (2001)
14. Gogolla, M., Radfelder, O., Richters, M.: Towards three-dimensional animation of UML diagrams. In: Proceedings of the International Conference on The Unified Modeling Language. (1999)
15. McIntosh, P., Hamilton, M., van Schyndel, R.: X3D-UML: enabling advanced UML visualisation through X3D. In: Proceedings of the International Conference on 3D Web Technology. (2005)

Parallel Logit-Logistic Fuzzy Colour Constancy and Automatic Colour Contrast Rule Extraction

Vrushank D. Mehta¹ and Napoleon H. Reyes, Ph.D.²

Computer Science, Institute of Information and Mathematical Sciences

Massey University, Auckland, New Zealand

vrushankmehta@yahoo.com¹ and n.h.reyes@massey.ac.nz²

Abstract. Previously, the Logit-Logistic Fuzzy Colour Constancy (LLFCC) algorithm has been developed and tested to improve colour object classification significantly by compensating for the effects of variations in illumination conditions. However, the implementation was only achieved in serial, and requires tedious hand calibration to extract the colour contrast rules used for optimal colour classification. Furthermore, as colour calibration for object detection entails repetitive colour classification and tuning of the colour descriptors, running these algorithms in serial proves to be extremely slow. Hand-calibrating the rules involve trial and error runs, and can take up to 1 hr to complete. In light of these problems, this paper presents a novel parallel technique which dramatically improves the speed of the LLFCC algorithm, as well as automates the colour contrast rule extraction system.

Keywords: Computer vision, image processing, fuzzy colour processing, parallel vision system, colour object recognition, colour classification.

1 Introduction

Colour descriptors are considered to be one of the essentials for achieving object recognition. Among many other object properties, colour descriptors rise above the rest in achieving the object identification task as it serves as a necessary precursor to calculating size, shape, location and orientation [1], [2], [3]. However, common to these object identification algorithms is that they are slow due to their sequential nature of colour classification and object identification. Such identification and colour classification tasks are complex, requiring a lot of data movement. In particular, the complete colour calibration process is extremely difficult to execute in real-time. One plausible solution to the above problem is to use more than one processor to carry out the identification/classification task. Many algorithms have ventured this avenue. In [4] and [5] a parallel approach has been used to track objects in an image. The identification system carries out a series of image processing tasks such as segmentation, and adaptive thresholding to name a few. Parallelism is achieved by breaking up the image into as many parts as there are processors. The master node opens up the image and distributes parts of the image to several other processors. This, however, slows down the process as the master needs to

communicate respective image coordinates, and pixel values to each processor, before any parallel processing can take place. On the other hand, the Parallel LLFCC (pLLFCC) algorithm and the Automatic Parallel Rule Extraction System (ApRES) work differently in this regard. Parallelism in these algorithms is not dampened by image communication anymore, as only the results are communicated. All nodes open up the same image and start processing as soon as they open the image. This saves communication time, and also, no two nodes are dependant on one another (i.e. a slave node does not depend on the master node). Interestingly, using the proposed approach, we were able to classify colors in approximately 0.04 sec.

Another problem encountered in [1], [2], [3], [4], [5] is that they often use classical colour sets [8], where a pixel is strictly labeled as either a member or non-member of a certain group of pixels. This can lead to incorrect colour classification results, especially if the lighting conditions are not uniform throughout. This problem is largely due to hue and saturation drifting caused by changes in the illumination – a non-linear transformation of the sensed colour tri-stimulus in the colour space. This colour instability problem has been tackled by various algorithms using different techniques. In [1] an evolutionary algorithm (EA) working in the YUV colour space (which is most efficient in skin detection applications) has been used. The EA is used for an efficient colour classification which corresponds closely to a certain threshold value. It uses “1 + 1 Evolutionary Strategy” (OES) [1] as the evolutionary strategy and uses a fitness function to test the outcome at each step. The fitness function is defined as the ratio of correctly classified pixels to misclassified ones. However, the problem with this method is that approximately 25 steps need to be completed before a successful classification is achieved. The classical colour set problem is approached in [3] by using a set of parameters in HLS colour space to classify the colour of a pixel. The parameters examined are, a lower (L) value set to 0.5, an upper (S) value set to 1.0 and hue intervals that define similar colors. However, as the Saturation component is being disregarded (as it is a constant), this algorithm is likely to fail in distinguishing between colours that have the same Hue component, but different S components (e.g. yellow and light green).

Another promising approach to solving the classical colour set problem is to use Fuzzy Logic. Fuzzy set theory has made several inroads into the treatment of uncertainty in various aspects of image processing and computer vision [2]. The boundaries between colour sets are inherently ill-defined due to the ambiguities in the colour descriptors itself, making Fuzzy Logic an amenable solution to the problem. The pLLFCC and ApRES algorithm explained in this paper uses a fuzzy mechanism which is lenient towards pixel classification, which is to say that the mechanism gives a degree of membership to each pixel believed to be a member of a certain group of pixels, rather than asserting strict member or non-member values. Fuzzy colour contrast operators explained in [6] have been used to reduce the number of misclassifications and increase the number of correct classifications.

However, rule extraction using Fuzzy color contrast operators depends on hand calibration to find the most optimal solution. Hand calibration involves trial and error runs, and is thus very slow (approx. 1 hr. to get to an optimal solution). The automatic rule extraction algorithm we present in this paper uses a calibration technique which when executed in serial gives a result in approximately 3 seconds, and 0.0019 seconds in parallel using three processors.

2 General System Architecture

The classification algorithm is comprised of two parts: the ApRES and the pLLFCC. The system was developed on the Linux operating system with the Fedora Core Distribution version 5. Parallelism was achieved using the Sisters Cluster [7] provided by Massey University. The cluster consists of a main server which runs on a Dual P3 processor at 677MHz, 30GB SCSI disks and a memory of 512MB, eleven other nodes (or processors) running on a P3 processor at 677MHz and 256MB RAM, four other nodes running on a Celeron processor at 500MHz and 256MB RAM. The cluster is connected via a 16 port 3Com fast Ethernet switch [7]. The system was developed in a C programming language. Further, for the purposes of creating a GUI and a camera interface, OPENCV was used. Also, for enabling message passing between processors, the MPI (Message Passing Interface) library was used. All experiments were carried out in an enclosed room, with a ceiling-mounted camera. Lastly, the camera used was the AVT-Marlin F-033B firewire (1394a) camera, operating in 8-bit mode, RGB format with a frame rate of 60fps, ensuring high quality images.

3 The pLLFCC and the ApRES

3.1 rg-Chromaticity Colour Space and the Pie-Slice Technique

The rg-Chromaticity colour space is a normalized RGB space, which reduces the effects of brightness. In order to extend its colour classification potential, new colour descriptors introduced in [6] have been used. The new colour descriptors enable the separation of similar colours, and are called rg-Hue and rg-Saturation.

The pie-slice technique is used for colour classification, and is able to reduce the effects of glare and hue drifting on colour classification. The pie-slice decision region is characterized by rg-Hue (bounding angles) and rg-Saturation (radius) values in rg-chromaticity colour space.

Sample colour classification rule using the pie-slice approach, given that the pixel being examined is characterized by rg-Hue and rg-Saturation values:

$$\text{If } (\text{rg-Hue} \geq \theta_1 \text{ and rg-Hue} \leq \theta_2) \text{ and } (\text{rg-Sat} \geq r_{\min} \text{ and rg-Sat} \leq r_{\max}) \quad (1)$$

Then Colour is Pink.

3.2 Hits and Misses

Hits: If a pixel's angle and radius lies within the pie-slice decision region of the target colour, and the coordinates of that pixel lies within the bounds of the target object, then the pixel is considered to be a hit. **Miss:** If a pixel's angle and radius lies within

the pie-slice decision region of the target colour, and the coordinates of that pixel lies outside the bounds of the target object, then the pixel is considered to be a miss.
Score = Hits / Hits + Misses.

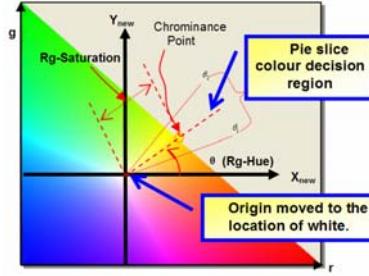


Fig. 1. Pie-slice decision region.

3.3 The ApRES

The automatic rule extraction system works as follows:

For each node:

```

Open the captured image.
Loop:
    Extract different colour contrast rule combinations of green and
    blue, while keeping the red component constant.
    Apply colour contrast rule combination to acquire the new RGB
    values.
    Transform new RGB values into rg-Hue and rg-Saturation.
    Perform pie-slice colour classification.
    Calculate the total number of hits and misses over all patches
    in the image.
    Store results in a structure (i.e. Rule combination, Hits and
    Misses).
Until end of image.

```

For the Master node:

```

Use the results stored in the structure to calculate the score for
each rule combination.
Determine the combination with the highest score.

```

3.4 The pLLFCC

Figure 3 illustrates the inner workings of the parallel LLFCC algorithm. Fig. 2 shows 2 processors (1 master, 1 slave); however, in practice, n number of processors can be used to carry out the classification task. Rx denotes no operation for the Red

component, while RE and RD stands for Red-Enhance and Red-Degrade respectively. The pseudo-code is shown below:

```

Step 1. Each node opens up the same image.
Step 2. Master Node:
Select primary patch.
Calculate minimum and maximum values for the angles and radii.
Send above values to slaves.
Step 3. Slave Node:
Receive data from master.
Scan part of the image based on the slave id
Apply rule extracted from ApRES.
Perform colour classification.
Send classified colours back to master.
Step 4. Master Node:
Receive data from slave(s)
Detect objects.

```

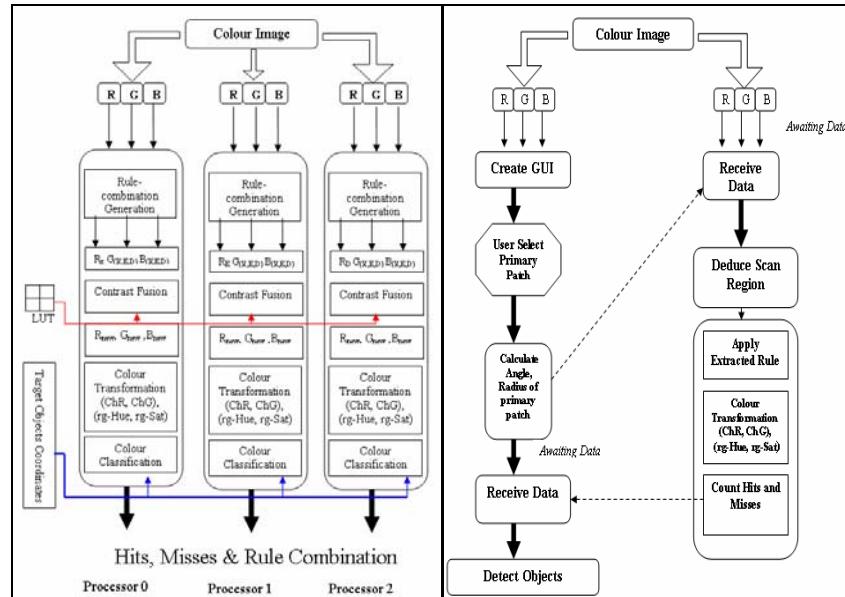


Fig. 2. The ApRES and the Parallel LLFCC.

4 Results and Analysis

In testing the pLLFCC and ApRES algorithms, we used a maximum of 7 nodes and 5 colour patches, with each patch containing either green, pink, blue or yellow for colour classification.

4.1 Speedup Factor in Colour Classification Using Varying Number of Nodes

The results shown in the table below were calculated using different configuration of the number of nodes used and the number of patches selected by the user. Each configuration was run 5 times, and then the average of those 5 readings was calculated, this average is the final result.

From the results showed in Table 1, it can be observed that the number of target colour patches does not really affect the speed of colour classification. Moreover, it can be seen that the configuration using 4 nodes returns the best result. Utilizing more than 4 nodes does not improve the classification speed anymore as it only increases the communication delay between the master node and the slave nodes until the communication time between the master and the slave nodes is more than the colour classification time.

Table 1. Speedup factor in colour classification using varying number of nodes and patches.

Number of Patches	3 Nodes	4 Nodes	5 Nodes	6 Nodes	7 Nodes
2	1.99676	2.98626	2.1834	2.0062	1.98
3	1.85437	3.0142	2.188	2.4108	2.248
4	1.9457	2.863	2.3908	2.3426	2.2765
5	1.996	3.059	2.1843	2.146	2.171
6	1.99676	2.98626	2.1834	2.0062	1.98
7	1.85437	3.0142	2.188	2.4108	2.248

4.2 Effects of Colour Contrast Rule Combinations to Colour Classification Accuracy

Shown below in Fig. 3 are the scores achieved by applying the various R, G, B colour contrast rule combinations (enhance, degrade or no operation) over the image.

A sample colour contrast operation combination is as follows:

If (rg-Hue depicts Pink) Then Enhance red, Degrade Green and No operation on blue. (2)

A closer look at the results in Fig. 3 show that a few combinations of colour contrast for R, G and B produce low scores. Thus these combinations can be excluded in further executions.

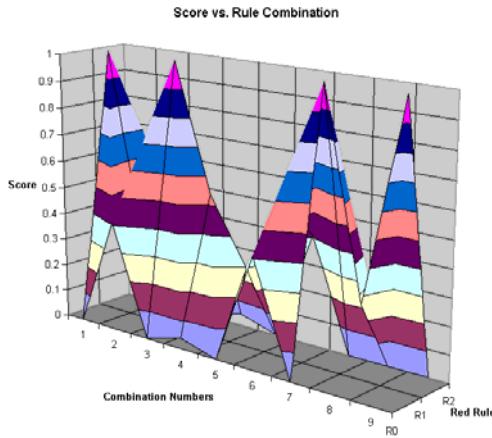


Fig. 3. RGB combinations and colour classification scores. R0 denotes red enhance, R1 denotes red degrade, R2 denotes no contrast operation. The combination numbers correspond to a specific Green and Blue colour contrast combination.

Table 2 compares the calibration speed when the calibration is done manually (i.e. by hand), using one processor in the automatic calibration system and using three processors in the automatic calibration system.

Table 2. Calibration Speed Comparisons.

Calibration Technique	Speed (sec.)
By hand	3600
Serially	3.5985
Using 3 Processors	0.001948

4 Conclusions and Future Work

A parallel colour calibration technique (pLLFCC) has been developed and tested for use in the robot soccer game. From the empirical results, it is evident that parallelism improves the speed of colour classification task to a good extent. By increasing the number of nodes from two to three, a 50% increase in speed is achieved, and on the average, a speedup factor of almost 45% is acquired.

The automatic parallel rule-extraction system (ApRES) developed achieves a much higher speed, with better accuracy as compared to manual colour contrast rule extraction. Furthermore, when the ApRES is executed using three processors, the speed achieved is greater than running it in serial.

As an extension to ApRES, we are currently in the process of incorporating an intelligent colour guidance system for colour rule extraction which would automatically filter out combinations that confer low scores.

References

1. Dahm, I., Deutsch, S., Hebbel, M., Osterhues, A., Robust Color Classification for Robot Soccer, 7th International Workshop on RoboCup 2003, Lecture Notes in Artificial Intelligence, Padova, Italy (2004)
2. Keller, J., Matsakis, P., Aspects of High Level Computer Vision Using Fuzzy Sets, FUZZ-IEEE 1999 International Conference on Fuzzy Systems, Seoul, Korea (1999) 847-852
3. Weiss, N., Hildebrand, L., An Exemplary Robot Soccer Vision System. CLAWAR/EURON Workshop on Robotics in Entertainment, Leisure and Hobby, Vienna, (2004)
4. Yu, J., Wang, S., Tan, M., A Parallel Algorithm for Visual Tracking of Multiple Free-Swimming Robot Fishes Based on Color information, International Conference on Robotics, Intelligent systems and Signal Processing, (2003)
5. Sha, H. Wah, C., Morphological image processing and its parallel implementation, ICSP, (1996) 539-542
6. Reyes, N., Dadios, E., Dynamic color object recognition using fuzzy logic, Journal of Advanced Computational Intelligence and Intelligent Informatics, Vol. 8, 1, (2004) 29-38
7. Sosna, G., Parallel Computing Facility, <http://iims.massey.ac.nz/research/sisters>
8. Hildebrand, L., Reusch, B., Fuzzy Color Processing, in Fuzzy Techniques in Image Processing, Studies in Fuzziness and Soft Computing, Vol. 52, Physica-Verlag, Heidelberg, (2000)

Intelligent Tutoring Systems Respecting Human Nature

Kon ZAKHAROV¹, Antonija MITROVIC¹, Lucy JOHNSTON²

¹Intelligent Computer Tutoring Group, ²Department of Psychology
University of Canterbury, Christchurch, New Zealand
{Konstantin.Zakharov, Tanja.Mitrovic, Lucy.Johnston}@canterbury.ac.nz

Abstract: The current level of development in Intelligent Tutoring Systems (ITS) ensures successful cognitive support. However, a number of studies suggest that learning outcomes are significantly influenced by a complex interaction between cognitive and affective states of learners. Little research has been done to investigate the effectiveness of learning with the help of affect-aware ITSs. Recently used approaches to affect recognition rely on facial feature tracking and physiological signal processing, but there is no clear winner among them because of the complexity and ambiguity associated with the task and the low-level data interpretation. The goal of our project is to develop a robust way of affect recognition for creating affect-aware pedagogical agents with the view to improve learners' engagement, motivation and learning outcomes.

1 Introduction

The semantic component of social interaction, most frequently expressed as speech, is often accompanied by the affective (otherwise known as emotional) component of social interaction; this is considered equally or sometimes even more important than the semantic component [1]. Although people are not always aware of how their language, posture, facial expression and eye gaze convey their emotions, these underpin people's interaction and navigation in the social world [2]. People also have a propensity to interact with computers in a social way, as if they were other people [3]. This undermines the idea that computers are merely neutral tools and emphasises the importance of the social relationships that can and will develop between a computer and a user. A better understanding of these relationships is essential for building of smarter tools for learning. For tasks fundamentally social in nature, a failure to include the emotional component in human-computer interaction (HCI) is potentially trimming the bandwidth of the communication channel. By ignoring human nature, computers force their users to act as if they too were machines; this contradicts the main assumption guiding HCI research [4]: "*People should not have to change radically to 'fit in with the system'; the system should be designed to match their requirements*".

The rest of this paper presents a Master of Science thesis project aimed at enabling an Intelligent Tutoring System (ITS) with affective awareness. Section 2 introduces prior research underlying our work. Section 3 outlines our approach and current state of our project. Section 4 concludes the paper with the plans of our further research efforts.

2 Background Research

Researchers have been grappling with the question of what appropriate behaviour is on the part of an interactive learning environment. Since etiquette is highly context-dependent, what may be appropriate in one situation, may be inappropriate in another. Frequently mentioned in HCI solutions to avoid a user being swamped by negative affect include either (a) trying to determine and fix the problem causing the negative feelings, and/or (b) pre-emptively trying to prevent the problem from happening in the first place [5]. Etiquette considerations in educational HCI are complicated by the fact that learning from a computer is not just about ease of use. Learning can be frustrating and difficult because it implies exposing learners' errors in thinking and gaps in their knowledge. Therefore, there are some fundamental differences between general HCI etiquette and etiquette for educational HCI. Cognitive psychology theory of learning from performance errors suggests that errors are an inseparable component of a learning process [6]. Error correction, in fact, has a critical significance for the improvement of future performance.

The foundation of our project is the research on applications of AI in Education. Learners vary in their amount of prior knowledge, motivation, learning style, natural pace and working memory capacity. Consequently, a uniform predefined instructional sequence can not provide the optimal learning environment for all learners. Educational research results published over two decades ago single out one-to-one tutoring as the most effective model of instruction [7]. Its success is based on the ability of human tutors to adjust their feedback based on their interaction with the learner. Real-life human tutors under one-to-one tutoring conditions are aware of the learner's cognitive state; this aspect of tutoring has been successfully captured by ITSs.

2.1 Intelligent Tutoring Systems

ITSs are task-oriented problem-solving environments designed for learning support in specific instructional domains. The capability of individualised instructions in ITSs hinges on the interaction of the target knowledge domain model, usually described as the expert module, and the model of the learner's knowledge, usually described as student model. Student models are maintained and updated by the ITS through interaction with the student; the ITS infers the relevant changes to the mental model of the target domain knowledge. There have been many approaches to student modelling including overlay models, perturbation models, models based on machine learning, and more recently, constraint-based models [8]. ITSs are known to improve learning performance by 0.3-1.0 standard deviations in a variety of target knowledge domains. For example, SQL-Tutor, an ITS for teaching Structured Query Language (SQL) for databases, improves performance by 0.65 standard deviations in just two hours of interaction with the system [9]. Atlas, a tutoring system for teaching Physics, improves performance by 0.9 standard deviations [10]. Between 20 and 25 hours of interaction with SHERLOCK, a tutor for technical troubleshooting in avionics, are equivalent to four years of on-the-job experience [11].

Constraint-based modelling (CBM), the student modelling approach we rely on in our research, arises from Ohlsson's theory of learning from performance errors [6]. A

CBM model represents domain knowledge as a set of explicit constraints on correct solutions in that domain [12]. At the same time, constraints implicitly represent all incorrect solutions. In this way, constraints partition all possible solutions into correct and incorrect solutions. There have been a number of constraint-based tutors developed within the Intelligent Computer Tutoring Group, at the University of Canterbury; one of the examples is EER-Tutor – a system for teaching the skill of Enhanced Entity-Relationship (EER) data modelling [13].

2.2 Affective and Physiological Processes

Emotions are described as psychological states or processes that function in the management of goals. An emotion is typically elicited by evaluating an event as relevant to a goal; it is positive when the goal is approaching and negative when progress towards the goal is impeded. Literature on emotion theory points out that negative affective states characterised by increased levels of adrenaline and other neurochemicals coursing through the body, diminish abilities with respect to attention, memory retention, learning, creative thinking, and polite social interaction [14].

There are two major theoretical approaches to the study of emotion: dimensional and categorical. Theorists who use the categorical approach to emotion attempt to define discrete categories or types of emotion [15]. Research in this area suggests that there are a number of basic emotions (estimates range from three to more than 20) which combine to produce all the emotional states which people experience. The dimensional approach conceptualises emotion as having two or perhaps three basic underlying dimensions along which the entire range of human emotions can be arranged [16]. The most common dimensions are valence (which ranges from happy to sad) and arousal (which ranges from calm to excited). Research using the dimensional approach has shown that emotions elicited by pictures, television, radio, computers and sounds can be mapped onto an emotional space created by the arousal and valence axes. Emotions involve multiple responses and thus it is common to group them into three broad categories: overt acts of behavioural sequences, emotional language and physiological reactions. In our research we focus our attention on the physiological reactions, characterised by changes in the somatic muscles (regulating voluntary movement) and in the viscera (internal bodily organs, like heart, liver or intestine). Physiological and behavioural reactions to affective stimuli significantly correlate with judgements of affective valence and/or arousal [17]. In respect to valence, there is a high dimensional correlation between valence reports and electromyographic activity of corrugator¹ and zygomatic² muscles. There is also significantly greater heart rate deceleration for unpleasant pictures, and relatively greater peak acceleration for pleasant materials. The same literature sources describe electro-dermal activity as a useful measure of arousal.

¹ The corrugator muscles are responsible for a lowering and contraction of the brows, a facial action to be an index of distress, associated with unpleasant affective stimuli.

² Activity of zygomatic muscle is involved in the smile response. Zygomatic activity increases for pleasant stimuli, is greatest for stimuli high in affective valence.

2.3 Affective Pedagogical Agents

Following in the vein of Topffer's law, "*All interfaces, however badly developed, have personality*" [2], recent research based on the Computers as Social Actors (CASA) paradigm explores ways of broadening the communication channel between people and computers with a view to improving the effectiveness of computer-based educational environments. This research relies on the use of affective pedagogical agents (APAs) which act as a medium for delivering feedback from a computer to its users [18]. APAs are anthropomorphic software characters capable of expressing human-like behaviours and emotions; they are known to enhance the social view/interaction with computers [19]. Pedagogical agents in instructional environments draw upon human-human social communication scripts by embodying observable human characteristics such as the use of gestures and facial expressions. Several studies show that animated agents improve students' learning, engagement and motivation [2, 18]. STEVE, ADELE and Cosmo and are just a few examples of pedagogical agents described in ITS literature. STEVE (Soar Training Expert for Virtual Environments) teaches students how to perform procedural tasks, such as operating or repairing complex devices [18, 20]. ADELE (Agent for Distance Education: Light Edition) is designed to support students solving exercises delivered over the World Wide Web [21]. In the application of a case-based clinical diagnosis, ADELE can highlight interesting aspects of the case, as well as monitor and provide feedback as the student works through a case. Cosmo inhabits the Internet Protocol Adviser, which is a learning environment for the domain of Internet packet routing [19]. He provides advice to learners as they decide how to ship packets through the network to the specified destination.

3 Our Approach and Project Status

The goal of our project is to develop an affect-aware pedagogical agent, integrate it with EER-Tutor and conduct an evaluation study. Our hypothesis is that the effectiveness of computer-mediated learning environments will improve from recognition of the affective state of their users. Incorporating analysis of affective state in the synthesis of feedback can elevate the interaction with the learner to a new level and make a difference not only in the learner's perception of the interaction, but in the learning outcomes as well.

3.1 Pedagogical Agent

We have developed a pedagogical agent and integrated it into EER-Tutor. During the persona design process, we tried to determine what kind of affective displays an affective agent should offer in a learning context in order to support the learner's determination in the face of the inevitable stress, anxiety and frustration involved in learning. One simple rule of thumb suggested by Bickmore [1] is to apply what has been found appropriate for human-to-human interaction (HHI) to the design of educational HCI. Klein et al. [14] identify two types of support for emotion regulation.

First, passive support is used to manipulate moods without necessarily addressing or discussing emotions themselves. Media, activities, food and other substances fall into this category; interactions with people can fall into either category. In contrast, active support occurs when people discuss or otherwise directly address their emotions as a means of managing them.

We aimed for the agent to be able to acknowledge the learner's emotions indirectly through its emotional appearance, while trying to keep the learner focused on the task at hand. Thus we designed an agent which would express solidarity with the user at all stages of the interaction – it will cheer with the users' success, be sympathetic with the user facing difficulties and keep company to the user in neutral situations. Similar agent behaviour was earlier adopted and implemented in the work of Lester et al [19].

While in our case, the agent does not yet have a way of determining users' affective state, prior research shows that in a learning context affective state can be indexed on the basis of cognitive state [22]. EER-Tutor maintains long and short-term student models; the state of the student model can be used to index the student's affective states. In our agents' design however, affective logic does not directly rely on the student model. Instead, the logic relies on session history, which includes the history of a wide variety of user actions.

The agents' implementation is split between the server-side and client-side of EER-Tutor. The server-side carries the agents' affective logic and controls the agents' behaviour, while on the client-side the agent appears to the users as a "talking head" with an upper body, embedded in EER-Tutor's work-space. The agent figures have been designed with the help of PeoplePutty toolkit; the web browser displays the agent with Haptek³ player plug-in. Haptek's character affective appearance is controlled by a number of parameters called switches. Some switches, for example, control lips and eyebrows positions. Haptek characters communicate with the server through AJAX requests. Figure 1 shows the EER-Tutor workspace with a male agent seen on the right-hand side above the feedback pane. The screenshot shows the state of the work-space immediately after a solution submission.

In July 2006, we conducted a formative study aimed at assessing learners' perception, expectations and response to the agents. The general response to the agents was positive – 75% rated the agents as a useful feature. At the same time, half of the participants who thought the agent's presence was unnecessary rated audible narration as useful. Overall, the participants were enthusiastic about narration – 50% stated that narration was the most helpful feature, because it made it possible for them to keep their eyes on the diagram and begin correcting errors while listening to the narration. Participants commented that this helped save time and enabled them solve problems faster.

3.2 Affect Measurement

In our research we have adopted the dimensional approach, since continuous nature of valence and arousal is more suitable for our method of measurement. In order to make EER-Tutor aware of the emotional state of its users we will use a set of sensors from

³ <http://www.haptek.com/> – Haptek Inc., PeoplePutty is a product of Haptek.

Thought Technology⁴ to capture the physiological data. We have acquired four sensors for reading physiological signals: blood volume pulse sensor, galvanic skin response sensor, surface electromyography sensor and respiration sensor. In order to convert the analogue signal from the sensors into digital signal we have acquired a data acquisition card (DAQ card) and implemented a prototype module for controlling the DAQ card and sending data to EER-Tutor server.

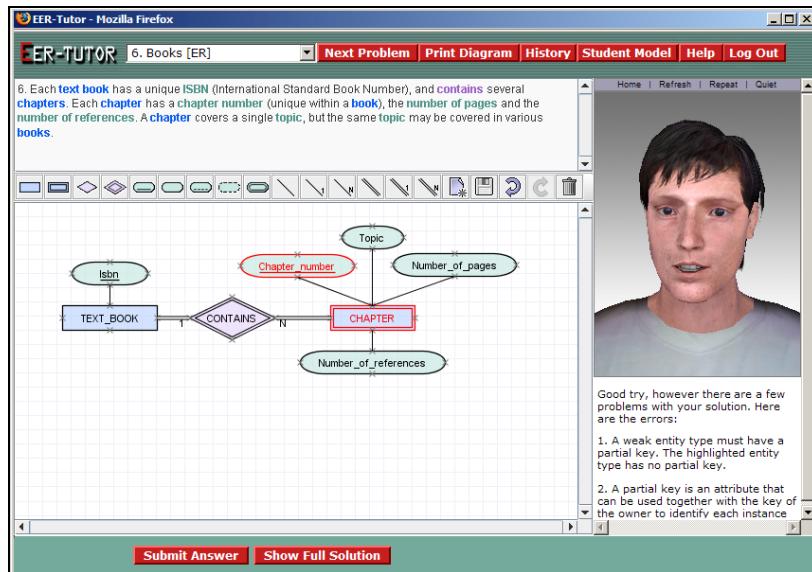


Figure 1. EER-Tutor work space with an agent.

3.3 Facial Feature Tracking

Literature on physiological data processing indicates that the EMG signal is not always reliable unless the top layer of dead skin cells is removed with abrasive cream before the application of the sensor electrodes. This can be perceived as quite intrusive by the experiment participants. As an alternative to EMG we have been working on a facial feature tracking application. Positive affective valence, for example, can be read as the distance between the corner of the mouth and the outer corner of an eye on one side of a person's face. Out of the numerous facial feature tracking approaches and implementation descriptions we have chosen a hybrid approach based on a Haar Classifier for face region detection [23] and feature extraction through common image processing techniques, such as edge detection, adaptive thresholding and integral projections [24]. In our implementation we used the Intel's OpenCV library⁵. Figure 2 shows a video stream frame with the important facial features detected with our prototype application: pupils, eyes and mouth corners. Next we will extend the application

⁴ <http://www.thoughttechnology.com> – Thought Technology Ltd.

⁵ <http://www.intel.com/technology/computing/opencv/> – Open Source Computer Vision Library.

to enable retrieval of eyebrow positions. Before the facial feature application is integrated into EER-Tutor, we will run a brief evaluation of our implementation.

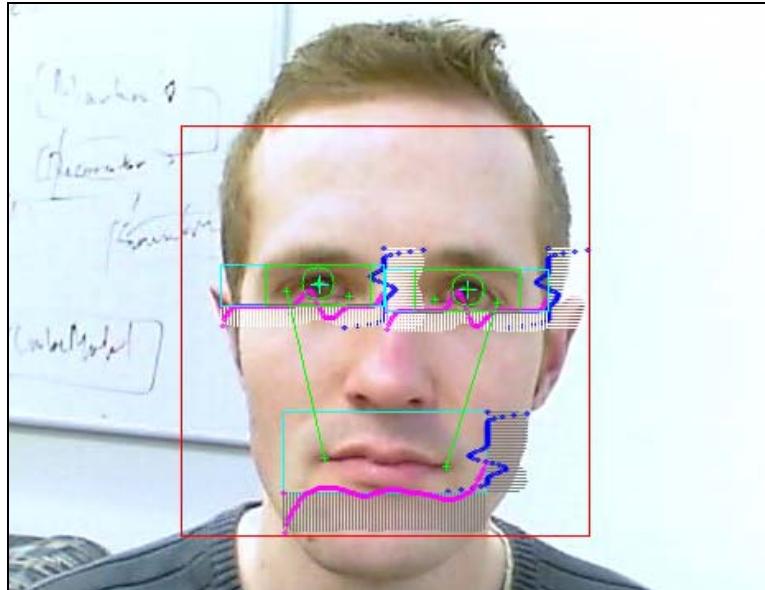


Figure 2. Retrieval of facial features with integral projections.

4 Future Work

In our future work, we will extend our system to identify students' affective states via real-time facial feature tracking and physiological sensors. Incorporating data from the sensory channel and changing the agents' persona rules will give the agents a finer level of affective awareness. This will be followed by a larger evaluation study to assess the agents' impact on users' view of interaction and learning with EER-Tutor.

References

1. Bickmore, T.W.: Unspoken Rules of Spoken Interaction. *Communications of the ACM* **47** (2004) 38-44
2. Mishra, P., Hershey, K.A.: Etiquette and the Design of Educational Technology. *Communications of the ACM* **47** (2004) 45-49
3. Reeves, B., Nass, C.: *The Media Equation. How People Treat Computers, Television, and New Media Like Real People and Places*. Cambridge Publications (1996)
4. Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S., Carey, T.: *Human-Computer Interaction*. Addison-Wesley, New York (1994)
5. Norman, D.: *The Psychology of Everyday Things* (a.k.a. *The Design of Everyday Things*). Basic Books, New York (1988)
6. Ohlsson, S.: Learning From Performance Errors. *Psychological Review* **103** (1996) 241-262

7. Bloom, B.S.: The 2 Sigma Problem: The Search for Method of Group Instruction as Effective as One-to-One Tutoring. *Educational Researcher* **13** (1984) 4-16
8. Ohlsson, S.: Constraint-Based Student Modelling. In: McCalla, G.I. (ed.): *Student Modelling: The Key to Individualized Knowledge-based Instruction*, Vol. 125. Springer-Verlag GmbH, Berlin (1994) 167-189
9. Mitrovic, A., Mayo, M., Suraweera, P., Martin, B.: Constraint-based Tutors: a Success Story. In: Ali, M. (ed.): *Proceedings of the 14th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, Vol. 2070. Springer-Verlag GmbH, Berlin (2001) 931-940
10. Freedman, R.: Atlas: A Plan Manager for Mixed-Initiative, Multimodal Dialogue. *International Journal of Artificial Intelligence in Education, Workshop on Mixed-Initiative Intelligence* (1999)
11. LaJoie, S.P., Lesgold, A.: Apprenticeship Training in the Workplace: Computer-coached Practice Environment as a New Form of Apprenticeship. *Machine-Mediated Learning* **3** (1989) 7-28
12. Mitrovic, A., Ohlsson, S.: Evaluation of a Constraint-based Tutor for a Database Language. *International Journal of Artificial Intelligence in Education* **10** (1999) 238-256
13. Zakharov, K., Mitrovic, A., Ohlsson, S.: Feedback Micro-engineering in EER-Tutor. In: Breuker, J. (ed.): *Proceedings of the 12th International Conference on Artificial Intelligence in Education*, Vol. 125. IOS Press (2005) 718-725
14. Klein, J., Moon, Y., Picard, R.W.: This Computer Responds to User Frustration: Theory, Design, and Results. *Interacting with Computers* **14** (2002) 119-140
15. Ortony, A., Turner, T.J.: What's Basic About Basic Emotions? *Psychological Review* **97** (1990)
16. Lang, A., Dhillon, P., Dong, Q.: Arousal, Emotion, and Memory for Television Messages. *Journal of Broadcasting and Electronic Media* **38** (1995) 1-15
17. Bradley, M.M., Lang, P.J.: Measuring Emotion: Behavior, Feeling and Physiology. In: Nadel, L. (ed.): *Cognitive Neuroscience of Emotion*. Oxford University Press, New York (2000) 242-276
18. Johnson, W.L., Rickel, J.W., Lester, J.C.: Animated Pedagogical Agents: Face-to-Face Interaction in Interactive Learning Environments. *International Journal of Artificial Intelligence in Education* **11** (2000) 47-78
19. Lester, J.C., Towns, S.G., Fitzgerald, P.J.: Achieving Affective Impact: Visual Emotive Communication in Lifelike Pedagogical Agents. *International Journal of Artificial Intelligence in Education* **10** (1999) 278-291
20. Rickel, J., Johnson, W.L.: Intelligent Tutoring in Virtual Reality: A Preliminary Report. *Proceedings of the 8th International Conference on Artificial Intelligence in Education*. IOS Press (1997) 294-301
21. Shaw, E., Ganeshan, R., Johnson, W.L., Millar, D.: Building a Case for Agent-assisted Learning as a Catalyst for Curriculum Reform in Medical Education. *Proceedings of the 9th International Conference on Artificial Intelligence in Education*. IOS Press (1999) 509-516
22. Conati, C., McLaren, H.: Evaluating a Probabilistic Model of Student Affect.: Proceedings of 7th International Conference on Intelligent Tutoring Systems, Vol. 3220/2004. Springer, Brazil (2004) 55-66
23. Viola, P.A., Jones, M.J.: Rapid Object Detection using a Boosted Cascade of Simple Features. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 1. IEEE Computer Society, Kauai, HI, USA (2001) 511-518
24. Mateos, G.G.: Refining Face Tracking with Integral Projections. *Lecture Notes In Computer Science* **2688** (2003) 360-368

Exploring Better Techniques for Diagram Recognition

Rachel Patel¹, Beryl Plimmer¹, John Grundy^{1, 2}, Ross Ihaka³

¹Department of Computer Science

²Department of Electrical and Computer Engineering

³Department of Statistics

University of Auckland

Private Bag 92019, Auckland, New Zealand

{rpat088@ec, beryl@cs, john-g@cs, ihaka@stat}.auckland.ac.nz

Abstract. A critical component of diagramming sketch tools is their ability to reliably recognise hand-drawn diagram components. This is made difficult by the presence of both geometric shapes and characters in diagrams. The goal of our research is to improve sketch recognition by improving the accuracy in grouping and classifying strokes in a diagram into text characters and shapes. We have done this by identifying the most significant features of strokes that can be used to distinguish shapes from text using a decision tree based partitioning technique. Implementation and evaluation of this new “shape divider” using these features against InkKit’s existing divider and the Microsoft divider has shown that our divider is more accurate at dividing text and shape strokes and can therefore improve overall sketch recognition.

Keywords: Human Computer Interaction, Pattern Recognition

1. Introduction

Computers can be used to produce formal diagrams e.g. software designs, user interface designs, CAD specifications, hierarchy charts and so on. However, in the beginning of a project it can be better to use pen and paper. The reason is that these simple, human-centric design tools are far more flexible which encourages creativity and in turn this produces better designs in the end. However a computer offers easy editing and distribution features and greater formality to the look of a diagram and so sometimes the important pen and paper design stage is overlooked.

Imagine being able to sketch diagrams directly onto a computer screen and have them accurately translated into formalized representations. The Tablet PC supports sketching of characters and recognition of these but has much weaker support for general diagram drawing and accurate recognition. InkKit [8] is a sketch tool for the Tablet PC that has been designed to bridge the gap between pen and paper and computers. InkKit allows you to sketch any type of diagram using the Tablet PC’s stylus. Its recognition engine then identifies each component of the diagram and transforms it into a formal, tidy version in a specified format such as HTML or a Word drawing object. InkKit has been used as a foundation for this research.

2. Motivation

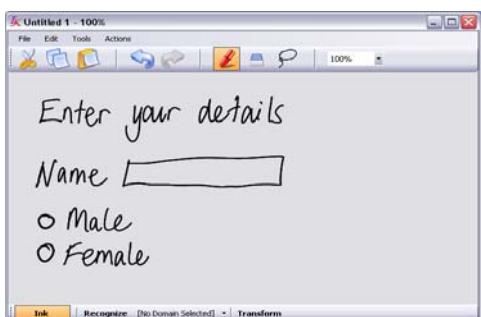


Fig 1. Sketch of a user interface design

A critical part of diagramming sketch tools is being able to reliably recognise the hand-drawn diagram components. The purpose of this Masters project is to improve the recognition algorithms that currently exist.

One of the reasons why recognising a diagram is difficult, is because we are dealing with words and shapes at the same time. Figure 1, shows a typical user interface design that can be used to highlight this problem. For example,

what is it that makes the circles in front of the words “Male” and “Female” circles indicating radio buttons and not the letter o?

The existing InkKit system automatically divides the ink into words and shapes, and then recognises the words using the Tablet Operating System recognizer and shapes using a variant of Rubine’s algorithm [5]. Once the basic shapes and words have been recognised they are combined back together to suggest the most probable diagram component. This method of dividing the words and shapes is preferred over using a modal interface to separate the two as it preserves human’s natural sketching approaches used on pen and paper [3]. The aim of our research has been to greatly improve the algorithm that divides the ink into either text or drawing as this is the area where we can expect the most improvement in diagram recognition [8].

Most sketch recognition mechanisms measure various features of the strokes in a sketch to guide its recognition. This is also evident in the recognition engine of Ink-Kit. However what is lacking is a definitive, principled set of the most significant features that can be used to provide an accurate division of the shape and text strokes in a sketch. Therefore our goal is to find these features in order to improve division of shape and text strokes and in doing so improve the accuracy of sketch recognition as a whole.

We first detail the methodology we have used to identify the most significant features of sketches that will aid division of strokes. Section 4 will briefly describe the selected feature set and then detail the experiment conducted to identify which features from the set are significant. The implementation and evaluation of the resulting divider of shape and text strokes is presented in Section 5. Finally we conclude with a discussion of the results and suggestions for future work.

3. Methodology

In this section we provide an overview of the approach we have used beginning with the investigation of possible sketch features, how feature data is collected and analysed, and the implementation and evaluation of the resulting divider of text and shape strokes.

Feature Discovery

Our first step was to identify all the possible features that could be useful in distinguishing between text and shape strokes in a sketched diagram. The origin of these features were from (1) related work in sketch recognition; (2) stroke features we felt may be useful in classifying strokes; (3) and stroke features from newly available hardware. Many stroke features have been documented by past work to be significant to solving various sketch recognition problems [2, 5, 6], and several of these features we have already implemented in InkKit's existing recognition engine [8]. Therefore it was in our interest to include as many of these features as possible in our investigation. Newly available hardware allows us to consider features that have not been widely studied before such as pen pressure and tilt.

Data Collection

We wanted to determine which features are actually the most significant to use when dividing text and shape strokes. This enables us to sample and use only those features when classifying strokes in InkKit's divider and not the myriad of other features that have little or no effect on stroke classification. To be able to test which are important features and which aren't we collected measurements of each feature from sketches and compiled these measurements into a dataset.

We identified a set of diagrams to be used for our experiment that displayed a wide range of common characteristics of diagrams. This allowed us to sample our stroke feature set on a realistic group of diagrams so that we are more likely to achieve a more accurate identification of significant features.

The next step was to get people to sketch examples of this set of diagrams using InkKit on Tablet PC's. We needed to collect diagram sets from as many people as possible to avoid bias in our samples of individual variation that may occur in their sketching. Once these diagrams were collected we processed them by calculating all stroke features identified from each stroke of each diagram. The resulting data was then collated into a dataset ready for statistical analysis.

Analysis

Once all the sketches were collected and processed into a dataset they were analysed to determine the features of strokes that are significant and should be used to divide text and shape strokes. We wanted to use a formal technique for this analysis so that we would gain a clear, accurate and principled view of the degree of significance each feature has in distinguishing between shapes and text in a hand-drawn diagram.

The use of trees has become a common and effective way to assist in decision making problems, our decision being whether to classify a stroke in a diagram sketch as a text or shape stroke. A tree-based partitioning technique was used to analyse the dataset and construct a classification tree [1, 7]. A classification tree has decision variables at each node, which would correspond to the most significant features found, and a classification label at each leaf, which would be either text or shape in our case (see Figure 2). Employing this technique allowed us to clearly identify significant fea-

tures to help division, and also provided us with the most optimal combination of features for implementation.

Implementation & Evaluation

Once the significant stroke features had been identified through formal statistical analysis a new shape divider was implemented for InkKit. InkKit's existing divider remains as we wanted to compare its performance and the Tablet PC Microsoft divider to our new shape divider. We evaluated these three divider implementations to determine which is more accurate at dividing text and shape strokes.

4. Experiment

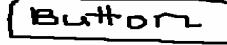
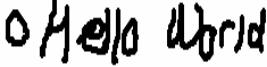
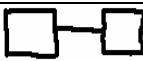
All the possible stroke features that may be helpful in distinguishing text from shape strokes were compiled together into a feature set. They came from related work in sketch recognition [2, 5, 6, 8]; features we thought may be significant; and the discovery of features from newly available Tablet PC hardware. We attempted to include as many features of strokes as possible to ensure that all avenues would be explored by the statistical analysis and the most significant features could be identified – 52 features were selected in all. The 52 features can be grouped into seven categories, size, time, intersections, curvature, pressure, Tablet Operating System recognition values and others.

Our stroke feature set was then analysed to discover which of these features provide the greatest contribution to dividing text and shape strokes in a diagram. We describe this experiment beginning with the collection and processing of data from a range of sketched diagrams, then analysing this data using a tree-based partitioning technique that consequently constructs a classification tree containing the most significant features for division of text and shape strokes. This process allowed us to draw conclusions as to which features are most significant to division.

Data Collection & Processing

To perform a full analysis of the feature set sketched diagrams were collected from as many people as possible. A set of nine diagrams were identified to be used for analysis, illustrated in Table 1. In compiling this set we looked for examples of shapes and text that would represent those typical of a wide range of diagram types and therefore would allow us to identify the most significant features of strokes for division to be identified for a general-purpose, reusable shape divider. Our diagram set includes basic shapes and text, complex shapes, composite shapes and various combinations and ordering of shapes and text. Sketches were gathered from 26 people. Each person completed a set of 9 sketches. Each sketch was then processed to obtain the 52 features from our feature set, forming a final dataset ready for statistical analysis with 1519 observations in all. We manually categorized each stroke as SHAPE or TEXT as base data for the statistical analysis.

Table 1. The Diagram Set

Shape Description	Example Sketch
<i>Circle</i>	
<i>Button</i> : rectangle with label “Button” inside it.	
<i>Text</i> : “Hello how are you”, without punctuation.	
<i>Radio-text</i> : radio button with label “Hello world” next to it. <i>Text-radio</i> : same as radio-text above but label written before radio button (spatial ordering is the same).	
<i>Combo box</i> : rectangle with a triangle inside.	
<i>Resistor</i> : spiked line, from electrical diagrams.	
<i>Hexagon</i> : six-sided polygon.	
<i>Connector</i> : two rectangles with a line connecting them in the middle.	

Analysis

The analysis of the dataset was performed using the R statistical package, pre-release Version 2.5.0 [9]. The dataset was used as training data for the rpart function [7] which applies a tree-based partitioning technique to identify the significant features to use as decision variables in a binary classification tree and determines how to split those features so that they can accurately classify strokes as either text or shape strokes. For each feature rpart is provided with each example stroke’s feature data e.g. length, average speed, curvature, average pressure value etc, and the known classification of the stroke i.e. SHAPE or TEXT. From the total training set the rpart function constructs a binary classification tree starting with the most significant feature, then the next, then the next and so on.

Results

The binary classification tree resulting from our rpart analysis of our full training set is shown in Figure 2. Eight different features of strokes, named in each node of the classification tree, were identified from the feature set as being significant for dividing shape from text strokes, each one is described in Table 2. The majority of these features are measuring some element of size, as five out of eight of the features come from this category. Features of time have also registered their importance as making up two out of eight of the significant features. Also one feature of curvature has been identified as important to the division of shape and text strokes.

This resultant decision tree is used to classify strokes into SHAPE or TEXT. For example, consider classifying a given stroke. First we sample its bounding box width. If its bounding box width is ≥ 1848 we follow the left branch of the tree. We then measure its total angle. If its total angle is < 10.1 the left branch is taken once again and the stroke is classified as a SHAPE. Taking another stroke, we calculate its

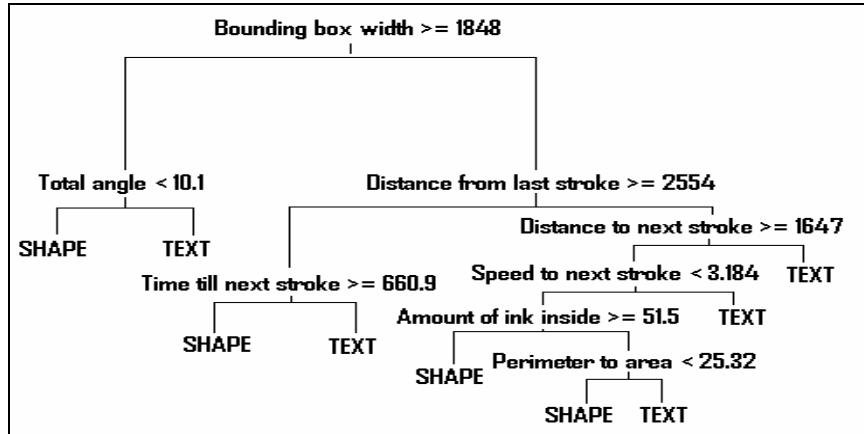


Fig 2. Classification tree for text and shape divider

Table 2. Description of the significant features identified in the classification tree

Feature	Description
Bounding box width	Width of the bounding box of the stroke.
Distance from last stroke	Distance the pen travels between the current stroke and the previous stroke.
Distance to next stroke	Distance the pen travels between the current stroke and the next stroke.
Amount of ink inside	Amount of ink inside the strokes bounding box.
Perimeter to area	Ratio of perimeter to area of the strokes convex hull.
Time till next stroke	The time between the current stroke and the next stroke in the sketch.
Speed to next stroke	Speed (distance/time) between the current stroke and the next stroke in the sketch.
Total angle	Total angle traversed by the stroke.

bounding box width and if this is < 1848 we follow the right hand side branch. We then calculate the strokes distance from the last stroke, and if this is < 2554 we again follow the right hand side branch. We then calculate its distance to the next stroke, and if this is < 1647 , we classify the stroke as TEXT.

5. Implementation and Evaluation

A new shape divider based on the classification tree from Figure 2 was implemented in InkKit, as an alternative to InkKit's existing divider and the Microsoft Tablet PC divider. An evaluation was carried out comparing the three dividers performance using our example diagrams to establish which would be able to divide text and shape strokes most accurately, i.e. with the lowest misclassification rate, where the misclassification rate is a measure of the proportion of strokes that were incorrectly classified.

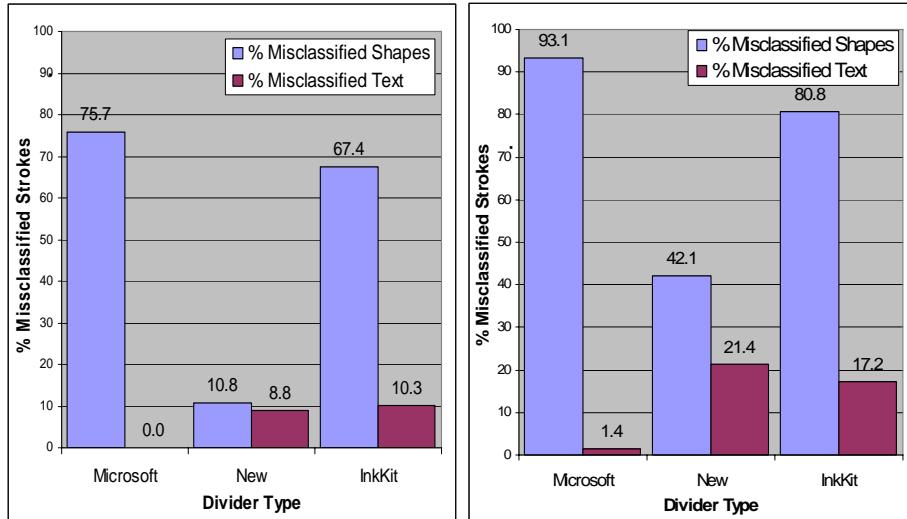


Fig 3. Percentage of misclassified shape and text strokes for each divider using the training diagram set

Fig 4. Percentage of misclassified shape and text strokes for each divider using the new diagram set

The dividers were used to divide the original training set of diagrams shown in Table 1 and also a new set of diagrams that included more complex diagrams exhibiting characteristics not considered previously. The new diagram set consisted of a directed graph, musical notes, check boxes and a crossed out word. Results were compiled based on the proportion of strokes that were correctly or incorrectly classified as text or shape strokes. The level of accuracy of each divider was then assessed from these results to ultimately determine if our new shape divider did in fact perform better.

Using the training set of diagrams, the percentage of shape and text strokes that were misclassified for each divider is shown in Figure 3. The Microsoft divider has the highest percentage of misclassified shape strokes at 75.7% and the lowest percentage of misclassified text strokes, where no text strokes were incorrectly classified at all. The new divider has the lowest proportion of misclassified shape strokes when compared with the other dividers at 10.8%, and the second lowest proportion of misclassified text strokes at 8.8%. The InkKit divider has a very high misclassification rate for shape strokes at 67.4%, coming in as the second highest of all dividers, however in contrast it has a low percentage of misclassified text strokes at 10.3%, however when compared with the other dividers, InkKit's rate of misclassification for text strokes is the highest. All dividers showed much greater accuracy in classifying text strokes than shape strokes.

Using the new diagram set the percentage of misclassified shape and text strokes for each of the three dividers are shown in Figure 4. The Microsoft divider once again has the worst rate of misclassification of shape strokes where 93.1% were incorrectly classified. It has the best percentage of misclassified text strokes at 1.4%. This follows the pattern shown in the evaluation results for the training diagram set shown in Figure 3. Also following the results of the first evaluation, our new divider has the lowest misclassification rate for shape strokes at 42.1%, although this is still very

high. The new divider has the highest percentage of misclassified text strokes at 21.4% however this is only a little above InkKit at 17.2% for text strokes. InkKit's rate of misclassification for shape strokes comes in at 80.8%. Again, all dividers show a greater degree of accuracy in classifying text strokes than shape strokes.

6. Conclusion

Eight features out of the 52 that were in the feature set were found to be significant to dividing text and shape strokes. These features were mostly measures of size and time, with one measuring curvature. The evaluation shows that overall the new shape divider was the most accurate at dividing the training set of diagrams when compared with the InkKit divider and the Microsoft divider. For the new diagram set it was considerably better at classifying shape strokes and marginally the worst at classifying text strokes. Overall, when compared with the InkKit and Microsoft dividers, the new divider was more accurate at dividing the new diagram set as well. Therefore we can conclude that the features selected can be used to improve the accuracy of division of text and shape strokes in a sketched diagram.

7. Future Work

The classification tree approach has worked well at combining the significant features together for more accurate division of strokes into character text and geometric shapes. Future work in this area could involve experimenting with other algorithms such as using Hidden Markov Models [4] for a more robust divider.

References

1. Breiman, L., et al., *Classification and Regression Trees*. 1984, New York: Chapman & Hall / CRC Press.
2. Fonseca, M.J., C.e. Pimentel, and J.A. Jorge. *CALI: An Online Scribble Recogniser for Calligraphic Interfaces*. in *AAAI Spring Symposium on Sketch Understanding*. 2002: IEEE.
3. Plimmer, B., *Using Shared Displays to Support Group Designs; A Study of the Use of Informal User Interface Designs when Learning to Program*, in *Computer Science*. 2004, University of Waikato.
4. Rabiner, L.R., *A tutorial on hidden Markov models and selected applications in speech recognition*, in *Readings in speech recognition*. 1990, Morgan Kaufmann Publishers Inc.
5. Rubine, D.H. *Specifying gestures by example*. in *Proceedings of Siggraph '91*. 1991: ACM.
6. Sezgin, T.M., T. Stahovich, and R. Davis. *Sketch based interfaces: early processing for sketch understanding*. in *Proceedings of the 2001 workshop on Perceptive user interfaces*. 2001, Orlando, Florida: ACM Press.
7. Venables, W.N. and B.D. Ripley, *Modern Applied Statistics with S*. Forth ed. 2002, New York: Springer.
8. Young, M., *InkKit: The Back End of the Generic Design Transformation Tool* 2005, University of Auckland.
9. R Development Core Team. *R: A Language and Environment for Statistical Computing*. 2006, Vienna, Austria, R Foundation for Statistical Computing.

Increasing the Expressiveness of Subscriptions and Advertisements in Distributed Content-Based Publish/Subscribe Systems

Sven Bittner*

Department of Computer Science
The University of Waikato, New Zealand
`{s.bittner}@cs.waikato.ac.nz`

Abstract. This paper gives a synopsis of the author's PhD project. In order to address a broad range of conference participants, we have specifically decided to give a general, rather non-technical outline of the tackled research problems. This decision is reflected in the overall structure of this paper.

After introducing the general research area and the context of the project, we outline the three research problems that are addressed within the associated dissertation. We then give a conceptional overview about the research that has already been carried out but also about the remaining steps that need to be undertaken. We conclude this paper by stepping back from the actual dissertation and relating its contributions to the broader research community.

1 Introduction

Within the last years, there has been an increasing academic interest in the publish/subscribe (pub/sub) communication paradigm. One reason for the growing popularity of this subject is the wide applicability of the general pub/sub approach, ranging from the low-level monitoring of distributed systems to high-level electronic commerce areas [11]. Another reason for its success is the asynchronous, loosely-coupled nature of pub/sub, allowing for the implementation of dynamic, event-driven, and thus independent application components.

The overall idea of pub/sub is straightforward and as follows: Pub/sub systems have two kinds of users, publishers and subscribers. *Publishers* provide information to the system, using *event messages*. Before this information can be published, publishers have to specify their future messages by the help of *advertisements*. *Subscribers*, conversely, are interested in information and specify their interests using *subscriptions*. The pub/sub system decouples the communication of these users and *notifies* subscribers whenever an incoming message *matches* their subscriptions. We have given an overview of these two user groups and their communication directives in Fig. 1.

The granularity of the concept of matching, i.e., a subscription specifies an interest in incoming event messages, divides pub/sub into topic-based and content-based systems. Early research focused on the former concept, and supposed event messages to

* The author has been partially funded by the New Zealand Government under the New Zealand International Doctoral Research Scholarships (NZIDRS) scheme.

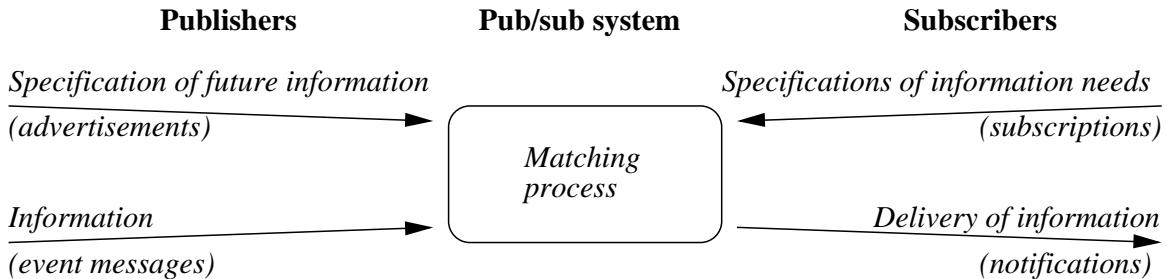


Fig. 1. Schematic overview of users and their communication directives in pub/sub

specify a topic and subscriptions to state an interest in topics. The subscriber thus receives notifications about all messages of the specified topic. A representative of this approach is newsgroups.

Current research allows for a more sophisticated concept of matching based on the actual content of messages (e.g., GRYPHON [1] and SIENA [10]). Within this paper, we assume event messages to state an *event type* and a set of *attribute-value pairs* according to this type. Thereby, the type and these pairs represent the content of this message. Subscriptions contain a *filter expression* for event messages, restricting the values of attributes. Hence, subscriptions specify the content a subscriber is interested in, leading to the term content-based pub/sub. We define advertisements in the same way as subscriptions. However, the semantics is different: Advertisements specify the content a publisher will send in the future.

To confirm the common claim of an increased popularity of the pub/sub area, we have analyzed the number of academic publications on this topic in various digital libraries (ACM, DBLP, Google Scholar, IEEE, and SpringerLink): Whereas the overall number of publications per year has doubled since approximately 2001, the number of publications on pub/sub has quadrupled since that year.

Pub/sub, however, is not only an academic subject. Various companies apply this general event-driven paradigm in the context of message queuing systems, including IBM, Microsoft, and Oracle. Other well-known (commercial) applications of pub/sub include RSS feeds and Google Alerts.

Having introduced this general context of our PhD project, we outline the identified research problems in the next section.

2 Research Problems

An investigation of content-based pub/sub systems reveals their focus on subscriptions and advertisements as conjunctive filter expressions. Nearly all of the proposed solutions strongly depend on this restriction and are not directly applicable to, e.g., subscriptions as arbitrary Boolean filter expressions. Although one could convert arbitrary expressions to disjunctive normal forms (DNF) and handle every conjunction as separate subscription/advertisement [14,15], we believe that this canonical conversion is not a reasonable approach in pub/sub. The reason for this hypothesis is the exponential size of DNFs in the worst case: The already large number of subscriptions and advertisements explodes (exponentially) due to conversions.

In our PhD project, we want to build a content-based pub/sub system that supports arbitrary Boolean subscriptions and advertisements, and show its advantages compared to conjunctive solutions. In order to allow for this support, our research needs to contribute to various parts of these systems and to extend different components, as presented in the following subsections.

2.1 Matching of Arbitrary Boolean Subscriptions

The first component in pub/sub systems that needs to be enhanced is the matching algorithm. Current approaches that target at an efficient and scalable matching only support conjunctions, e.g., [12,16]. Thus, we need to develop a novel matching solution for arbitrary Boolean subscriptions.

This novel approach should be a general-purpose solution that also efficiently supports the use of mere conjunctive subscriptions. An obvious idea is to extend an existing conjunctive algorithm to support arbitrary Boolean subscriptions.

2.2 Optimizing the Routing of Arbitrary Boolean Subscriptions

The second component to be extended is the event routing optimizations. These optimizations are required in distributed pub/sub systems to reduce the sizes of event routing tables. Similarly to matching algorithms, current optimizations are only applicable to conjunctions, e.g., covering [10,14] and merging [13,14]. The reason is the complexity of these approaches for arbitrary Boolean expressions, being (co-)NP-hard.

Our novel routing optimization should efficiently support both arbitrary Boolean and conjunctive subscriptions to be universally applicable. When using an optimization method that works orthogonally to current approaches, there should even exist an optimization potential in situations that cannot be optimized by today's solutions. Additionally, conjunctive systems could simultaneously apply different optimizations to improve the overall optimization effects.

2.3 Supporting Arbitrary Boolean Advertisements

The final area requiring enhancements is the symmetrical support of arbitrary Boolean advertisements. To practically use advertisements, pub/sub systems need to determine the *overlapping* relationships among subscriptions and advertisements. Informally, a subscription s and an advertisement a overlap if a describes at least one event message that matches s . Current computation methods for overlappings exploit the conjunctive nature of subscriptions and advertisements, and are thus not applicable.

Distributed pub/sub systems internally forward advertisements to build up subscription routing tables. To reduce the sizes of these tables, one applies subscription routing optimizations. Currently, event routing optimization approaches (covering and merging, cf. Sect. 2.2) are likewise applied for this purpose. However, these optimizations only support conjunctions.

As a final step, we thus need to develop an overlapping calculation approach for arbitrary Boolean subscriptions and advertisements, and an subscription routing optimization. Both of these solutions should also be applicable to conjunctions.

3 Undertaken Steps

Within our PhD project, we have addressed all three problem areas of pub/sub systems that have been identified in Sect. 2. In this section, we briefly outline these contributions.

3.1 Matching Algorithm

We have developed a matching algorithm [4] for arbitrary Boolean subscriptions. This approach is the first matching solutions that both supports this class of subscriptions and utilizes one-dimensional predicate indexes to realize an efficient and scalable matching process. It is an extension of the conjunctive counting algorithm [16]. An important property of our approach is the support of conjunctive subscriptions in nearly the same way as the original algorithm.

The algorithm works in three steps: Firstly, it determines all predicates that match the attribute-value pairs of an incoming message. Secondly, it computes a set of *candidate subscriptions*, i.e., subscriptions that potentially match the incoming message, by conducting a set of subscription indexes. Finally, the Boolean operators of these candidates are evaluated to determine whether they constitute a real match. We have illustrated these steps in Fig. 2.

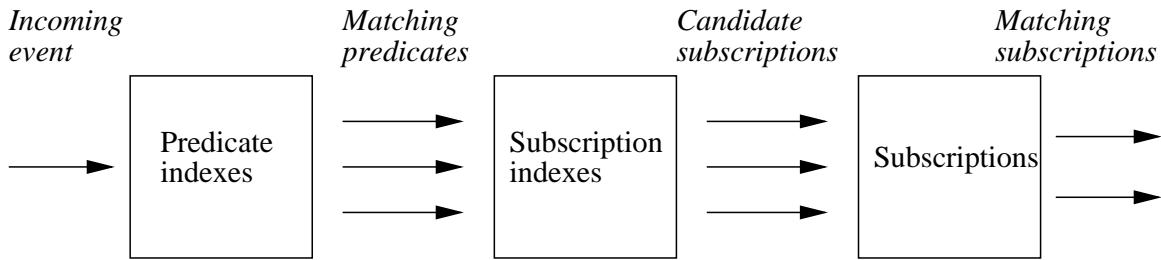


Fig. 2. Overview of the applied 3-step arbitrary Boolean matching algorithm

Memory Requirements. To analyze the memory requirements of various matching algorithms, we have developed a characterization scheme for subscriptions [3]. This scheme allows for the determination of the matching algorithm that requires less memory resources for a given set of subscriptions. Based on this scheme, we have then compared the memory usage of our Boolean and two conjunctive approaches [12,16]. We have found that various settings, even involving only one disjunction in subscriptions, should apply a Boolean matching algorithm due to its space efficiency.

Efficiency. As a next step, we have analyzed the efficiency properties of our Boolean and the conjunctive counting approach. We have evaluated both an artificial scenario but also a typical set of subscriptions for an online auction setting [7]. Our findings show that both matching approaches lead to comparable efficiency properties. Generally, the larger the canonical converted subscriptions, the less efficient the conjunctive and the more efficient our Boolean approach.

3.2 Event Routing Optimization

For the distributed part of pub/sub systems, we have developed a novel routing optimization, *subscription pruning* [9]. This optimization works for all kinds of Boolean subscriptions and can be tailored to optimize in respect to different target parameters, e.g., memory usage, efficiency, and network load [6].

The broad idea of subscription pruning is to remove (prune) parts of the tree structure that represents a Boolean subscription. To ensure correct event routing, every pruning operation needs to create a more general subscription. Pruning subscriptions in routing tables decreases the sizes of these tables by reducing the complexity of their routing entries. We have given an example event routing table before and after applying pruning in Fig. 3.

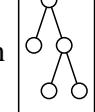
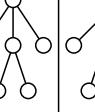
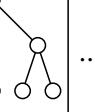
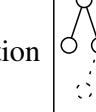
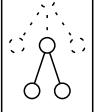
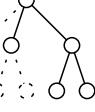
Un-optimized routing table					Optimized routing table				
Subscription				...	Subscription				...
Neighbor	N_1	N_2	N_2	...	Neighbor	N_1	N_2	N_2	...

Fig. 3. Event routing table before (left) and after (right) applying subscription pruning

Analysis. We have analyzed and evaluated subscription pruning in an online book auction scenario [7] and found a promising optimization effect: At the same time, subscription pruning strongly decreases the sizes of event routing tables, increases the system throughput, and only slightly increases the internal¹ network load. The network-based optimization variant has led to the best overall results [6].

Comparison. Although subscription pruning is the first optimization that is practically applicable to Boolean subscriptions, we have additionally undertaken a comparison to conjunctive solutions: We have analyzed the conjunctive counting algorithm in combination with the subscription covering optimization, and our Boolean matching approach in conjunction with subscription pruning. Subscription pruning shows a more stable optimization behavior than subscription covering, i.e., it optimizes in all of the analyzed scenarios. Subscription covering, however, only optimizes if subscriptions fulfil its assumption, i.e., there are strong subset relationships among them.

Subscription pruning optimizes orthogonal to subscription covering, as it has been our design goal (cf. Sect. 2.2). Pruning can thus be additionally applied to subscription covering, which leads to an improved overall optimization, as we have shown in experiments.

¹ Subscribers are still notified correctly because the final notification delivery is based on the original, un-pruned subscription.

3.3 Advertisements

Our final step on the way to arbitrary Boolean pub/sub has been the symmetrical support of arbitrary Boolean advertisements. We have firstly developed an algorithm to calculate the overlapping relationships among arbitrary Boolean subscriptions and advertisements. Secondly, we have proposed the first optimization approach that is tailored to advertisements, advertisement pruning.

Overlapping Calculation. Our overlapping calculation approach uses the notion of *violating predicates* [5] (i.e., non-overlapping predicates) to determine the overlaps among subscriptions and advertisements. Conjunctive computation algorithms, conversely, can be based on overlapping predicates due to their common assumption that subscriptions contain exactly one predicate per attribute. Our semantics of Boolean subscriptions and advertisements, however, allows any number of predicates per attribute.

Similar to the matching approach (Sect. 3.1), our algorithm firstly determines violating predicates, secondly computes a set of *candidate overlappings*, and finally analyzes these candidates to resolve their real status of overlapping. The illustration in Fig. 4 exemplifies these steps from the viewpoint of subscriptions, i.e., for an incoming subscription the algorithm determines overlapping advertisements. Our approach works similarly for the other direction due to the symmetry of the overlapping relationship.

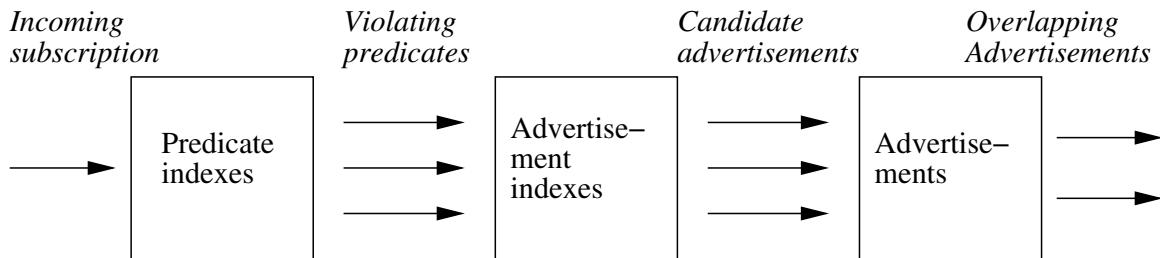


Fig. 4. Overview of the applied 3-step overlapping calculation algorithm

When comparing our approach to conjunctive solutions, we have found similar efficiency properties for the function problem (determination of all overlappings) in our online auction scenario. Our Boolean approach, however, shows a much better performance for the decision problem (determination whether at least one overlapping exists).

Advertisement-Based Optimization. *Advertisement pruning* [8] is the first designated subscription routing optimization. It takes a similar approach to subscription pruning (Sect. 2.2), i.e., it removes selected branches of the Boolean tree structure of advertisements. The overall goal of advertisement pruning is to reduce the memory usage for subscription routing tables (containing advertisements) without strongly increasing the existing overlapping relationships. The influence on the existing overlappings is estimated based on the violating predicates of advertisements and subscriptions, used in the overlapping calculation algorithm that has been presented in the previous paragraph.

4 Future Steps

In the previous section, we have described the research we have undertaken so far. These contributions widely align with the problems that have been identified in Sect. 2. However, some work still needs to be done.

4.1 Experimental Study

Currently, we are undertaking a large set of practical experiments to (1) further analyze the behavior of our approaches, and (2) to compare our solutions to existing methods in detail. One can, obviously, undertake an enormous amount of studies to investigate all details of the performance of our and existing approaches. One of the main challenges of our experiments is thus to identify those settings that provide the most valuable insights into the behavior of the analyzed algorithms.

4.2 Writing Up

Undertaking research is only the first step towards a PhD. The second (and at least equally important) step is to write up all findings and methods of research, to structure and organize thoughts and ideas, and finally to hand in a complete dissertation. We have just started this second step and, ultimately, work its conclusion.

5 Conclusions

The rather general presentation of our PhD project in this paper has aimed to give an overview about the broad topic we are working on in the associated dissertation. We have published details about our research findings in the three outlined subareas of pub/sub (cf. Sect. 3), as referenced in the respective sections. A more technical summary of our general PhD topic can be found in [2]. To conclude this paper, we now want to outline the general contributions of our research to the pub/sub area.

In our opinion, one of the most severe problems that pub/sub is facing is the lack of realistic, widely applied applications. A popular setting in the literature is the stock broker example, simplifying the pub/sub paradigm to the selection of stocks of certain companies. We believe that the choice of such scenarios has contributed to the focus of current pub/sub systems on conjunctive expressions, and that the analysis of more sophisticated settings will reveal the requirement of more complex subscriptions. We hope to have set a starting point with our choice of an online auction example scenario [9].

The chosen application scenario also influences the test settings that are applied in conducted studies. We believe that our analysis of typical event distributions [7] and the identification of various subscription/advertisement classes [8] is a step in the direction of more meaningful experiments and evaluations. This focus on actually existing problems, hopefully, will lead to an even wider adoption of pub/sub in everyday systems.

Although we strongly believe that academic research should not be controlled by economical constraints, and it should be possible to undertake research merely for the sake of it, a look into the real world might not always be damaging. This is particularly important to counteract the common prejudice that research creates new problems but does not solve existing ones.

References

1. G. Banavar, T. Chandra, B. Mukherjee, J. Nagarajarao, R. E. Strom, and D. C. Sturman. An Efficient Multicast Protocol for Content-based Publish-Subscribe Systems. In *Proceedings of the 19th IEEE International Conference on Distributed Computing Systems (ICDCS '99)*, pages 262–272, Austin, USA, May 31–June 4 1999.
2. S. Bittner. Supporting Arbitrary Boolean Subscriptions in Distributed Publish/Subscribe Systems. In *Proceedings of the 3rd International Middleware Doctoral Symposium (MDS 2006)*, Melbourne, Australia, November 27–December 1 2006.
3. S. Bittner and A. Hinze. A Detailed Investigation of Memory Requirements for Publish/Subscribe Filtering Algorithms. In *Proceedings of the 13th International Conference on Cooperative Information Systems (CoopIS 2005)*, pages 148–165, Agia Napa, Cyprus, October 31–November 4 2005.
4. S. Bittner and A. Hinze. On the Benefits of Non-Canonical Filtering in Publish/Subscribe Systems. In *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW '05)*, pages 451–457, Columbus, USA, June 6–10 2005.
5. S. Bittner and A. Hinze. Arbitrary Boolean Advertisements: The Final Step in Supporting the Boolean Publish/Subscribe Model. Technical Report 06/2006, Computer Science Department, The University of Waikato, June 2006.
6. S. Bittner and A. Hinze. Dimension-Based Subscription Pruning for Publish/Subscribe Systems. In *Proceedings of the 26th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW '06)*, page 25, Lisbon, Portugal, July 4–7 2006.
7. S. Bittner and A. Hinze. Event Distributions in Online Book Auctions. Technical Report 03/2006, Computer Science Department, The University of Waikato, February 2006.
8. S. Bittner and A. Hinze. Optimizing Pub/Sub Systems by Advertisement Pruning. In *Proceedings of the 8th International Symposium on Distributed Objects and Applications (DOA 2006)*, pages 1503–1521, Montpellier, France, October 30–November 1 2006.
9. S. Bittner and A. Hinze. Pruning Subscriptions in Distributed Publish/Subscribe Systems. In *Proceedings of the Twenty-Ninth Australasian Computer Science Conference (ACSC 2006)*, pages 197–206, Hobart, Australia, January 16–19 2006.
10. A. Carzaniga, D. S. Rosenblum, and A. L. Wolf. Design and Evaluation of a Wide-Area Event Notification Service. *ACM Transactions on Computer Systems*, 19(3):332–383, 2001.
11. M. Cilia and A. P. Buchmann. An Active Functionality Service For E-Business Applications. *ACM SIGMOD Record, Special Issue on Data Management Issues in Electronic Commerce*, 31(1):24–30, 2002.
12. F. Fabret, A. Jacobsen, F. Llirbat, J. Pereira, K. Ross, and D. Shasha. Filtering Algorithms and Implementation for Very Fast Publish/Subscribe Systems. In *Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data (SIGMOD 2001)*, pages 115–126, Santa Barbara, USA, May 21–24 2001.
13. G. Li, S. Hou, and H.-A. Jacobsen. A Unified Approach to Routing, Covering and Merging in Publish/Subscribe Systems based on Modified Binary Decision Diagrams. In *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS '05)*, pages 447–457, Columbus, USA, June 6–10 2005.
14. G. Mühl and L. Fiege. Supporting Covering and Merging in Content-Based Publish/Subscribe Systems: Beyond Name/Value Pairs. *IEEE Distributed Systems Online (DSOnline)*, 2(7), 2001.
15. P. R. Pietzuch. *Hermes: A Scalable Event-Based Middleware*. PhD thesis, University of Cambridge, Queens' College, February 2004.
16. T. W. Yan and H. García-Molina. Index Structures for Selective Dissemination of Information Under the Boolean Model. *ACM Transactions on Database Systems*, 19(2):332–364, 1994.

Fitting spatial ability and free-formed questions into Intelligent Tutoring Systems development

Nancy MILIK, Antonija MITROVIC and Michael GRIMLEY

Intelligent Computer Tutoring Group
Department of Computer Science and Software Engineering
University of Canterbury, New Zealand
{nmi14,tanja}@cosc.canterbury.ac.nz

Abstract. Building effective learning tools is an art that can only be perfected by a great deal of explorations involving the tools' audience: the learners. This paper focuses on accounting for the learners' spatial ability as well as providing an additional help channel in Intelligent Tutoring Systems. We modified ERM-Tutor, a constraint-based tutor that teaches logical database design, to provide not only textual feedback messages, but also messages containing combinations of text and pictures, in accordance with the multimedia theory of learning [1]. We also added a question-asking module which enables students to ask free-form questions. Results of preliminary studies performed show a promising indication for further explorations. We plan to use these results as the basis for another evaluation study in early 2007.

1. Introduction

In today's society, people constantly face the challenge of acquiring new skills and knowledge. Rapid and widespread developments in technology have made information available and easily accessible more than ever before. Such ease of access alone however, does not necessarily result in a better learning gain for students. Although e-learning tools, such as WebCT [2], are becoming more popular in educational institutions, they do not effectively support learning. While they make it easier for teachers to present instructional material and carry out some administrative tasks, they do not provide students with individualised feedback based on their performance, which is crucial for successful learning. An effective solution that provides adaptive pedagogical assistance for each student is Intelligent Tutoring Systems (ITSs).

ITSs are interactive computerised tutors that provide an environment where students carry out problem-solving activities, and receive feedback on their actions. As the student interacts with the system, it tracks their behaviour and produces/maintains a model of the student's state. This model is used in adapting the environment towards the needs, knowledge, learning abilities and preferences of the student. This includes decisions about the timing and content of teaching actions and feedback to be presented to each individual student. Such adaptations have been shown to result in significant improvement over simplistic e-learning tools, especially

in fields that require practical proficiency [3, 4]. Nevertheless, this is still a growing discipline that is utilising findings in educational and psychological theories and new developments in Artificial Intelligence and software and hardware technology.

In this paper, we describe a master's project which focuses on enhancing ERM-Tutor, a constraint-based ITS that teaches logical database design (i.e. the algorithm for mapping conceptual to logical database schemas), by (a) adapting the feedback presentation mode towards the student's spatial ability and (b) incorporating a module where students are able to ask for additional clarifications.

The next section presents an overview of ERM-tutor, followed by an overview of spatial ability in Section 3 and the question-asking module in Section 4. We then describe the preliminary studies and the results obtained in Section 5, followed by conclusions and future work in the final section.

2. ERM-Tutor

Constraint-based tutors enhance learning in a variety of domains, such as database querying (SQL-Tutor [5]), conceptual database design (ER-Tutor [6]) and data normalisation (NORMIT [7]). ERM-Tutor [8] is another web-based tutor in which students practice the 7-step algorithm for mapping conceptual database schemas (i.e. ER diagrams) into relational schemas. Each step in the algorithm maps one ER concept by either creating a new relation or altering previously created relations by adding foreign keys and attributes.

The interface (Fig. 1) enables students to view problems, work on their solutions and receive feedback. The problem-solving area is the main part of the page, and its general layout is the same for all steps. The student creates or alters one relation at a time. Each step of the algorithm is broken into subtasks. For example, in step one, the student maps one regular entity type at a time, and the system checks the resulting relation before moving on to the next entity type. Fig. 1 illustrates a situation when the student has mapped the MEETING weak entity type, and has specified a relation (with the same name) with three attributes (*timing*, *id* and *description*). For each attribute, the student can specify whether it is a primary and/or foreign key. When the student completes the relation, he/she can request the system to check the solution. If there are any mistakes in the solution, ERM-Tutor provides feedback. In Fig. 1, the system informs the student that there are some missing attributes as well as a foreign key from the owner for the MEETING relation. If the solution is correct, the student can move on to the next entity type, or to the following step of the algorithm.

3. Spatial Ability

Spatial ability is a psychometric construct essential to activities related to spatial reasoning, such as the ability to manipulate images or spatial patterns into other arrangements [9]. Learners with high spatial abilities perform better with graphic or spatially-oriented content than those with low spatial ability.

Psychometric tests used for determining spatial ability typically consist of paper-and-pencil tasks requiring inspecting, imagining or mentally transforming shapes or objects at the *figural* scale of space. These tests do not provide a discrete value on the spatial ability scale, but rather a relative position within a sample group that determines high or low classifications. We explored short versions of two tests from the battery of cognitive tests [10]: a ten-item Paper Folding Test intended to evaluate a component of spatial ability called visualisation, and an eighty-item mental Card Rotation Test which evaluates spatial orientation. Each test has a three-minute time limit and is suitable for ages 13-18.

It is worth noting, however, that a low spatial ability score is not a deficit, and there is even evidence that it can be improved through training and practice [11]. Nevertheless, changing ITSs to accommodate low spatial ability learners, rather than providing a spatial ability training environment, could be more practical and beneficial for the system/domain's problem solving task. That is, learners with different spatial abilities should receive different types of content.

The theory of multimedia learning [1] presents a number of principles for customising instructional content towards individuals' spatial ability. Mayer defines multimedia as the presentation of material using both words and pictures, and proposes that presenting verbal explanations alone in instructional situations is less conducive to learning for some students than presenting verbal explanations in conjunction with pictures [12]. Subsequently, he defines a multimedia instructional message as communication that makes use of our dual learning channel [13] which is intended to foster learning.

The screenshot shows the ERM-Tutor software interface. At the top, there is a navigation bar with links for 'Problem Text', 'Completed Tables', 'Change Problem', 'Help', and 'Logout'. Below the navigation bar, the title 'Step: 2. Map all the weak entity types' is displayed. To the right of the title is a 'Feedback' section containing two numbered items:

1. There are some missing attributes from your table. Check that you have correctly identified and spelt all the attributes for the table.
2. For this step you need to specify all the foreign keys from the owner entities.

In the center of the screen, there is an Entity-Relationship (ER) diagram. The diagram shows entities: LECTURER, STUDENT, and MEETING. Relationships: MEETS (between LECTURER and STUDENT), and MEETING (between LECTURER and STUDENT). Attributes: id (under LECTURER), Student_number (under STUDENT), Description and Topic (under MEETING).

On the left side, there are several sections with labels and input fields:

- Instructions:** Choose the entity you want to map, then specify each attribute you want to add to the table you have created, use the checkboxes if the attribute is a key or foreign key.
- Table attribute:** A text input field with an 'Add attribute' button, and checkboxes for 'Key' and 'Foreign Key'.
- Current table:** A table named 'MEETING' with columns: 'time' (with an 'id' checkbox), 'description', 'Edit', 'Delete' (all in red).
- Relationship:** A relationship named 'MEETS' with an 'Edit' and 'Delete relationship' button.
- Feedback:** Buttons for 'List All Errors', 'Check table', and 'Clear'.

On the right side, there is a 'Questions' section with a 'Enter your question:' text area and a 'Submit question' button.

Fig. 1. Screenshot of ERM-Tutor

Fig. 2 shows a representation of the dual channel theory. One channel is dedicated to processing words, whether printed or spoken, and the other is for processing pictorial forms. Based on this assumption, along with the assumptions that each channel has a limited capacity and require active processing, Mayer defines the Cognitive Theory of Multimedia Learning [1]. The theory states that learning occurs when learners attend to relevant incoming information (sensory memory), select and organise important information and integrate it with their prior knowledge (working memory) into mental representations (long-term memory). Mayer argues that making use of both visual and auditory channels when presenting learning instructions aids in deep, or meaningful, learning, indicated by good retention and transfer performance. His rationale is that when presenting a message combining an image and text, the information is effectively being perceived and processed twice (once through each channel). Moreover, the words and pictures complement each other, aiding the learner to mentally encode and integrate the information.

Mayer defines a number of principles for designers of instructional environments to follow in order to make the maximum use of the learners' dual channels. The principle that is of most interest to us however, is the individual differences principle, which states that “[multimedia] design effects are stronger for low-knowledge learners than for high-knowledge learners and for high spatial learners rather than from low spatial learners” (p. 161) [1]. This is because high-knowledge learners are able to use their prior knowledge to compensate for the cognitive processing needed to integrate the information received by the dual-channel. On the other hand, low-spatial learners must devote so much cognitive capacity to mentally integrate the information. Therefore, it is the combination of the learners' spatial ability and level of knowledge that influences their meaningful/deep learning.

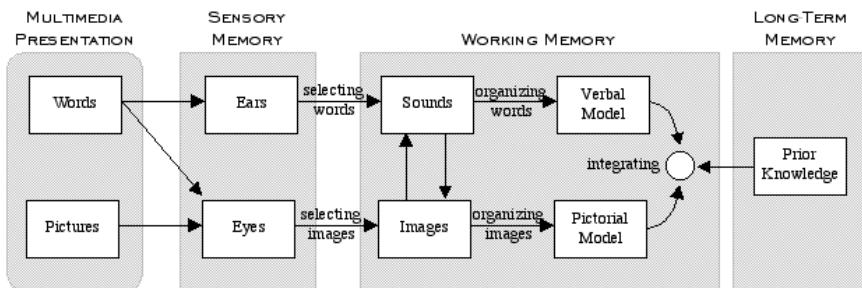


Fig. 2. Information processing via dual learning channels (Figure 3.2 from [1])

Influenced by Mayer's work, we created a new version of the system. The original ERM-Tutor only provides text-based feedback. Following the multimedia learning theory, we decided to incorporate a pictorial aspect in the messages; for each feedback message, we created a graphically annotated version. To make the original and the newly created messages comparable, we kept the text identical in both versions. The only difference is the addition of a pictorial representation in the new version. Fig. 3 shows the multimedia (text and picture) version of the second feedback message given in Fig. 1. A total of 112 images were created, each corresponding to a

single feedback message. In addition, ERM-Tutor was modified to cater for both versions of feedback and prepared for an evaluation study described in Section 5.

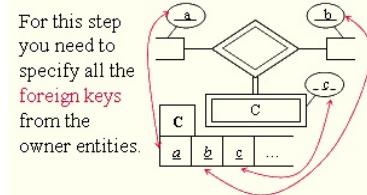


Fig. 3. An example feedback message in multimedia representation

4. Question Asking Module

ITSs provide feedback on students' actions, but students do not always fully understand the feedback they receive. Therefore, it would be beneficial for students to be able to ask questions at any time. Although researchers in cognitive science and education have reported learning benefits for environments that encourage students to generate questions [14], question-asking is still a young concept in the ITS field (e.g. [15]). Question generation is believed to be a primary attribute of active learning which reveals how deeply the learner has mastered the material and even shifts the student's goals from performance toward learning orientation [16].

In this light, we added a question-asking module to ERM-Tutor. We defined 98 distinct questions, based on our experiences in teaching the mapping algorithm and our experience with other constraint-based tutors. These questions can be categorised into interface usage (“What does the button Check Step do?”), definitions of terms (“What is a foreign key?”), diagram notations (“How is an attribute represented in the ER-diagram?”), mapping regulations (“How is a relationship mapped?”), and deeper questions (“Why are the steps arranged in this order?”). Each question is stored along with its textual answer. The question database additionally includes a number of repeated questions that are phrased differently, resulting in a total of 182 questions.

The TFIDF (Term Frequency Inverse Document Frequency) vector weighting scheme [17] was chosen as the information retrieval mechanism. In our system, the questions are read from the database and separated into words. The weight of each question and word is calculated, and words are indexed in a hash table. When the student asks a question, the same calculations are applied to the query string: it is also broken-up into words and their weights are calculated. Each question is then allocated a query weight. Finally, the answer corresponding to the question with the highest query weight is returned to the student. To evaluate the subjective relevance of the answers returned, students are encouraged to submit their ratings of the answers; however, the system does not enforce it to avoid mode errors and distractions from the problem solving task.

5. Preliminary Studies

We preformed two preliminary studies with students enrolled in an introductory database course at Canterbury University in November 2005 and March 2006. The aim of the first study was to investigate the usage of free-form questions. 29 students logged into ERM-Tutor at least once, but five students used it for less than two minutes and so their logs were excluded from analyses. The average interaction time was under one hour (54min, $sd=63min$), ranging from several minutes to 4.5 hours over several weeks. The number of sessions ranged from one to four (mean=1.67, $sd=0.96$). On average, students attempted 4.6 problems and completed 25% of them.

Only eight students asked questions, with a total of 24 questions submitted. The number of questions per student ranged from one to five. The questions can be categorised into task-focused (50%), definition-focused (8%) and phatic questions (42%). Task-focused questions ask directly for help solving the problem (e.g. "How could I solve this table?"). For instance, three students copied the feedback messages, added a question mark at the end or a "How to" at the start, and submitted them as the questions. Definition-focused questions ask for definition of terms. There were only two such questions submitted: "What is foreign key?" and "What is multivalue?" Phatic questions establish a sense of social mood (e.g. "What is your name?", "How are you?" and "How do you answer questions?"). Excluding phatic questions, 14 questions were relevant for students' actions. Five of these questions were answered correctly, and for two of these, the students specified highest relevance. The answer could not be found for one question. The remaining questions received answers which were related to the query, but not useful to students. This happened when the students did not formulate questions well, but instead copied a part of the feedback message then added a question mark at the end (e.g. "Make sure the relationship is 1:1?").

Our hypothesis for the second study was that students with a high spatial ability level will benefit more from multimedia feedback than students with a low spatial ability, given the same background knowledge. As each student's spatial ability level (either high or low, as opposed to the actual value) is determined relatively to the sample group, it was decided to compute it in a post-hoc manner. The students were randomly allocated to one version of the system, providing either textual or multimedia feedback. The assumption was that each group would ultimately include students with high and low spatial abilities. Therefore, the experiment allows for a 2x2 comparison: textual messages for high (TH) and low spatial ability students (TL), and multimedia messages for high (MH) and low spatial ability students (ML).

The study was conducted in two sessions of scheduled labs on ER mapping, straight after students had attended lectures on the topic. Each participant attended one of the sessions, and worked with ERM-Tutor individually, solving problems at his/her own pace. At the start of a session, the students were given an information sheet describing the study, a consent form, and a pre-test on paper (with the maximal score of 4). The average score on the pre-test for all students was 2.23 ($sd=1.15$). To make the results of the pre-test and post-test comparable, two tests were used; students in the first session used version A as the pre-test and version B as the post-test and students in the second session used the reverse.

When a student logged onto the system, he/she was presented with a set of instructions explaining the two spatial ability tests, with sample problems.

Additionally, for each test, they were asked to rate their own ability on a scale of 1 to 5 before sitting the tests. They had three minutes to solve the problems in each test. Once the spatial tests were completed, or their time was up, the students were randomly assigned to one of the two versions of the system. At the end of the session, students were asked to fill in a post-test and a questionnaire about the system. Finally, the students were encouraged to use the system at any time until the end of the course.

Out of 74 students enrolled in the course, 55 students participated and completed both spatial tests. Before completing each test, the students were asked to rate their own ability of the spatial skill. For the paper fold test, the average rating of 6.62 was close to the actual test score of 6.89 out of a possible 10. The students' personal rating for the card rotation test had a mean of 7.6. As explained, the total possible score for the card rotation test is 80. We computed the total test score by dividing the score by 8, thus giving a range of 1–10. The students scored a mean of 6.43. To compute the spatial ability of each student, we added both test scores giving a possible range of 1–20. Using a median split, a total of 28 students scored above the median and were classified as high spatial, and the other 27 students were classified as low spatial.

The pre-test was collected at the start of the session, while the post-test was administered after two hours of interaction. Only 13 students completed both tests, scoring a mean of 1.92 ($sd=1.04$) on the pre-test, and 3 ($sd=1.15$) on the post-test, resulting in significant improvement of their performance ($t=3.09$, $p < 0.001$). The scores for the four groups are given in Table 1. These preliminary results (although with small numbers) seem to refute Mayer's prediction that high spatial learners will benefit most from multimedia messages. However, it does seem that the subsets of participants from the TH and MH groups who completed both tests started with higher pre-existing knowledge, and therefore Mayer's individual differences principle may be more pertinent in that low knowledge individuals will have a higher gain. Of course, with such low numbers of submitted tests, we might expect a lot of error and therefore further investigation is warranted.

The system recorded all student actions in logs. Due to a technical problem however, the logs from the first session could not be used. A total of 17 students used the system for more than 10 minutes. On average, students attempted 3.4 problems and completed 33% of them. The numbers of valid logs in each condition are too small, and we are therefore unable to closely analyse the effect of the students' spatial ability on their performance. Analyses of the questionnaires showed that students who received multimedia feedback rated the overall quality of the feedback messages 25% higher (mean of 4 out of a possible 5) than those who received textual feedback (mean of 3 out of a possible 5). There were also some encouraging comments from the questionnaires submitted in the second session. Students liked the system, and appreciated the problem solving environment provided to solve the problems.

Table 1. Pre/post test results for the students who sat both tests

Feedback	Low Spatial			High Spatial		
	No.	Pre-test	Post-test	No.	Pre-test	Post-test
Textual	TL: 4	1.5 (1)	3.5 (0.6)	TH: 3	2 (1)	2.3 (0.6)
Multimedia	ML: 2	1.5 (0.7)	3.5 (0.7)	MH: 4	2.5 (1.3)	2.75 (1.9)

6. Conclusion

Rapid and widespread development of computerised learning tools have proven the need for further exploration of the learners' personal characteristics in order to maximise the use of the current technology. In particular, this paper has looked at the potential of accounting for spatial ability and providing an additional help channel in ERM-Tutor; a constraint-based tutor that teaches the procedural task of mapping ER diagrams into relational schemas. We presented results from two preliminary studies.

The first study, which investigated the question-asking module, showed some evidence that students welcome the idea of asking free-form questions and confirmed the need for eliciting deeper questions. The results from the second study, which evaluated the effectiveness of the type of content representation (text only vs. multimedia) to the learner's spatial ability level, show an overall improvement in the students' domain knowledge level after using ERM-Tutor for the duration of the study (2 hours). Although the amount of data collected was small, the results show a promising indication for further explorations. We plan to use these studies as the basis for another evaluation study testing the same hypotheses in early 2007.

References

1. Mayer, R., *Multi-media Learning*. 2001: University of California.
2. Lu, J., C.S. Yu, and C. Liu, *Learning style, learning patterns, and learning performance in a WebCT-based MIS course*. Information and Management, 2003. 40(6): p. 497-507.
3. Koedinger, K.R., et al., *Intelligent tutoring goes to school in the big city*. International Journal of Artificial Intelligence in Education, 1997. 8(1): p. 30-43.
4. Mitrovic, A., et al., *Constraint-Based Tutors: A Success Story*. IEA/AIE, 2001: p. 931-940.
5. Mitrovic, A., *A Knowledge-Based Teaching System for SQL*. MEDIA, 1998: p. 1027-1032.
6. Suraweera, P., *An Intelligent Tutoring System for Entity Relationship Modelling*. International Journal of Artificial Intelligence in Education, 2004. 14(3): p. 375-417.
7. Mitrovic, A., *NORMIT, a Web-enabled tutor for database normalization*. Proc. ICCE, 2002: p. 1276-1280.
8. Milik, N., M. Marshall, and A. Mitrovic. *Responding to Free-form Student Questions in ERM-Tutor*. in *Proc. Intelligent Tutoring Systems*. 2006. Taiwan: Springer.
9. Carroll, J., *Human cognitive abilities*. 1993, England: Cambridge University Press.
10. Ekstrom, R., J. French, and H. Harman, *Manual for kit of factor referenced cognitive tests*. 1997: Princeton.
11. Vicente, K.J. and R.C. Williges, *Accommodating individual differences in searching a hierarchical file system*. Journal of Man-Machine Studies, 1988. 29(6): p. 647-668.
12. Mayer, R., *Multimedia learning: Are we asking the right questions?* Educational Psychologist, 1997. 32: p. 1-19.
13. Paivio, A., *Mental Representations: A Dual Coding Approach*. 1986: Oxford University.
14. Beck, I.L., et al., *Questioning the author: An approach for enhancing student engagement with text*. Delaware: International Reading Association, 1997.
15. Anthony, L., et al. *Student Question-Asking Patterns in an Intelligent Algebra Tutor*. in *Proc. Intelligent Tutoring Systems*. 2004. Brazil: Springer.
16. Graesser, A.C. and N.K. Person, *Question asking during tutoring*. American Educational Research Journal, 1994. 31: p. 104-137.
17. Salton, G. and C. Buckley, *Term-weighting approaches in automatic text retrieval*. Information Processing and Management: an International Journal, 1988. 24(5): p. 513-523.

The Great Keyboard Debate: QWERTY versus Dvorak

Kathryn Hempstalk

University of Waikato, Hamilton, New Zealand
`kah18@cs.waikato.ac.nz`,
WWW home page: <http://www.cs.waikato.ac.nz/~kah18>

Abstract. Many layouts have been proposed for keyboards since the invention of the typewriter in the 1800s. The two most popular formats, QWERTY and Dvorak, have been at the centre of a great debate for decades—both layouts are claimed to be optimal. Academic evidence is available supporting each layout, yet over 75 years since the debate began there is still no clear answer. This paper discusses the history of each format and presents new scientific experiments designed to give a final determination of the optimal layout.

1 Introduction

Since the beginning of the 20th Century a battle has quietly been fought over the optimal layout of alphanumeric keys on a keyboard. One might surmise the winner to be QWERTY because this is virtually the only keyboard layout available for purchase today. Although not universally accepted, the most commonly cited reason that QWERTY is so popular is because such an overwhelming market share was gained by QWERTY keyboards that it was unprofitable to continue making Dvorak ones [1]. The question of which keyboard format is optimal still rages on. Academic writing seems to point to QWERTY and Dvorak reaching a draw; equal numbers of papers exist in favour of each format [2–4].

This paper investigates which layout is the most efficient for text entry on a keyboard. The research conducted in this paper is part of a PhD research project on identifying a typist by the way they type. Whilst the cognitive difficulties associated with text entry are ignored in this paper because they are largely irrelevant with regards to keyboard layout, these difficulties can be used to identify a computer user. This paper is related to, but not part of, the main goal of the PhD: typist recognition. The PhD is mainly focused on implementation of machine learning techniques to process typing recordings into profiles of users. The work in this paper resulted from experimentation of building profiles based on finger movement instead of absolute times. The PhD project began in March 2006.

The next section in this paper describes the history of the two formats. Section 3 describes the design of the experiments and Section 4 gives the results of the experiments. Finally some conclusions are drawn in Section 5 and future work is discussed in Section 6.

2 The History of the Keyboard Layout

The typewriter was first invented in the early 1800s and the keyboard was laid out in alphabetical order with piano-like keys. The modern typewriter, invented in 1868 by Christopher Sholes, also had an alphabetical layout (see Figure 1) [5]. Because the key hammers were not spring-loaded and required gravity to return to their rest position, the hammers frequently jammed when two neighbouring keys were pressed in quick succession. Sholes continued to make improvements to his typewriting machine and eventually patented the QWERTY layout in 1878 (see Figure 2) [6]. The new layout was designed to place popular letter pairs onto opposite hands in order to reduce the occurrence of jamming. The introduction of QWERTY had the side effect of slowing typists down because the keys were no longer laid out as intuitively as before. Over time some minor improvements were made to the layout, until it finally became a standard layout (as seen in Figure 3).

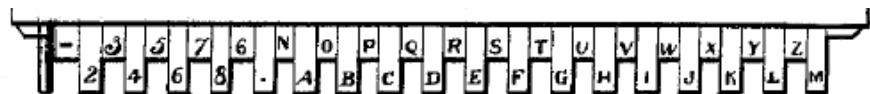


Fig. 1. Shole's alphabetic layout [5].

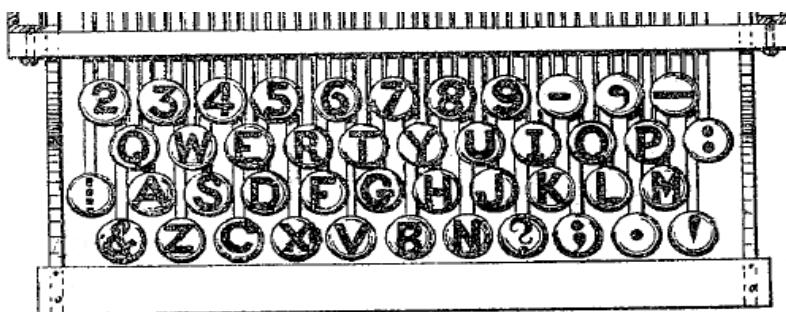


Fig. 2. The original QWERTY keyboard layout [6].

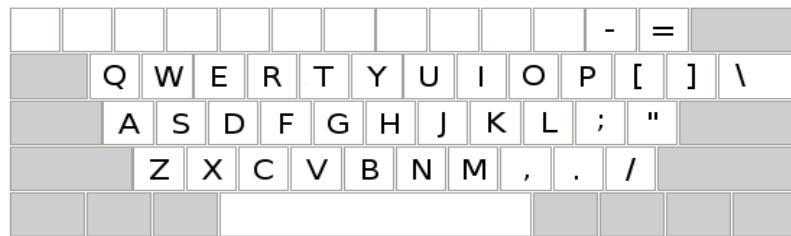


Fig. 3. The modern QWERTY keyboard layout.

To solve the problem of slow typists, regimented lessons were introduced. These lessons helped transition typists from a hunt-and-peck technique through to touch typing and by the early 1900's typists were typing faster than ever before.

In 1936 August Dvorak et al. published *Typing Behavior* [7], a book cataloging an 11 year study into keyboard layouts. Dvorak believed that the QWERTY format was not optimal because it did not place the popular letter pairs in easily accessible positions. He patented his own format, the Dvorak Simplified Keyboard (also known as DSK or Dvorak) layout, which placed popular letters like t, h and e directly under the typist's fingers. The Dvorak layout can be seen in Figure 4. Even Sholes was not happy with his original QWERTY layout, and in his final typewriter patent he included a new layout that has striking similarities to the Dvorak layout—it places all the vowels on one row and under one hand (see Figure 5) [8]. Sholes's final layout was never included in any manufactured typewriting machines.

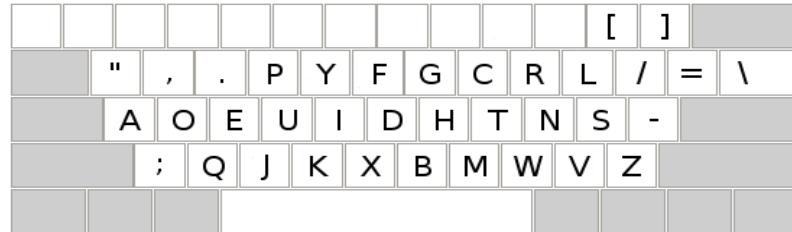


Fig. 4. The Dvorak Simplified Keyboard layout.

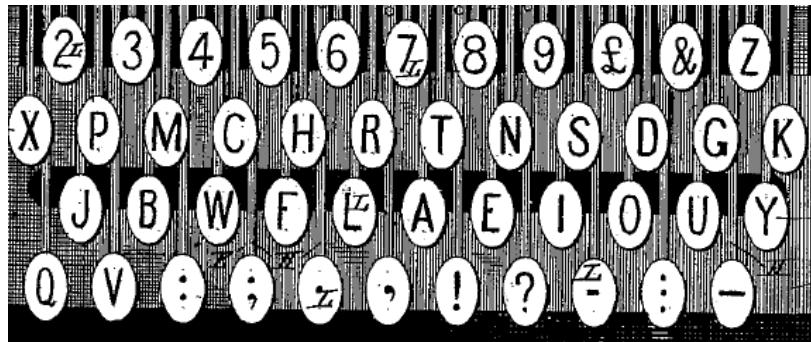


Fig. 5. Sholes's final keyboard layout [8].

As proof that his format was superior, Dvorak trained up some typists at the Navy base where he worked and sent these typists around to various typing competitions. These typists were so successful that the Dvorak layout was eventually barred from all the competitions. Even today, the fastest known typing speed was recorded in the 1940's using the Dvorak Layout—212 words per minute (wpm) burst speed and 150 wpm sustained over 50 minutes [9]. In marketing

the layout, Dr. Dvorak invested heavily in producing a Remington typewriter featuring his layout. Unfortunately this particular typewriter was “silent” (it did not make the typical click-clack noises whilst typing) and did not sell well because many typists disliked this aspect. The Dvorak layout failed to take off and QWERTY is the common format we see in practice today.

3 Experimental Design

Rather than retraining several typists to the Dvorak layout, we assume that there are no cognitive differences caused by laying out a keyboard in a different format: a movement (such as the index finger moving from the home to top row) takes a fixed amount of time, regardless of the keyboard layout. Of course, typing is affected by cognitive aspects—typists may pause between syllables or words as they prepare to type the next section. Assuming equality between touch typing lessons, the pauses should be caused by the typist’s language and keyboarding abilities, not the keyboard layout, and therefore occur in the same places regardless of the format being used.

There are 676 possible combinations of digraphs (also known as letter pairs) among the alphabetical characters on a QWERTY keyboard, and 1936 if the 9 punctuation keys operated by the right little finger are included. To simplify all the possible combinations, the following types of movements are defined for digraphs:

- tap** Both letters in the digraph are the same, meaning the finger can simply ‘tap’ the same key, as in **ee**, **tt**.
- reach** A digraph typed by the same finger on the same hand, moving over a distance of only one key, as in **ft**, **ju**, **ed**.
- hurdle** A digraph typed by the same finger on the same hand, hurdling over the home row, as in **ce**, **un**.
- trill** A digraph typed by adjacent fingers on the same hand, as in **fe**, **op**, **te**.
- rock** A digraph typed by remote fingers on the same hand, called so as the fingers often move in a rocking motion, as in **af**, **jp**, **on**.
- opposite** A digraph typed by different hands, as in **if**, **od**, **ma**.

It is possible to have up, down and side reaches, and up and down hurdles. Here these movements are only used to determine key distance so direction is irrelevant—regardless of the movement being upwards, downwards or to the side, a reach always moves the finger one key distance and a hurdle moves two. Taps and opposites are 0 key distance digraphs. A trill or rock could be any of a 0, 1 or 2 key distance depending on the row distance between the first and second keystrokes. For example, the trill **et** is typed all on the top row and the index finger is already over the position **t** when **e** is pressed, so it has a 0 key distance. Similarly, **ct** has a 2 key distance and **ef** has a 1 key distance. The actual key distances are simplified to the set {0,1,2} to eliminate varying distances caused by typing style and hand, finger and keyboard sizes. The format that is the most distance efficient will have a lower key distance total across a given set

of digraphs. Using key distances is an approximate way of utilising Fitts' law [10] for the keyboard where movements are in three dimensions rather than the required single dimension. Fitts' law is used to calculate efficiency of moving between two objects based on the distance between them and the size of the objects.

Instead of assuming that one key distance will always take a fixed amount of time, the keyboard layouts are also evaluated in terms of timing efficiency. Timing data from two separate research projects focusing on continuous typist recognition were used to obtain the times between two subsequent key presses for all digraphs that appeared in the datasets (for more details on the datasets see [11, 12]). The digraph time was recorded against the key position rather than the letters, meaning that the time on a Dvorak keyboard could be inferred by using times recorded on a QWERTY layout. For example, the digraph **ff** with a 200ms time on a QWERTY layout would be recorded as {uu, 200ms} on the Dvorak layout. Given a set of digraphs, any that did not have a recorded time under either layout were discarded. Of the 1029 unique digraphs found in the books, 474 were discarded. Although this seems like half the data was not used, the 474 unique digraphs make up only 13.4% of the total number of digraphs encountered in the corpus.

The two datasets used were not ideally suited to this task. One of the datasets is composed entirely of Italian text and is recorded with a coarse resolution of 10ms [12], while the other is in English and is recorded with an accurate resolution of 1ms [11]. The keyboard was laid out in the same format in each dataset (it was in QWERTY), but it is not known whether any of the typists were skilled touch typists or not. To combat the inaccuracies in this dataset only the average time for each digraph was used in evaluating the timing efficiency.

In order to have a measure of distance, a corpus of 21 free classic books (including *Moby Dick* and *The Last of the Mohicans*) was mined for digraph appearance. This particular corpus was chosen because it was freely available and has been used in at least one similar experiment (see [13]). Digraphs featuring one or more characters not in the bottom, home or top rows of either keyboard layout were discarded. The space character is always typed by the thumb in touch typing and because the thumb always rests on this key it can be discarded. Numerical keys (which feature above the top row and again on the number pad) and modifiers are discarded because of ambiguity—it is unclear which hand or finger may press each key. Incidentally, these keys are also in the same position in both formats.

The counts of each digraph found in the corpus were multiplied by their key distance and were totaled to give an overall comparison of QWERTY versus Dvorak in terms of distance. In measuring the timing efficiency, key-specific timings were initially used, relating each recorded time on a QWERTY keyboard with the same keys on a Dvorak keyboard (which may have different letters on those keys). The average time for each digraph was then multiplied by the number of times it appeared in the corpus, summing up to give a total amount of time taken to type the corpus under each format. To prevent the data being

skewed because of key placement changes between QWERTY and Dvorak, each layout was also measured using movement specific timings. That is, the average time for a particular type of motion on the keyboard (as defined previously) was recorded, then used to calculate the total time by multiplying the total number of the movements in each book by the average time for that type of movement. The results of the measurements are presented in the next section.

4 Experimental Results

The results from using the 21 books to determine the distance efficiency are presented in Table 1 below. The Dvorak layout clearly requires the fingers to move less over the keys, almost half as much as the QWERTY layout does. Most of the gains from the key distance can be seen in the number of hurdles—an 8 percentage point drop is seen between the QWERTY and Dvorak layouts for digraphs with distances that require the fingers to move across at least two keys. Dvorak is the more distance-efficient layout, and Dvorak users require less effort for text entry because the overall amount of movement around the keyboard is less.

Measures	QWERTY	Dvorak
Total Key Distance	3203413	1774265
Percentage of 0 Key Distance Digraphs	69.3	79.7
Percentage of 1 Key Distance Digraphs	19.7	17.5
Percentage of 2 Key Distance Digraphs	11.0	2.8

Table 1. Key distances over a corpus of 21 books.

Although Dvorak may require the fingers to move less over the keyboard, it does not mean that people will be able to type faster using this layout. As seen in Table 2, using the timing datasets to calculate the timing efficiency produces an entirely different view of the two layouts. After discarding counts of letter pairs which did not appear in both formats, it appears that QWERTY was faster to type on than Dvorak, even though the typist's fingers had to move further. However, the movement-specific times show that Dvorak is slightly faster, contradicting the key-specific measurements. Table 3 shows the average times for each movement (used to calculate the overall movement time), found by averaging all the times for a given movement from both datasets. The timing data used is biased towards QWERTY because it may be possible to achieve different average movement or average key times using the Dvorak layout.

Measures	QWERTY	Dvorak
Total Time - Key Specific (ms)	1.7×10^9	2.1×10^9
Total Time - Movement Specific (ms)	1.83×10^9	1.81×10^9

Table 2. Total time over a corpus of 21 books.

Movement type	Average Time (ms)
tap	170
opposite	238
trill	224
rock	245
hurdle	458
reach	244

Table 3. Average time taken for each movement.

The movement type times, as seen in Table 3, reinforce the ideas behind the Dvorak layout—it is faster to press the same key in succession or use different hands to type a key pair than it is for the fingers to reach or “hurdle” to a different row. This is expected considering that pressing the same key or using alternating hands allows for the finger to be moved into position over the key before it is pressed. Hurdles and reaches require additional time for the finger to leave the key before moving to the next one. However, it was surprising that although the Dvorak layout significantly reduces the incidence of moving over rows, it is not quicker than QWERTY in producing the same text. On closer inspection it was discovered that Dvorak has a lower average time for each of 0, 1 and 2 key distances but the incidence of digraphs with longer times was greater. This is an example of bias favouring QWERTY because it is likely that frequently made movements will have lower times since they are more practiced. Often unpopular QWERTY movements are more common in the Dvorak format due to the different positions of letters on the keys.

5 Conclusion

The results in this paper are not conclusive enough to end the debate over which keyboard layout is the optimal one. The distance measure clearly shows that the Dvorak layout requires you to move your hands almost half as much as the QWERTY layout. The timing measures contradict each other and are biased towards QWERTY. We settle on the following conclusion: the Dvorak layout is the most efficient because it requires the least amount of effort to type some given text, even though it may take approximately the same amount of time as the QWERTY layout.

6 Future Work

Although no further work is intended in this particular area as part of this PhD research project, somewhat related to this is the intention of collecting a better dataset of keyboard recordings. As mentioned earlier, the two datasets are not suited for the task of evaluating keyboard layouts—one is not in English, contains coarse timings but has lots of data, the other is accurate and in English but contains very little data. The non-English dataset also does not contain key

release events and although this is of no significance in the context of keyboard layout comparisons, it has a huge impact on being able to use this dataset for typist recognition (the subject of the PhD under which this research was conducted). Therefore the next logical step is to collect some more typist data that is in English and contains both key press and key release events. Unfortunately the easiest way to collect such data is by using a web-based approach, where timings have a coarse resolution of approximately 10ms. Work is currently underway to collect recordings of emails using a web-based client that has been altered to be able to record keyboard patterns. The data collection period will last 3 months, hopefully collecting at least one email per day for each participant during that time.

References

1. Buzing, P. (2003) "Comparing Different Keyboard Layouts: Aspects of QWERTY, DVORAK and Alphabetical Keyboards", Retrieved July 2006 from: <http://pds.tudelft.nl/~buzing/Articles/keyboards.pdf>
2. Liebowitz, S.J., Margolis, S.E. (1990) "The Fable of the Keys", *Journal of Law and Economics*, Vol. 33, No. 1, University of Chicago Press, Chicago, USA, pp. 1–25.
3. Hiraga, Y., Ono, Y., Yamada-Hisao (1980) "An Analysis of the Standard English Keyboard", Proceedings of the 8th Conference on Computational Linguistics, Tokyo, Japan, Association for Computational Linguistics, Morristown, NJ, USA, pp. 242–248.
4. West, L.J. (1998) "The Standard and Dvorak Keyboards Revisited: Direct Measures of Speed", *Santa Fe Institute*, Working Paper, 98-05-041.
5. Sholes, C.L., Glidden, C. and Soule, S.W. (1868) "Improvement in Type-Writing Machines", US Patent 79868, July 14, 1868.
6. Sholes, C.L. (1878) "Improvement in Type-Writing Machines", US Patent 207559, August 27, 1878.
7. Dvorak, A., Merrick, N.L., Dealey, W.L. and Ford, G.C. (1936) *Typewriting Behavior: Psychology Applied to Teaching and Learning Typewriting*, American Book Company, New York, USA.
8. Sholes, C.L. (1896) "Type-Writing Machine", US Patent 568630, September 29, 1896.
9. Ranger, B (2004) "Barbara Blackburn: The World's Fastest Typist", Retrieved December 2006 from: <http://whitman.syr.edu/facstaff/dvorak/blackburn.html>
10. Fitts, P.M. (1954) The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, Vol 47, No. 6, pp. 381–391.
11. Nisenson, M., Yariv, I., El-Yaniv, R. and Meir, R. (2003) Towards Behaviometric Security Systems: Learning to Identify a Typist. *Principles and Practice of Knowledge Discovery in Databases*, LNAI 2838, Springer-Verlag, Berlin, pp. 363–374.
12. Gunetti, D. and Picardi, C. (2005) Keystroke Analysis of Free Text. *ACM Transaction on Information and System Security*, Vol. 8, No. 3, New York, pp. 312–347.
13. Koppel, M and Schler, J. (2004) Authorship Verification as a One-Class Classification Problem. *Proceedings of the 21st International Conference on Machine Learning*, Banff, Alberta, Canada.

Wireless Sensor Networks for Agricultural Applications

Sam Bartels

University of Waikato

Abstract. This paper explores the concept of wireless sensor networks and discusses the focus of our research. We introduce the reader to what a wireless sensor network is, how it works, and also give some example applications of wireless sensor networks. The paper then identifies the focus of this research as a wireless sensor network design system for agricultural applications, and discusses some of the ideas involved.

1 Introduction

This paper will provide an introduction to the area of wireless sensor networks and potential research. We begin with a section that explains what a wireless sensor network is and how it works. Some potential applications will be discussed in order to give the reader an idea of what is possible with wireless sensor networks. The second section will address the objectives of this research, and the third section discusses what has been achieved so far. Finally, the fourth section will discuss further work to be undertaken.

2 Wireless sensor networks

Wireless sensor networks (WSNs) are an active research area [1][2][3], predominantly in computer science and electrical engineering. Wireless sensor networks are designed to be autonomous devices that cooperate together in order to monitor environmental or physical conditions. Sensor networks make it possible to monitor many different environmental conditions quickly and easily, such as temperature, humidity and motion. WSN research was initially motivated by the military[4], but is now used for many civilian applications. One particular civilian application, agriculture environmental monitoring, will be the key focus of this research.

2.1 What is a wireless sensor network?

WSNs consist of small devices that are called nodes. Each node contains a power supply, sensors and a small computer. The power supply is typically a battery, which may also have a small solar panel for charging. A node can have one or more sensors attached to it of various types. For example, it may have a

temperature sensor for use in a greenhouse, or a vibration sensor for detecting earthquakes. The computer is a low-power, low-cost device that listens to the sensors and then transmits the received information to other nodes.

Typically, the nodes will relay the information back to a single point, such as a central computer server. Nodes have been designed that are millimetres in size [5], and experts are working on shrinking that down to micrometres. Nodes are relatively inexpensive, and prices are falling as technology improves.

2.2 How do wireless sensor networks operate?

WSNs work much like a conventional wireless network; node's establish links between each other following a particular set of rules. For instance, nodes may establish links to all nodes within transmission range. The difference between how WSNs and conventional wireless networks operate is the communication protocols that are used. The protocol needs to be specifically designed for sensor network operation [6].

WSNs typically have low volumes of data, hence only a low bitrate is required. Communication needs to be reliable so that only correct data is received by the end user. The most important aspect of a WSN protocol is that it needs to be energy efficient. Nodes have very limited power and hence cannot afford to waste it on unnecessary data transmissions. This leads to a tradeoff between reliability and energy efficiency.

For the protocol to be completely reliable, the node would need to be able to retransmit repeatedly (worst case) and this would quickly consume energy. On the other hand, for an energy efficient protocol, we could not retransmit and hence communication would not be reliable. A compromise needs to be made between reliability and energy efficiency, such as that used in the concept of directed diffusion [7].

2.3 What applications do wireless sensor networks have?

With the ability to monitor sensors remotely using a home computer, a farmer could quickly and easily discover if any troughs were low on water, or check the soil moisture and temperature in the maize fields. These are just a few examples of the potential uses that WSNs have in the agricultural industry alone. Farms are not the only benefactors from wireless sensor networks. Orchards would also greatly benefit, especially vineyards [8] and kiwifruit growers. Sensors would indicate when the temperature dropped below a particular threshold, and then action could be taken to protect the crops from frost. Agriculture is a large sector of New Zealand's economy [9], as well as many other countries, and hence WSNs could lead to economic benefit.

There are many other potential applications of wireless sensor networks, and not only in agriculture. Wireless sensor networks could be used in bridges and buildings for health monitoring. An example of this was a research project where a wireless sensor network was established on an 80m footbridge in Berkeley, San Francisco. The sensors measured structural vibration and temperature in an effort to determine health status. Another use would be for natural disaster detection; for example, a wireless sensor network may be established around the perimeter of a lake as a early flood warning system.

3 Research direction

The key focus of this research is to develop a system for simple/automated design of a wireless sensor network, in particular, for applications in the agricultural industry. This research will not only explore the physical topology of a wireless sensor network, but also to examine the relationship between the desired logical network, and the actual physical network. Our definition of topology can be found in Fig. 1. A logical network is an abstract method of representing a physical network. Suppose it is desired to construct a wireless network between two houses, A and B. Logically we would view this as a single link AB, however in a physical context, AB may consist of several physical links. It is of interest to this research to examine this relationship and determine methods which will help automate the deployment of wireless sensor networks.

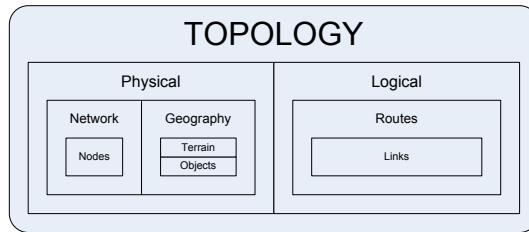


Fig. 1. Topology definition

The physical network topology of a wireless sensor network can be defined in a high or low level context. We will refer to these as high level distribution and low level distribution. High level distribution refers to the intended application of the wireless network. The first high level distribution is the ‘dispersed distribution’. Placement of nodes in a dispersed distribution is not a priority, as long as they are in communication range.

On the other hand, a wireless sensor network where node placement is a priority is referred to as a ‘focused distribution’; that is, the focus is on particular sites in the physical terrain. The above example of a link between houses A and

B would be a focused distribution. The third high level distribution is a hybrid of dispersed and focused, and hence is known as a ‘hybrid distribution’. A hybrid distribution occurs when we have focused clusters of dispersed nodes. Examples of all three high level distributions can be seen in Fig. 2. It is important to realise that the placement of nodes may not result in a connected network. There will often be a requirement for nodes to simply relay information without having any sensing functionality themselves. Part of this research will be dedicated to optimising the placement of, and reducing the need for, relay-only nodes.

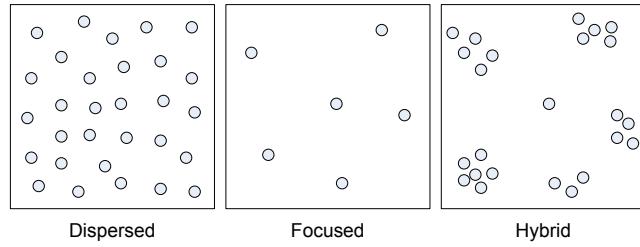


Fig. 2. High Level Distributions

Low level distribution refers to the geometric topology of the network. This is an area that we intend to research more in-depth. We have identified that there are two types of geometric topology, ‘regular’ and ‘random’. Regular distributions include such topologies as those based on grids or cells. Random distributions consist of all topologies that have some element of randomness to their construction. The popular normal and uniform distributions are obvious choices, but there exist other variations that are yet to be identified. Some of these distributions can be seen in Fig. 3.

3.1 Research questions

The key question for this research is:

- How can wireless sensor network deployment be simplified and/or automated, particularly for agricultural applications?

However three other questions are also of interest to answer the above question.

- What physical distributions for wireless sensor networks exist?
- How can topology awareness benefit wireless sensor networks?
- How can this knowledge be incorporated into some protocol and/or management system?

So far we have focused on physical distributions and this is what we discuss next.

4 Defining physical topologies

We have identified several topologies so far; we will possibly identify more in the near future. Fig. 3 shows four of these topologies; Fig. 3(a) and 3(b) use the unit circle as a basis for their construction. A unit circle has a radius of 1 and is centred at the origin (0,0). The unit circle can then be scaled as required.

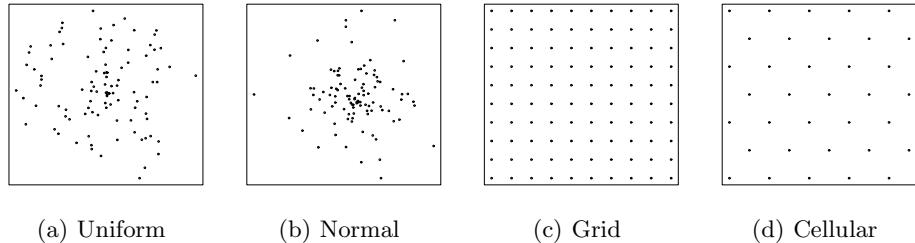


Fig. 3. Common topologies

Fig. 3(a) models the uniform unit circle topology; this was implemented by rotating uniformly about the origin and placing a point at a uniform radius. Fig. 3(b) is very similar but instead places a point at a normal distance from the origin. This is the sort of topology we would expect to see when modelling users of a wireless access point, as users prefer to be close to the access point in order to get better signal quality.

The other two topologies shown in Fig. 3 are the grid topology (Fig. 3(c)) and the cellular topology (Fig. 3(d)). We suspect that a grid topology is too regular to model an ad-hoc network, but it may be suitable for modelling infrastructure-based networks (in a city for example). The cellular topology represents the ideal layout for cellular networks (hence the name); each cell is hexagonal to approximate the circular coverage of a cellular transmitter which allows maximum coverage with some overlap for handover.

An interesting piece of analysis we have conducted is brute force placement simulation. Brute force implies that using a particular distribution, we keep placing nodes until we obtain a connected network. We do this to prove that ‘random’ placement is an inefficient way of achieving connectivity in a wireless sensor network, where we define connectivity as ‘there exists no more than one subgraph for this set of nodes’. For the purposes of this analysis, using simulation techniques, we placed nodes in an area of 10 by 10 units. Each node had a maximum transmission range of 1 unit. Nodes were generated until either connectivity was achieved or the number of nodes exceeded 2500.

Six distributions were tested in this simulation. Each distribution was evaluated with 100 runs for exponentially-increasing numbers of nodes. Only when all 100 runs reached connectivity did the simulation end. We used the uniform and normal distributions discussed above (uniform, uniform unit circle, normal and normal unit circle). We also took the uniform and uniform unit circle, and applied a heuristic that nodes must be at least 0.2 units apart. These are called uniform disc and uniform unit circle disc. Regular networks are connected by their very nature, and hence were not included in the simulation. The normal distributions behaved as expected and did not reach full connectivity after 1000 nodes.

The four uniform based distributions all reached full connectivity in under 1000 nodes, and this is where our interests lie. As can be seen in Fig. 4, the distributions based on the uniform unit circle initially require fewer nodes, but with respect to connectivity, all uniform-based distributions are fully connected at about 400 nodes. This reinforces our theory that ‘random’ placement of wireless sensor nodes is inefficient, and that we need to take a more structured approach to the placement problem.

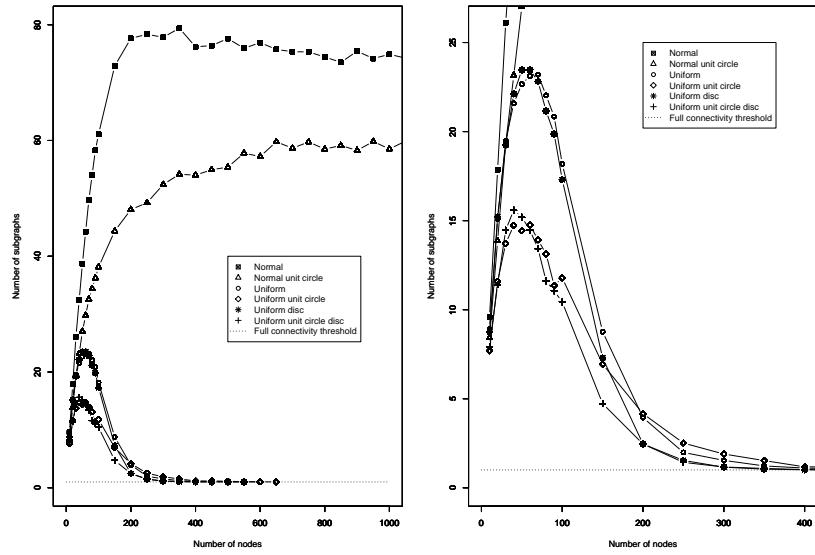


Fig. 4. Brute force node placement (full graph on left, zoomed-in version on right)

5 Further work

The main goal of this research is to develop a custom-built mapping system that allows for simplified/automated wireless sensor network deployment, in particular for agricultural applications. It should provide 2D and 3D terrain modelling with the ability to place both wireless nodes and other environmental objects. Environmental features will include vegetation such as trees and bushes, as well as lakes and streams. The system should also have support for placing man-made structures such as buildings, pylons and fences. This system will be implemented using existing open-source software and modified to meet requirements.

The system will then be able to analyse the network design by using various mathematical models (such as fresnel zoning and signal strength calculation) to verify links in terms of quality. The system should be able to propose amendments should one or more links fail verification.

The idea behind this system is to reduce the levels of expertise required to deploy wireless sensor networks. The system will ensure that sufficient levels of redundancy are put in place to maintain throughput and reliability requirements in case of node failure.

We will also endeavour to discover how topology awareness benefits wireless sensor networks and make recommendations for (and possibly develop) a topology-aware wireless sensor network protocol. This will involve looking at how nodes choose their neighbours and how the physical terrain and placement of nodes affects their communication.

6 Summary

This paper has introduced the concepts of what a wireless sensor network is and how it works. It has also discussed some example applications of WSNs, giving an idea of WSN potential. The direction of this research was examined and it was identified that the key focus of this research is to develop a simple/automated system for wireless sensor network design, specifically for use in agricultural applications. We showed by simulation that brute force placement of nodes is inefficient and that greater care needs to be taken when placing nodes. The paper concluded with a section on further work, identifying ideas and techniques that will be further researched and developed.

References

1. Cardei, M., Wu, J.: Energy-efficient coverage problems in wireless ad hoc sensor networks. *Computer Communications, special issue on Sensor Networks*
2. Wang, J., Zhong, N.: Efficient point coverage in wireless sensor networks. *Journal of Combinatorial Optimization* **11**(3) (May 2006) 291–304
3. So, A.M., Ye, Y.: On solving coverage problems in a wireless sensor network using voronoi diagrams. *Lecture notes in computer science (Lect. notes comput. sci.) ISSN 0302-9743*
4. Chong, C.Y., Kumar, S.P.: Sensor networks: evolution, opportunities, and challenges. *Proceedings of the IEEE* **91**(8) (2003) 1247–1256
5. Kahn, J.M., Katz, R.H., Pister, K.S.J.: Next century challenges: Mobile networking for "smart dust". In: *International Conference on Mobile Computing and Networking (MOBICOM)*. (1999) 271–278
6. Zhu, Y.W., Zhong, X.X., Shi, J.F.: The design of wireless sensor network system based on zigbee technology for greenhouse. *J. Phys.: Conf. Ser.* **48**(1) (2006) 1195+
7. Intanagonwiwat, C., Govindan, R., Estrin, D.: Directed diffusion: a scalable and robust communication paradigm for sensor networks. In: *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, New York, NY, USA, ACM Press (2000) 56–67
8. Holler, M.: Camalie net <http://camalie.com/WirelessSensing/WirelessSensors.htm>.
9. MAF: Contribution of the land-based primary industries to new zealand's economic growth. Technical report, Ministry of Agriculture and Forestry (September 2003)

Optimised Transparent and Automated Handoff between WLAN and WWAN

Mayank Keshariya, Ray Hunt

Department of Computer Science and Software Engineering,
University of Canterbury, Christchurch, New Zealand
{mke38, ray}@cosc.canterbury.ac.nz

Abstract. The combination of 3G and WLAN wireless technologies offers the possibility of achieving anywhere, anytime, always best connected services, bringing benefits to both end-users and service providers. The motivation for these heterogeneous networks arises from the fact that no single technology can provide ubiquitous coverage and high throughput across all geographical areas. Also, the mobility requirements of mobile users changes with various scenarios. Such users typically want to connect to the public or private networks most convenient to them at the time of connection. However, every switching between interfaces may result in the loss of data packets resulting in network congestion with additional load on the network traffic.

We present a versatile mobility solution, which accommodates different interfaces with different levels of security and authentication, and could be deployed as and when required. We have also proposed and analysed an automated algorithm for optimised handoff to integrate different heterogeneous wireless networks. Introduction of link quality, hysteresis effect and dwell timers into the interface selection algorithm optimises the handoff initiation time as well as selection of the most optimal network. We present performance analysis to validate our architectural approach.

Keywords: WLAN, 3G, CDMA2000, Mobile IP, Heterogeneous networks, vertical handoff, hysteresis effect, dwell timers, decision time algorithm, mobility solutions.

1. Introduction

Wireless local-area networks (WLANs) based on the IEEE 802.11 standards [1], also known as WiFi, support reasonable high data rates (from 1 Mbps to 54 Mbps) but are limited to small geographical areas, although this makes them ideal for public hotspot usage. 3G networks [2], offers comparatively expensive and slow speed (64 Kbps to nearly 2 Mbps) communication but over a significant geographical area.

Given the complementary characteristics of both these technologies, it is compelling to combine them to achieve *always best-connected* services i.e. essentially continuous communication via the interface that offers the best price/performance characteristics. It will allow 3G service providers to economically offload data traffic from WAN spectrum to WLAN in indoor locations, hotspots and other areas with high density of users. For WLAN service providers, this integration will bring them a lar-

ger customer base from partner 3G networks, without having to win them through per-customer service contracts. Also, the customers will benefit from enhanced performance in the form of greater coverage, higher data rate, and lower overall cost of such a combined service.

We present a versatile mobility solution, which accommodates different interfaces with different levels of security and authentication, which could be deployed as and when required. We have also proposed and analysed an automated algorithm for optimised handoff between heterogeneous interfaces to integrate different wireless networks.

2. Handoff Issues in Heterogeneous Networks

The Internet Engineering Task Force (IETF) has defined Mobile IP (MIP) protocol [3] to enable IP-layer handoffs during ongoing Internet sessions. Specifically, this is a routing protocol, which has a very specialised purpose, to allow IP packets to be routed to mobile nodes (MN), which could potentially change their location very rapidly. MIP is an integral part of the 3G specification, and therefore the prospect for WLAN integration becomes feasible.

Introducing MIP allows MN to handoff¹ from one access interface to another, thus developing a link between the two technologies, referred as Loosely Coupled approach [4]. However, the architectural issues related to methodology, message exchanges, control signals, and software/hardware elements involved in rerouting the connection becomes more challenging especially if the MN is moving between different interfaces, homogenous and heterogeneous architectures, and/or performing Horizontal and Vertical Handoff [5].

Several other issues must be resolved when a user roams from a WLAN network to a 3G network. Firstly, there must be a secure mechanism through which the new provider can authenticate the user. Secondly, when switching occurs, a user may have several ongoing network sessions which should be *transparently* maintained. Thirdly, the new provider should be able to honour the service level, such as QoS, which the previous provider has guaranteed; and lastly, the accounting and billing infrastructures of both the providers must be interfaced to enable periodic revenue sharing and settlement to generate a common bill.

3. How to Handoff: Heterogeneous Network

We have implemented a system to investigate the performance of handoff between 802.11 WLAN (Home and Foreign Network) and CDMA2000² interfaces, paying

¹ Handoff is the mechanism by which an ongoing communication between the mobile node (MN) and a correspondent node (CN) is transferred from one point of access in the network to another

² CDMA2000 is a family of 3G mobile telecommunications standards. Although our experiments were based on CDMA2000 1x, it could be applied to any 3G access technology.

particular attention to factors such as handoff delay and overall throughput, which influence both the end user and service providers.

3.1 Handoff Delays and Optimisation

We have optimised the handoff *scripts* (shell commands running over mobility daemons), which considerably decreases the handoff latencies thus reducing the number of dropped packets. Table 1 shows the resulting improvement in performance.

Timing Scales (sec) of MN while roaming through Foreign network back to Home network	Original Delays	Avg. Packets Dropped (Ping packets from Goggle)	Delays after using scripts	Avg. Packets Dropped (Ping packets from Goggle)
Registration Time (Register to Foreign Agent)	27.6688 30.3701 40.2544	6	0.5364 0.7899 1.4422	2
Deregistration Time (Deregisters with Home Agent after Returning home)	63.2830 63.4149 69.7647	1	2.0650 6.9018 9.2924	0

Table 1: Handoff latencies and dropped packets while the MN roams between WLAN and CDMA Network.

4. When to Handoff: Heterogeneous Network

In-spite-of any handoff procedure to be as seamless as possible, however, with every handoff, users may lose data packets and unnecessary congestion control measures may come into play [6]. Thus, for any heterogeneous handoff, an efficient Decision Time algorithm will try to use the services of the WLAN as long as possible and perform any handoff to the CDMA network as the last alternative (transmitting at 11 Mbps for 1s is preferable to transmitting at 144 Kbps for 76 s) [7].

4.1 Decision Time Algorithms

The performance and metrics used by different decision time algorithms have been investigated in order to make the correct decision to handoff [8]. The algorithms can be evaluated in terms to (1) discovering the optimal network for the mobile node and (2) the optimal time to make a handoff. The performance of handoff algorithms is determined by their positive effects in reducing handoff rates (number of handoffs per unit of time), and minimising the number of unnecessary handoffs. Other criteria include maximising throughput, minimising latencies and algorithmic calculations.

4.2 When to handoff process - Handoff Initiation

Handoff initiation is the process of monitoring the current network connection, recognising the need for handoff and subsequently initiating it. The criteria used reflects the condition of the current network connection such as signal strength, coverage area, threshold, priority, hysteresis, dwell time and perceived QoS.

At any given time, the mobile node selects one of its physical interfaces as its *current* interface and registers with the mobility agent on that interface. To avoid data loss, it maintains association with the *current* interface while probing for an alternate *better* interface (Figure 1).

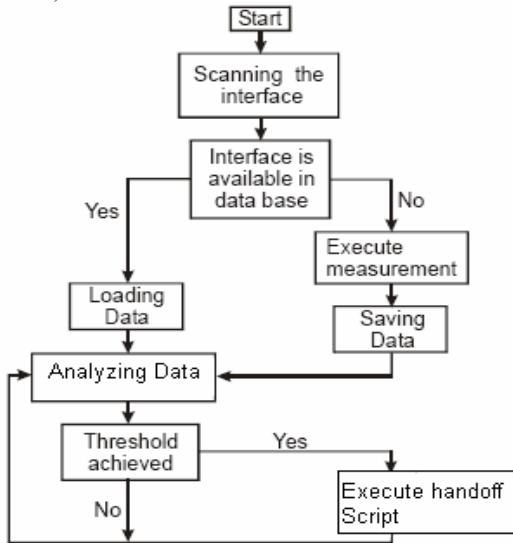


Figure 1: Interface Selection Flowchart

Any handoff decision is based on the selection of optimal available network (section 4.3) and the selection of optimum time to handoff (section 4.4) criteria.

4.3 Optimal Network Selection

The network selection [9] stage is used to select a network connection that can satisfy the requirements of the network provider and the user, such as low cost, good signal strength, optimal bandwidth, low network latency and high reliability.

Considering different network parameters, we have assigned different priorities to each interface. For the local intranet, WLAN has the highest priority, while the Foreign WLAN/Hotspot is assigned the next priority. CDMA2000 network has been assigned the lowest priority due to reduced throughput and higher cost. This scheme allows users to express policies on what is the best wireless system at any moment and makes tradeoffs among network characteristics and dynamics such as cost, performance and power consumption.

4.4 Optimum Time Selection

In addition to available signal strength, we have introduced some parameters for optimal time selection to handoff.

4.4.1 Handoff Latency

Handoff latency is defined as the period when the MN cannot receive application traffic during handoffs. Thus, using the link that has the lowest RTT (round trip time) is preferable in order to minimise the handoff latency. WLANs—even for unfavourable radio conditions—provide a higher throughput than CDMA. It is also beneficial to send the registration request for global mobility from the WLAN interface.

4.4.2 Threshold

Assuming CDMA2000 provides ubiquitous range of approximately the same signal strength everywhere, we calculated threshold values for WLAN (WiFi) to maintain acceptable working connection values. Considering the figures from the minimum required signal strength for WLAN connections to operate practically is 40% of the highest transmission power. Thus for -38dBm signal strength up to -94 dBm is an acceptable value [10].

4.4.3 Hysteresis Effect

Our goal was to introduce a hysteresis effect and let the MN stay with the current interface as long as possible, so as to prevent unnecessary oscillation. A hysteresis margin [11] of 10dBm is introduced, which is added to the signal strength of the current interface been used. Thus for any handoff, it is required that the new interface signal strength must not only be greater than the acceptable threshold values, but also the hysteresis margin. This procedure counteracts any spurious rise in the signal strengths, which generally is caused by the unreliability in the wireless network.

4.4.4 Dwell Timers

Whenever all the conditions for performing any handoff become true, generally a handoff should be performed. We start a timer at that instant and if the conditions for handoff are still true when the timer expires, the handoff is performed. For probing signal strengths, the optimal dwell timer was empirically calculated to 500ms, if AP beacon signals are transmitted periodically for 100ms, as in our case.

4.5 Proposed Algorithm

We have employed an interface-selection algorithm that uses the current signal strengths, threshold levels and the priority of respective interfaces to select the active interface. The priority is a numerical representation of factors such as network bandwidth, cost, overall throughput, network latency and reliability. Variables considered in the algorithm are normalized signal strength, priority, low threshold and high threshold. We denote these as $s_i, L_i, H_i \in [0,100]$, and $p_i \in \{1,2,3\}$ [5]. The mobile

node periodically computes the weight w_i for each interface i , and switches to the interface that has the highest weight.

Formulae:

If i is the current interface,

$$w_i = \frac{1000 * p_i + 2s_i + H}{2s_i + H} \quad \begin{array}{ll} \text{if } s_i \geq L_i \\ \text{if } s_i < L_i \end{array}$$

If i is not the current interface,

$$w_i = \frac{1000 * p_i + s_i - H}{s_i - H} \quad \begin{array}{ll} \text{if } s_i \geq H_i \\ \text{if } s_i < H_i \end{array}$$

where the normalization used is

Normalized(x) = $(x-L)/(H-L) * 100$, for range [0, 100].

4.6 Implementation

We have implemented the proposed algorithm (Figure 2), which continually observes all the handoff parameters, and provides the facility for a user to execute his own commands thus allowing manual overridden handoffs. If any of the parameters falls below an acceptable range, handoff is performed.

Our optimised algorithm considerably decreases the handoff latencies with a reduced number of handoffs, thus reducing the number of dropped packets. Figure 3 shows the resulting improvement in performance.

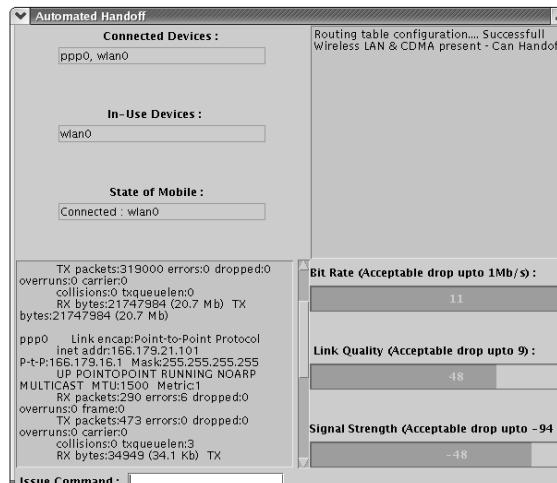


Figure 2: GUI showing different available interfaces and presently activated interface

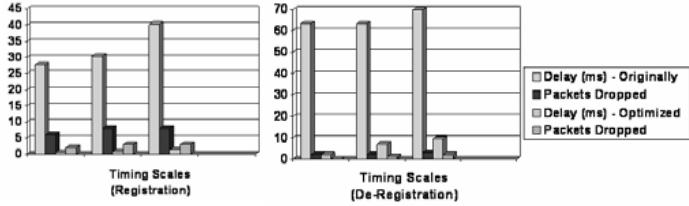


Figure 3: Performance analysis of proposed system compared to previous implementations

5. Versatile Mobility Solution

To integrate different proposed modules described in previous sections, we have introduced the concept of WLAN Gateway to provide mobility services to MN and Intelligent MN Client to cleverly make use of offered services.

5.1 Deployment of WLAN Gateway

We have deployed a WLAN gateway, which integrates different sub-systems, such as MIP-HA, MIP-FA, Local AAA Server, RADIUS AAA proxy, IPSec services, and provides following functionalities:

- Simultaneous support for Mobile IP and Simple IP operation.
- MIP FA/HA complaint with IETF standards
- AAA Radius Client and Server support
- Authentication MIP Authentication by FA and Proxy for 802.1x EAP
- Authorisation enforce policy obtained from HAAA Server on Local Network
- Mobile VPN Capacity: IPSec capacity
- Dynamic packet Filtering/ Firewall

5.2 Intelligent Mobile Node Client

To support mobility services across WLAN and WWAN network requires intelligent client software that can perform MIP signalling with HA and IPSec. Such client requires intelligent selection of the best interface available while probing for the other better interface, exchange messages to authenticate and enable required security features. Decision based Algorithm for selection of the most optimal network for a particular interval and optimal time to handoff are also incorporated into the MN.

The development of MN Client includes:

- Seamless Intra- and Inter- Technology handoff using Mobile IP
- Management of Multiple physical interfaces (802.11, 3G/PPP, Ethernet)
- Sophisticated movement detection
- Automated network selection algorithm based on priority, signal strength and preferred network list

- Mobile VPN capacity: IPSec over Mobile IP

6. Conclusion

In this paper, we have presented an overview of the issues related to handoff with particular emphasis on heterogeneous networks and presented a concept of a versatile mobility solution incorporating different interfaces. The process is categorised into *How to handoff* and *When to handoff*, considering handoff in optimised timeline towards most preferred network. The system is designed to be automated or it can be overridden manually with user's preferred preferences. We have described, implemented and analysed various integrated service scenarios that formed the basis of our work.

Handoff issues can be classified as two independent parts: selecting the most optimal network presently available, i.e. network selection, and the correct time to initiate a handoff, i.e. handoff initiation. We have proposed an automated algorithm for optimised handoff between local WLAN, foreign WLAN (hotspot) and WWAN interfaces. The interface selection algorithm is based on the parameters such as current signal strength, interface priority and throughput. Introduction of link quality, hysteresis effect and dwell timers are used for optimised performance. The presented performance analysis validates our architectural approach.

References

1. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. ANSI/IEEE Std 802.11: 1999 (E) Part 11, ISO/IEC 8802-11, 1999.
2. TIA/EIA/IS-835B – CDMA2000 Wireless IP Network Standard, Third Generation Partnership Program 2 (3GPP2), 2000.
3. C. Perkins (Editor), IP Mobility Support for IPv4, RFC 3220, IETF, January 2002.
4. Park, H., Yoon, S., Kim, T., Park, J., Do, M. and J. Lee, Vertical Handoff Procedure and Algorithm between IEEE 802.11 WLAN and CDMA Cellular Network, Lecture Notes in Computer Science (LNCS), 2003.
5. Buddhikot, M., Chandranmenon, G., Han, S., Lee, Y.W., Miller, S. and L.Salgarelli, Integration of 802.11 and Third-Generation Wireless Data Networks. IEEE Infocom, 2003.
6. Caceros, R., and L. Iftode, Improving the Performance of Reliable Protocols in Mobile Computing Environments, IEEE JSAC, June 1995.
7. Stemm, M., and R. H. Katz, Vertical Handoffs in Wireless Overlay Networks, UC Berkeley Computer Science Division, Report No. UCB/CSD 96/903, May 1996.
8. Pollini, G.P., Trends in Handover Design, IEEE Communications Magazine, March 1996.
9. Berezdivin, R., Breinig, R., and R. Topp, Next-Generation Wireless Communications Concepts and Technologies, IEEE Communications Magazine, March 2002.
10. J. Lindeman. WLAN Design - Surveying the Wireless LANscape, Mobile and Wireless Technology Workshop, Jan 2004.
11. Pahlavan, K., Krishnamurthy, P., Hatani, A., Ylianttila, M., Makela, J., Pichna, R., and J. Vallstom, Handoff in Hybrid Mobile Data Networks, IEEE Personal Communications, April 2000.

An Improved Encryption Key Management System for IEEE 802.16 Mesh Mode Security Using Simulation

Alastair Nisbet

*Institute of Information & Mathematical Sciences
Massey University at Albany, Auckland, New Zealand
a.j.nisbet@massey.ac.nz*

Abstract. Wireless networks have enjoyed explosive growth since their introduction in the late 1990s. However, of the various forms of wireless protocols available, only a few have become universally accepted. One reason for the ubiquity of one protocol over another is the perceived robustness of the security built into the protocol. As WiMax is poised to be unleashed worldwide within the next 12 months, already security concerns are beginning to appear. My proposed research will closely examine the encryption key management protocols of 802.16 used in mesh networks, and using simulation show that there are indeed some security flaws in the protocol. My research will then involve using simulation to model modifications to the existing protocols and show that my design may be used to increase the security effectiveness of the protocol and answer my research question: "What constitutes an effective but efficient solution to the security problems in 802.16 mesh networks?

Keywords: WiMax, Security, Simulation, 802.16, Information Systems, Network Research

1 Introduction

Whilst many households in developed countries have access to the Internet, many rural areas and developing countries do not enjoy the same level of access. The pervasiveness of the Internet has now become an accepted and necessary part of our daily lives, and as the sophistication of web sites increases, the lure of high-speed Internet becomes more attractive. Broadband Internet has enjoyed fairly rapid growth over the past few years. As speed increases and costs drop, more people are switching to broadband. In June 2005, New Zealand ranked 22nd in the OECD countries with 6.4% of the population getting broadband [1].

In a country like New Zealand where much of the population lives in rural areas and small towns, broadband may be unavailable due to the technical difficulties of supplying high speed lines that are distant from telephone exchanges. For those unable to get true broadband through their phone lines, the delivery of broadband through radio waves is a viable alternative. Many companies worldwide market their own proprietary versions of wireless broadband. However, consumers have shown themselves reluctant to engage these proprietary offerings from smaller companies. The time is right for a worldwide standard of broadband to hit the markets, and this is what WiMax is set to do. WiMax, or more properly IEEE 802.16 is in fact several protocols developed to deliver high-speed, last-mile wireless broadband. These protocols can be divided into two basic varieties, fixed and mobile.

The focus of my research will be the mobile WiMax version of 802.16 used in mesh mode. This protocol allows users with mobile devices to communicate directly with each other, or use intermediate stations as hopping points to communicate with other stations outside the range of the initiating station. By using a WiMax network in mesh mode, subscriber stations can connect indirectly to base stations through using other subscriber stations as intermediaries. This topology has the ability to greatly increase the availability of the network to greater numbers of subscribers.

However, with ease of use of wireless technology comes the difficulty of securing data transmissions over the airwaves. Indeed, much of the early reluctance to install WiFi wireless networks can be attributed to security concerns and the poor publicity that surrounded the initial versions of security protocols built into the IEEE's 802.11 standards. A recent survey of New Zealand organizations found that 48% of those surveyed claimed the major reason for non-consideration or non-deployment of wireless networks in their organisations was a perception of low levels of security [2]. For WiMax to avoid this reluctance shown to WiFi, and for WiMax to realise the benefits to individuals and organisations that many proponents are promising, the protocols must be shown to communicate data securely [3].

Due to the broadcast nature of wireless communications, privacy is a major issue for any user of the technology, and whilst the WiMax consortium are working hard to ensure the implemented security protocols will be the best available, history has shown that security flaws will always exist to a greater or lesser extent. The initial design of WiMax was classified as a point to multipoint topology. However, two major revisions have been made to the standards which are incorporating a variety of other topologies. The main revisions are 802.16e which allows mobility of subscriber stations and seamless handoffs to other stations at up to 120kmh, and mesh mode which allows individual subscriber stations to communicate directly with each other without a central controller such as a base station. This mesh mode network can involve only WiMax subscriber stations or can incorporate a 802.11 WiFi network to allow communications from the WiFi network to a base station for Internet or other network access [4].

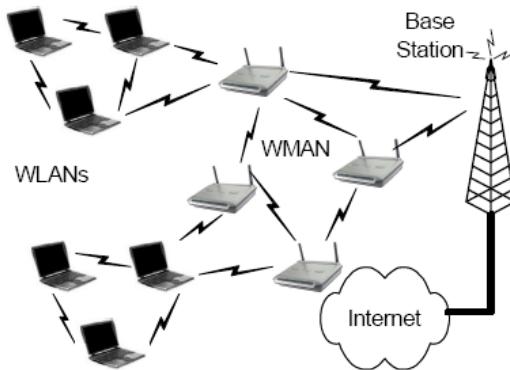


Figure 1: Mesh network [5]

With data communications travelling through the airwaves, interception by unauthorised eavesdroppers must be assumed to occur. Whilst it is impossible to prevent these eavesdroppers from intercepting communications, encryption can prevent the data from being understood. Encryption relies on encryption keys being used to encrypt the data before being sent, and decrypting the data when received. The problem of securely distributing, updating and revoking encryption keys is a complex one, but an effective key management protocol ensures security of communications and all the benefits that go with data confidentiality.

The aim of this paper is to outline a research approach suitable for identifying, modelling and evaluating the 802.16 mesh mode security key management protocols, and show that not only are attacks possible, but that an improved design is available through my research into this problem. Simulation is chosen as the research approach since it provides a cost-effective way to investigate the long-term behaviour of wireless networks under varying scenarios using varying patterns of communication.

2 Previous Work

The IEEE 802.11 standard has seen a lot of research into ad hoc topology (mesh mode), and with its considerably longer range 802.16 is expected to see considerably more. The IEEE therefore incorporated mesh mode specifications into their 802.16-2004 revision [6]. A wireless network operating in mesh mode has several problems to overcome specific to that type of topology. Firstly there is the ad hoc nature of the network where it may be temporarily created. Secondly, the dynamic nature of the network where communicating stations may join and leave the network several times during its lifetime mean that authentication and revocation of authentication is an on-going process, and thirdly the geographic spread of the network where communicating stations may have to rely on intermediary stations to reliably and securely pass on messages in hops until the intended recipient receives the message. Many of these difficulties must be overcome before the network can be said to be effective and secure.

The 802.16 protocols are based on DOCSIS (Data Over Cable Service Interface Specification). This protocol was designed specifically to solve the problem of delivering last mile data over cable [7]. The IEEE 802.16 MAC (Medium Access Control) defines a PKMv2 (Privacy Key Management version 2) sublayer providing device/user authentication, key management and data traffic privacy [8].

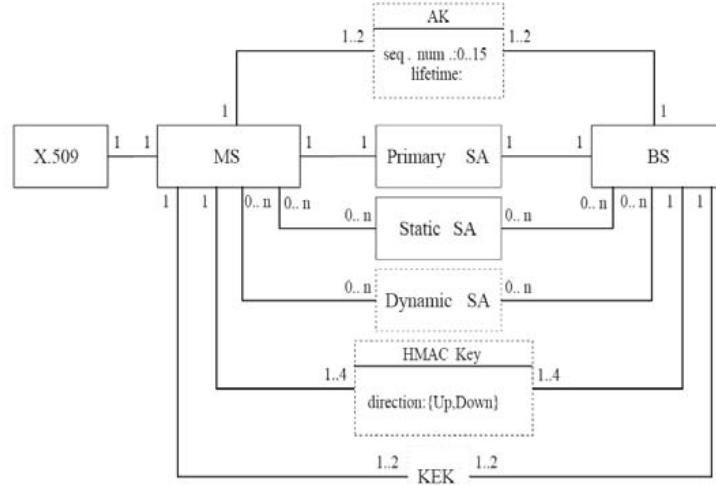


Figure 2: Security Model [9]

However, the 802.16 MAC is connection-oriented, meaning a SA (Security association) is maintained between a SS (Subscriber Station) and a BS (Base station). The SS searches for MSH-NCFG messages to synchronise with a network and build a list of available BSs and neighbouring SSs. In figure 2, the various security associations between the MS (Mobile Station) are shown. The MS (subscriber station) has a X.509 certificate installed by the manufacturer which is used for authentication using public key infrastructure (PKI). This initial authentication message forms the basis for the derivation of the other encryption keys used in data and management message exchange.

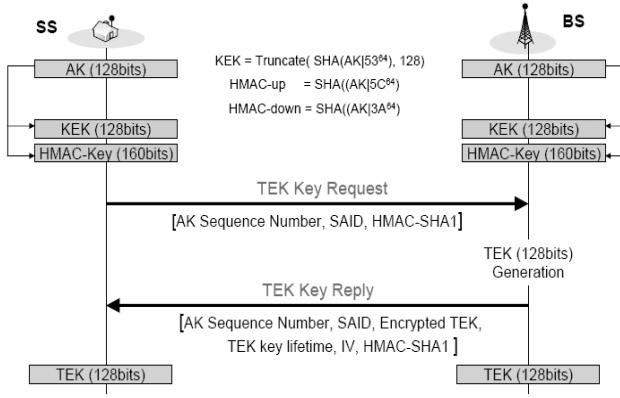


Figure 3: Encryption Key Exchange [10]

Figure 3 shows the communications between the two stations which rely on three security associations and four encryption keys. These keys are the Authentication Key (AK), the Transport Encryption Key (TEK), the Hashed Message Authentication Code (HMAC) and the Key Encryption Key (KEK). The nature of mesh networks means that neighbouring stations play an important role in assisting a new SS to join the network, and are relied on to pass authorisation messages between the new station and the BS until the station has officially joined the network. It is during these authorisation messages and key exchanges that most attacks can occur.

There have been several identified attacks that a mesh network is vulnerable to. Among these are sinkhole, wormhole and authentication attacks. In a sinkhole attack, a rogue station spoofs routing information to trick communicating stations into sending information through the rogue. In a wormhole attack, two rogue stations tunnel information from communicating stations that are distant so that they believe they are in fact neighbours, and in a replay attack, authorisation requests from innocent stations are intercepted and replayed to the base station. As there is currently no replay protection built into the proposed standard, the base station will not realise that this is the same as a previous authorisation request and will accept the message as genuine, allowing the requesting station to join the network. Effective encryption key management can prevent all of these attacks.[5].

During the initial protocol designs, many of the possible threats to mesh networks were identified. Among the most likely and those of greatest impact to a network were BS and SS masquerading and management message modification [9]. The following table shows the threat analysis of several of the possible attacks on mesh networks carried out by Barbeau in 2005. The scale runs from 1 to 9 where 1 is low and 9 is high.

Table 1: Threat Analysis. [9]

Threat	Type	Likelihood	Impact	Risk
Jamming	Availability	3	1	3
Scrambling	Availability	2	1	2
Eavesdropping of management Messages	Confidentiality	3	2	6
Eavesdropping traffic	Confidentiality	1	1	1
BS or MS masquerading	Confidentiality	3	3	9
Management message modification	Integrity Availability	3	3	9
Data traffic modification	Integrity	3	1	3
Denial of Service of BS or MS	Availability	3	3	9

As can be seen from this table, eavesdropping, masquerading and management message modification are amongst the highest risk of attack. These types of attacks can all be prevented by having a secure encryption key management system that: 1) Encrypts traffic securely so that only the intended recipient can decrypt the message 2) Only permits authorised stations to join the network 3) ensures integrity of traffic by showing whether data has been modified or not. Following the identification of possible attacks to the newly developed protocols, a host of security improvements were suggested including mutual authentication of the SS and BS, better scalability to allow mass marketing of WiMax, and secure handoffs to other stations [11]. However, whilst suggested, it is unlikely that WiMax will see all of these suggestions implemented in the final standards. Therefore, it is likely that when ratified, the 802.16 standard for mesh mode networks will remain with several already identified flaws and several that will come to light in the future. My proposed research will look at the identified security problems with the standard specific to mesh networks and will endeavour to correct at least one of those problems by using computer simulation to develop an improved encryption key management system.

3 Research Method

Simulation is a standard research technique for examining the long-term behaviour of complex, dynamic systems in both engineering disciplines and the social sciences. Information systems deals with the interactions between human actors and computing systems but due to the immaturity of the field in terms of theory development, simulation has been used less frequently relative to surveys, case studies and lab experiments (Vitolo & Coulson 2004). However, simulation is essentially a computerized lab experiment over a complex, and possibly dynamic system. In figure 5, the steps comprising a simulation research approach are shown as described by Vitolo & Coulson (2004, p. 250-260).

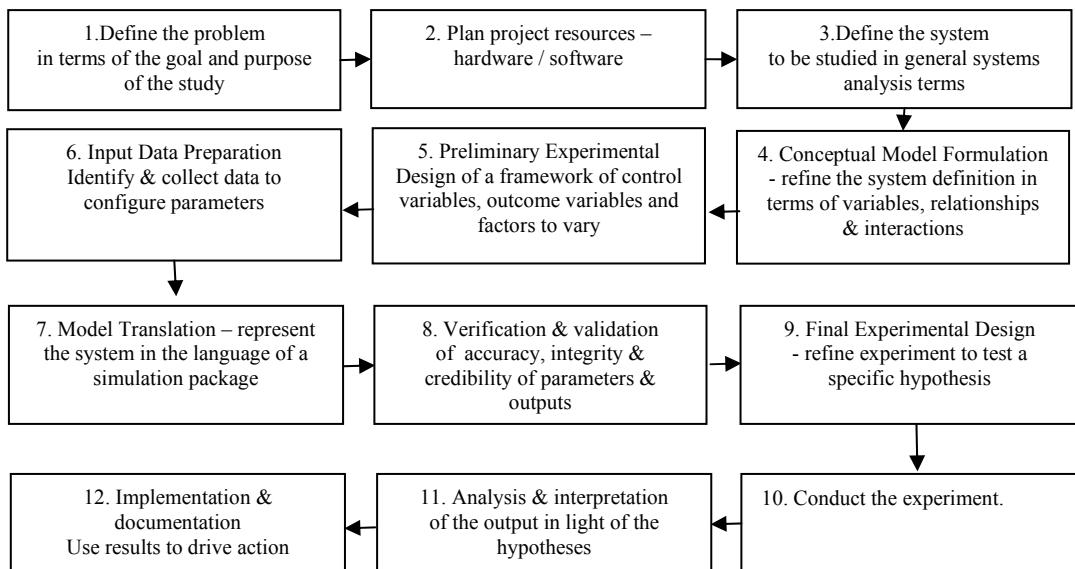


Figure 4: The 12 Steps of the Simulation Research Approach (Pegden, Shannon & Sadowski, 1990, p. 12-13) cited by (Vitolo & Coulston, 2004)

Steps 1 and 2 reflect the statement of the research goal and purpose and project planning with respect to resource needs. The goal of the research is to improve our understanding of 802.16

security in wireless mesh networks. The purpose of the research is to discover how different protocol options influence the security and efficiency of transmitting confidential messages over mesh networks. In terms of resource planning, access to Opnet, a well-known network simulation tool has been provided for this study. Opnet will be used to develop, model and examine the properties of alternative security solutions. The solution developed in this PhD study will be compared to other solutions proposed in the literature via simulation. A single simulation package will be used to ensure that the proposed solution is in fact a more effective design than the IEEE 802.16 and other proposed protocols, as simulations performed on different simulators can provide different results [12]. In terms of data sources, previous simulation studies are currently being examined to establish ranges of input and output variable values. The parameters of the simulation have yet to be finalised. Fixed parameters may include the size of the area of the simulation, the number of subscriber stations and an appropriate routing protocol. Input variables will take account of the dynamics of the network. These will include the number of stations that join and leave the network, whether stations are mobile or stationary, and traffic volumes.

Steps 3 and 4 involve constructing a hierarchical model of the network subscriber and base stations and their probable interactions in order to predict the behaviour of a proposed key management protocol solution and to make comparisons with other existing protocols. The solution is essentially the treatment that influences the outcome of the experiment in terms of the dependent variables of efficiency of a transmission and effectiveness with respect to the security of a transmission. Previous studies will be used to provide data on the ranges of measures used for each of the dependent variables produced under different assumptions with respect to authentication and encryption key management. A conceptual framework representing a solution in terms of input variables, output variables, parameters and interaction logic is then developed.

Steps 5 and 6 involve developing an understanding of the dynamics of the system through an iterative process of generating initial conditions for the simulated mesh network and then driving varying inputs into the simulation (Vitolo & Coulston, 2004). During step 5, preliminary experimental design and explicit measures are established for all variables. Functional and/or probabilistic relationships are established at this time. For example the fraction of time hackers are present could be used to parameterise the type of user in the model. The model outputs at this point are expressed as a function of the input variables and parameters.

In step 7, model translation is performed where the system is translated into the language of the simulation software, Opnet. This software uses a hierarchical structure to modelling. Each level of the hierarchy provides different aspects of the system to be simulated [13]. These levels are the network editor which allows amongst other variables, the physical size of the network, the number of stations (nodes) and the topology. Next is the node editor where data flow models are defined, and finally at the lowest level is the process editor where control flows can be adjusted. This gives thorough control to the researcher as to how detailed the adjustments to the model can be, thus ensuring a meticulous design of the system to be studied.

In step 8, verification and validation of the design is performed. At this stage, previous systems including the IEEE 802.16 protocol are simulated to verify that the results are those expected. If not, a process of refinement occurs until an accurate simulation of the existing systems occurs. Only when this is satisfied can the experiment move on to the next step.

In step 9, the final experimental design is refined to test the specific hypothesis. First, a simulation will be performed on the system to be compared with the newly developed system. In this case, this will be the IEEE 802.16 protocols key management system and any other solutions proposed by other researchers. The results of these simulations will be used as a benchmark for comparison with the subsequent experimental runs on the new system. Once satisfied that the experimental design mimics reality, the range of manipulation of the input variables is determined. Here, the simulation may be run several times with adjustments made to

the input variables. Analysis of the results is done by comparing the outputs of the experiments with the previously set benchmark simulations. Adjustments are then made to the inputs and the simulation is repeated with these adjustments in order to test the hypothesis that the newly developed key management system is more effective than those previously proposed. A cycle may then recur several times where adjustments are continually made to refine the experiment in order to thoroughly test the hypothesis.

In steps 10 and 11, the experiment is run with the refined input variables and statistics are collected on the models' behaviour based on the manipulated variables in order to determine an effective and efficient protocol that will provide the answer to the research question. Finally, in step 12 the direction and magnitude of the relationships between the independent and dependent variables is established and sensitivity analysis can be performed with respect to the parameterised aspects of the new protocol. Full documentation is then performed to ensure the accuracy of the experiment and provide the ability for the experiment to be repeated in the future.

4 Summary and Potential Contributions

At present, the research is in stage 3 of the framework shown in figure 4, where the system to be examined is being determined. The software to be used for the study is Opnet which has a WiMax module currently under development that will be available shortly. This module will allow WiMax network configurations to be simulated and experimented with in a controlled environment.

Already, military use of wireless communications is widespread and considerable interest by the military has already been shown for mesh networking topologies. This includes both battlefield communications and disaster relief efforts such as those of Hurricane Katrina in New Orleans in 2005 [14]. In this case, a WiFi network was quickly deployed allowing personal communications with email, and Internet connectivity for data communications. With WiMax providing much greater transmission distances than WiFi, there are considerable benefits in this area for being able to deploy a secure wireless network citywide within days or even hours. However, security of the data provides both a reliable and trusted communications tool, and this is where the proposed research will provide its greatest benefit. This research will enhance the ability to provide a secure and efficient encryption key management system for mobile mesh networks using IEEE 802.16, and as such will make contributions in the private, public and military arenas where the technology will be used.

References

1. OECD, *OECD Broadband Statistics, June 2005*. 2005.
2. Houlston, B. and N.I. Sarkar, *Wi-Fi Deployment: A Survey of Large New Zealand Organisations*. International Journal of Business Data Communications and Networking, 2005. 1(3): p. 37-58.
3. Richardson, M. and P. Ryan. *WiMax: Opportunity or Hype?* in *Proceedings of the Fourth Annual ITERA Conference 2006*. 2006. Las Vegas: Murray State University.
4. Sweeney, D., *WiMax Operator's Manual: Building 802.16 Wireless Networks*. 2004, New York: Apress.
5. Zhou, Y. and Y. Fang. *Security of IEEE 802.16 in Mesh Mode*. in *2006 IEEE Military Communications Conference (Milcom 2006)*. 2006. Washington DC.
6. IEEE, *IEEE Std 802.16-2004/Cor 1-2005*. 2004, IEEE.

7. Alessi, G., et al. *Adapting the DOCSIS protocols for military point to multipoint wireless links*. in *MILCOM 2000*. 2000: IEEE.
8. Abichar, Z., Y. Peng, and J.M. Chang, *WiMax: The Emergence of Wireless Broadband*. IT Professional, 2006. 8(4): p. 44-48.
9. Barbeau, M. *WiMax/802.16 threat analysis*. in *Proceedings of the 1st ACM international workshop on Quality of service \& security in wireless and mobile networks*. 2005. Montreal, Quebec, Canada: ACM Press.
10. Wongthavarawat, K., *IEEE 802.16 WiMax Security*, in *17th Annual FIRST Conference*. 2005: Singapore.
11. Puthenkulam, J. and J. Mandin, *802.16e Security Adhoc Proposal*. 2003.
12. Cavin, D., Y. Sasson, and A. Schiper, *On the accuracy of MANET simulators*. Proceedings of the second ACM international workshop on Principles of mobile computing, 2002: p. 38-43.
13. Chang, X. *Network simulations with OPNET*. in *Proceedings of the 31st conference on Winter simulation: Simulation---a bridge to the future - Volume 1*. 1999. Phoenix, Arizona, United States: ACM Press.
14. Donahoo, M. and B. Steckler. *Emergency Mobile Wireless Networks*. in *IEEE Military Communications Conference, 2005*. 2005: IEEE.

Building in Web application security at the requirements stage: A tool for visualizing and evaluating security trade-offs

Natalia Nehring

¹ Massey University, New Zealand

natalia.nehring@gmail.com

Abstract. One dimension of Internet security is Web application security. The purpose of this Design-science study is to design, build and evaluate a computer-based tool to support security vulnerability and risk assessment in the early stages of Web application design. The tool will facilitate risk assessment by managers and will help developers discover vulnerability in system requirements by providing a means for calculating potential losses and graphically visualizing risk levels of different system components. This represents a proactive approach to building in Web application security at the requirements stage as opposed to the more common reactive approach of putting countermeasures in place after an attack and loss have been incurred. The primary contribution of the proposed tool is its ability to make known security-related information (e.g. threat trees, countermeasures) more accessible to developers and to translate lack of security measures into potential dollar losses for managers so they can prioritize security spending.

Keywords: Internet security, Web application security, security evaluation, threat trees, risk assessment.

1 Introduction

Deloitte Touche Tohmatsu surveyed the top financial institutions around the world finding 78% had experienced external attacks [4]. In the Asia-Pacific region, excluding Japan, 100 percent of those surveyed reported attacks in 2006 while only 16 percent did so in 2005. This indicates significant growth in Web application security problems despite the existence of countermeasures to prevent or reduce the impact of these attacks. Unfortunately, organizations may not implement known countermeasures because they are not aware of them or think they are too costly. As a result the current practice is mostly a reactive one, where organizations spend on security only after attacks and losses have been incurred.

In contrast to this predominantly reactive practice of detecting and correcting security problems, this project takes a proactive approach. The IT community knows about countermeasures, security patterns, attack patterns and existing vulnerabilities but people are still developing Web applications which are not secure. To mitigate this problem, there is a need to make this information more accessible to developers and to provide realistic ways to assess potential losses so managers can justify spending on security. The history of software development has taught us that finding

and fixing a software problem after delivery can be 100 times more expensive than finding and fixing it during the requirements and design phases [1]. The same principles can be applied to security in Web applications.

Gathering and interpreting available data about vulnerabilities can be an onerous task, taking a considerable amount of a developer's or analyst's time. The old proverb, "a picture is worth a thousand words", implies humans may absorb complex information more readily from pictures than from volumes of text. Visual representations of complex relationships amongst software system components, their vulnerabilities, threats based on these vulnerabilities, and the magnitude of potential losses can quickly pinpoint areas of major concern, facilitating security risk assessment. We still need accurate and reliable products to calculate security quantitatively in order to assess where to improve security [9]. Security should not be treated as an add-on feature, it should be considered early on as a key system requirement. In order to help developers and managers effectively utilise existing knowledge of threats, attacks, vulnerabilities and countermeasures and to translate this information into objective measures of risk for Web application projects, this research has two objectives:

Objective 1: To design and prototype a tool for use by managers and developers for visualizing and evaluating security trade-offs and risks in alternative Web application designs.

Objective 2: To demonstrate the utility of the tool via evaluation in a real Web application development environment. Utility will be measured in terms of user satisfaction with the tool's ability to: 1) model relationships between assets, components, vulnerabilities, threats, countermeasures and risk, and 2) provide access to existing security knowledge.

2 Background

Security assessment is often associated with the concept of risk. Risk can be viewed as a function of the likelihood that a threat will materialise, the level of vulnerability and the potential for loss of resources. Thinking about negative scenarios in these terms is an essential skill for a test engineer [3]. In this sense, a Web application designer should also think about requirements in terms of negative scenarios, that is, from a hacker's point of view. Vulnerability, threats and attacks are relevant to software risk measurement [2]:

- Vulnerability is a characteristic of the software that makes it possible for a threat to occur.
- A threat is as an event which can have an undesirable effect on assets and resources.
- An attack is an action by a malicious user that involves exploiting vulnerabilities in order to cause a threat to occur.

Vulnerability and attacks are only of concern if they introduce the potential for threats that would involve significant resource loss [2]. If you increase any of these three variables, risk also increases. If you reduce them, it decreases. Countermeasures

in the context of Web applications are ways to eliminate vulnerabilities or reduce the impact of an attack.

3 Methodology

Figure 1 summarises the Design-science research approach [5] used in this study. Design-science research focuses on designing, building and evaluating an innovative IT artefact for a particular purpose [6]. This research began with a review of existing tools and risk assessment procedures to identify problems. Next an initial tentative design for a new tool to address these problems was proposed. The research is currently at step three, prototyping the tool (i.e. the artefact) in an iterative fashion in order to make improvements to the initial design. This iterative process involves several evaluations of the tool by security experts at a security company in New Zealand. Performance will be assessed in terms of usability and functionality. When performance is satisfactory, the study ends and conclusions are drawn resulting in implications for future work to extend the tool and for its use in practice.

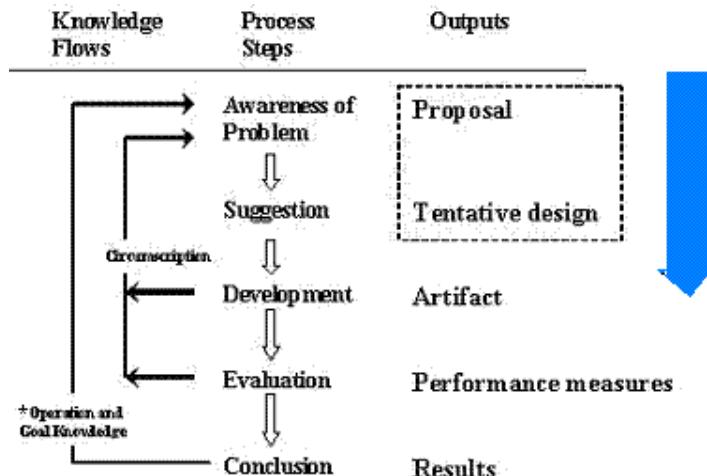


Fig. 1. Steps in the Iterative Design-science Research Process [11]

4 Theory Base for Design

Security-related knowledge should be accessible to developers and managers. The working tool will bring together existing knowledge about threat modelling, evaluation criteria (through threat trees) and a vulnerability database. Extensibility will be incorporated into the design of the tool to allow inclusion of other types of security knowledge in the future.

The tool will calculate risk using Sahinoglu's (2005) model of a decision-tree illustrated in Fig. 2. In this model [9], risk is defined as the possibility of a certain threat being realised by exploiting a particular vulnerability, resulting in a harmful

impact on the system. The system will calculate loss from the lack of countermeasures (LCM_i) using probabilities for vulnerabilities (V_i), threats (T_i), and countermeasures (CM_i) as input. Every event or object that can reduce risk to the system can be defined as a countermeasure (CM). For example, an action, a device, or a procedure can be represented as a CM . When the CM is applied, the risk is reduced, and the remaining risk is defined as residual risk. If a countermeasure can reduce risk completely, the value of residual risk will be zero. New components can be easily added to the system, modelled as shown in Figure 2, and overall system risk recalculated.

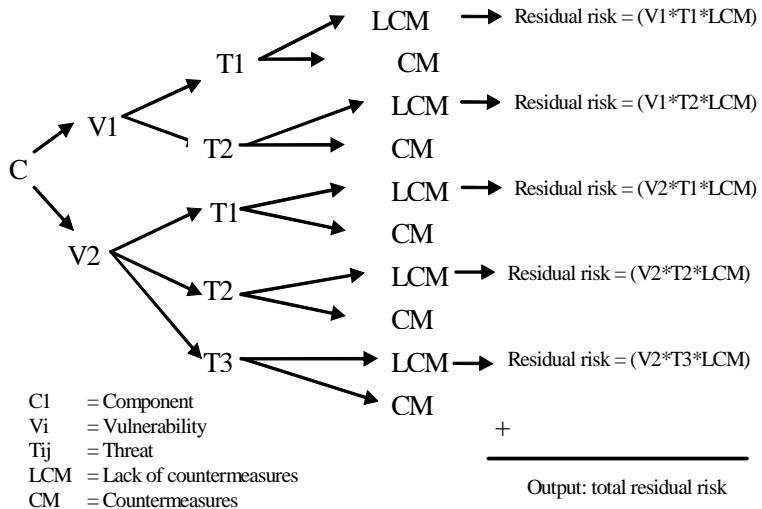


Fig. 2. Security Meter decision tree diagram [9].

The Security Meter algorithm uses a hierarchical tree as shown in Figure 2 to represent the residual risk calculation for each component of a Web application asset (i.e. root node not shown). Each path represents the joint occurrence of vulnerability, a threat and a level of use or non-use of countermeasures. The residual risk for each path is calculated as the product of these probabilistic inputs and then summed over all branches to arrive at the total residual risk for each component. Further details on this algorithm can be found in [9].

Table 1 provides a comparison of key aspects of the proposed tool with two existing tools: the Practical Threat Analysis (PTA) Tool [8] and the Microsoft Threat Analysis & Modelling Tool [7]. The tools differ in terms of the target users, risk algorithm, required inputs, types of reports/visualizations produced, threat modelling processes (steps) and navigation mechanisms. The Microsoft tool provides developers with libraries and templates for modelling security requirements and a wizard to guide the user. The PTA tool can download libraries of threats, vulnerability and assets, but does not provide links between them. One advantage of PTA for managers is measuring current, minimum and maximum risk in dollars. Ideally, a tool should incorporate multiple risk methods, good visualisations, pre-defined security knowledge and could be used by multiple groups of users.

Table 1. A Comparison of the Proposed Tool with Two Existing Tools

	Microsoft Threat Analysis & Modelling Tool [7]	Practical Threat Analysis (PTA) Tool [8]	Proposed Vi-secanto Tool
Intended Users	Developers without web security expertise	Managers & developers, with web security expertise	Managers & developers, without web security expertise
Navigation	Tree view with visibility at all times	Top menu, link menu on side of main page, and bottom of some pages	Top menu, pop-up menu
Input	User roles (N.identities), data, dependencies & other assets(no \$, no ranking), external, threats (risk rating 1-9), attacks, relevancies, use cases, calls, & linkages	Assets (\$), threats (\$), vulnerabilities, countermeasures (cost(\$), mitigation (%), cost.effectiv (%)) & linkages	Assets (\$), threats, vulnerabilities, countermeasures as probabilities, & linkages
Output	Graphs of call, data & trust flow: threat trees, & attack surface Risk rating & response (accept, reduce, avoid, transfer) Reports on inputs. Separate instructions for designers, developers and operations team for implementing countermeasures.	Max, min and current risk in \$'s for each asset. Graphs of top current threats, risk history, analysis history. Reports countermeasures by cost, mitigation level, ROSI (Return on Security Investment) Mitigation steps for target risk reduction Reports on inputs Top threats by current risk level.	Risk in \$'s for each as an interactive tree diagram (see Fig. 4.) Top threats by risk level Reports on inputs. Guidelines for order of countermeasure implementation.
Risk algorithm	Threats Risk = Bus. Impact* Probability Bus.Impact, Probability – qualitative value (Low, Med,High)	Threat's Max.Risk = (\$Asset /100) *\$Damage*NumOccurrence Min.Risk =Max_Risk-(Max.Risk/100)* Mitigation	Sahinoglu, M. Security Meter approach. See section 4.
Security knowledge provided	Modifiable templates: describe and relate attacks, threats, & countermeasures	No templates. Libraries as text files listing assets, vulnerabilities, threats, countermeasures, attacker types, & tags,	Modifiable templates: describe & relate threats, vulnerability, & countermeasures
Information stored	User input, current state of project	User input, history of countermeasure implementation	User input, current state of project

5 Architectural Layers of the Proposed Tool

The proposed tool currently exists as an early prototype called Vi-secanto (VIvisual -SECurity ANalisys TOOl). The architecture of the tool can be divided into three layers of classes as follows (see Figure 3):

1. Graphical user interface layer (i.e. classes that produce input forms, output diagrams)
2. Application layer (i.e. classes that implement the application logic)
3. Storage layer: (i.e. the database and classes to interface with the application layer)

The program was written in Microsoft Visual C#, using Microsoft Visual Studio 2005 .Net. For storage, the Vi-secanto tool utilises the Microsoft Access 2002 database. The database contains ten tables: asset, asset type, component, LCM, settings, threat, threat category, threat sub category, vulnerability, and vulnerability type.

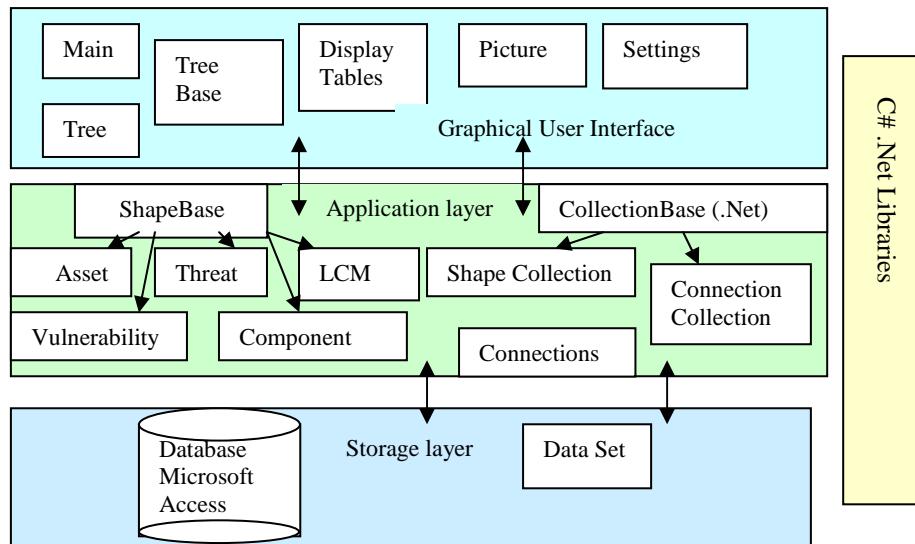


Fig. 3. Architectural design of the Vi-secanto tool.

The Vi-secanto tool contributes to supporting proactive security requirements analysis by providing a visual representation of each asset's threat tree (see Fig. 4). Threat trees can be saved to a database for later retrieval and modification. For example, the threat tree in Figure 4 shows a manager component1 has the highest level of risk (0.23) and an expected loss of \$1841 is possible. These calculations show that this component requires more attention than component2. Component1 has two vulnerabilities. Vulnerability2 has a narrow but high probability range (low=0.8, high=0.9) making it likely to occur. Two threats, threat2 and threat3, can be realized by exploiting vulnerability2. This information tells a manager with a limited budget, which countermeasures should be provided first. It is possible to examine different

scenarios as more information becomes available or to assess various alternatives. What-if scenarios can be explored to support decisions on which countermeasures to invest in first and second and so on. The output shows developers which components need more attention in terms of addressing vulnerabilities through design changes.

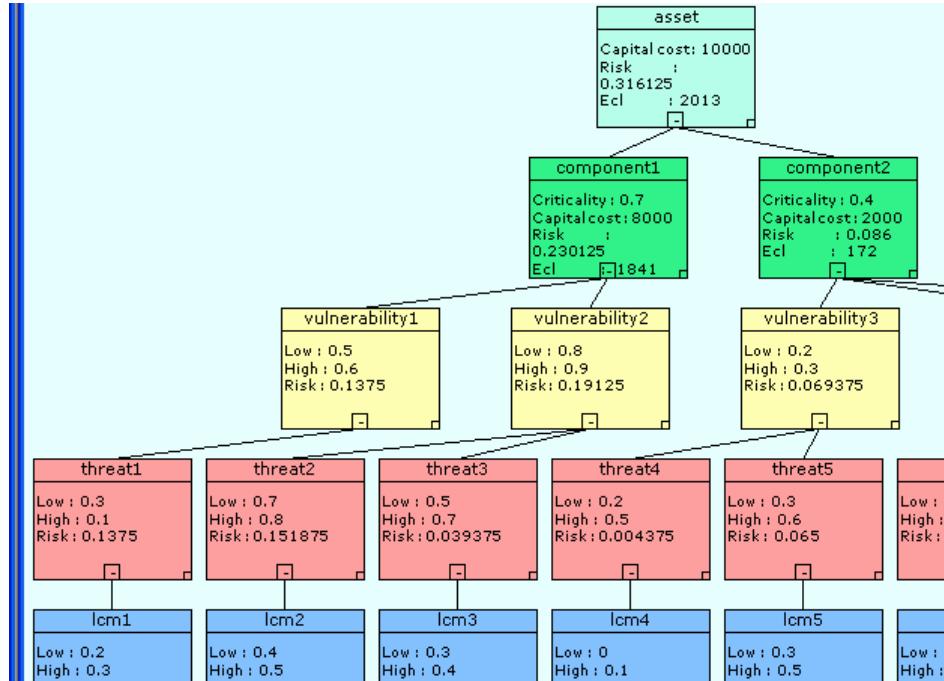


Fig. 4. Vi-secanto output tree diagram for one asset (screen shot from abstract example)

Because this information can be saved and retrieved, a user can easily add a new component, vulnerability, threat or LCM (lack of countermeasure). Each of these objects has properties, when the properties change, all graphics will be redrawn and numeric parameters will be recalculated.

The main aim of Vi-secanto is two-fold. First, it provides developers, who are not security specialists, with usable security knowledge and guidance for using that knowledge to improve the security of Web applications. Second, it identifies relative risk to help managers allocate limited resources to addressing security needs. Templates of existing security knowledge are provided as a starting point for new projects.

6 Summary & Contributions

This research will contribute to the improvement of Web application security by providing a computer-based tool that will make it easier for Web application developers to build more secure software and for managers to assess the risk and potential loss of not reducing these vulnerabilities. Currently, the tool is in the form of

an early prototype. The prototype will be tested and evaluated by a Web application development company with expertise in security problems specific to Web applications. The feedback from this evaluation will be used to improve the design of the tool.

References

1. Alexander, I.: Misuse Cases: Use Cases with Hostile Intent. *IEEE Software*, Vol. 20(1) (2003) 58-66
2. Amoroso, E.G.: *Fundamentals of Computer Security Technology*. Prentice-Hall PTR. (1994)
3. Boehm, B., Basili, V.R.: Top 10 list [Software Development]. *IEEE Computer*, Vol. 34(1) (2001) 135-137
4. Deloitte Touche Tohmatsu: 2006 Global Security Survey (2006)
5. Hevner, A. R., March, S. T., Park, J., Ram, S.: Design Science in Information Systems Research. *MIS Quarterly*, Vol. 28(1) (2004) 75-105
6. Hevner, A.R., March, S.T.: The Information Systems Research Cycle. *IEEE Computer*, Vol. 36(11) (2003) 111-113
7. Microsoft Corporation: Microsoft Threat Analysis and Modeling Tool. Microsoft Download Center, Retrieved 20 September, 2006, See <http://www.microsoft.com/downloads/details.aspx?FamilyID=59888078-9daf-4e96-b7d1-944703479451&DisplayLang=en> (2006)
8. PTA Technologies: Practical Threat Analysis. Tel-Aviv, Israel, Retrieved 9 February, 2006, See <http://www.ptatechnologies.com> (2006)
9. Sahinoglu, M.: Security Meter: A Practical Decision-Tree Model to Quantify Risk. *IEEE Security & Privacy Magazine*, Vol. 3(3) (2005) 18-24
10. Steffan, J., Schumacher, M.: Collaborative Attack Modeling. In the 2002 ACM Symposium on Applied Computing. Madrid, Spain: ACM Press (2002)
11. Vaishnavi, V., Kuechler, B.: Design Research in Information Systems. IS World Net Retrieved 14 March, 2006, See <http://www.isworld.org/Researchdesign/drisISworld.htm> (2005)

Model Checking Nonblocking Data Structures

David Friggens

Centre for Logic, Language and Computation
Victoria University of Wellington
david.friggents@vuw.ac.nz

Abstract. Nonblocking concurrent data structure implementations are complex and hard to reason about. We are investigating how model checking tools can be used to find errors in, and to verify properties of, these algorithms.

1 Introduction

Concurrent data structure implementations have traditionally been designed using locks to implement mutual exclusion among the threads at critical sections. Nonblocking algorithms generally provide much better efficiency and scalability than those based on mutual exclusion. However they are also harder to design and reason about; many papers have been published containing algorithms with bugs [DFG⁺00, SHC00], some with pseudo-mathematical ‘proofs’. One approach for avoiding incorrect algorithms is to formally generate them in a way that ensures their correctness. A number of general methods have been proposed for constructing nonblocking implementations from sequential and lock-based ones, but the performance of the resulting algorithms is very poor compared with the corresponding lock-based algorithms [LaM94]. Also, some efforts have been made towards refinement-based methods that produce correct implementations by construction [AC05, GC06]. An alternative approach is to use formal methods tools to discover the existence or verify the absence of bugs in a given algorithm.

Colleagues at Victoria University and Sun Microsystems are investigating the use of deductive proof methods using theorem proving tools; they have verified the linearisability correctness condition for a number of nonblocking data structures in PVS¹ [DGLM04, CDG05, CG05, CGLM06]. Verifications by theorem provers are, however, difficult and time-consuming. Furthermore, if a proof attempt has stalled it is not always easy to tell whether it is due to a bug in the algorithm or simply inexperience or mistakes on the part of the user [Doh03].

Model checking [CGP99] is an automated formal verification technique that explores the entire state space of a finite-state system, checking a specification property (traditionally in a temporal logic). Thus it is able to determine the satisfiability of the property in the system, returning “yes” if it is true, and “no” otherwise, along with a counterexample — an execution trace leading from an initial state to an error state. As the state space of a concurrent data structure

¹ Prototype Verification System, <http://pvs.csli.sri.com>

is infinite, model checking can only be applied to a finite instantiation — with a bounded number of threads etc — or to a finite-state abstraction.

The aim of my research is to investigate how model checking can be applied to verify nonblocking data structure implementations. The key problems are how to represent the systems and properties of interest, and how to construct finite-state models that can be used to prove properties of the general infinite-state systems. In this paper we describe this project, focussing on how to model check the linearisability correctness condition.

2 Nonblocking algorithms

One of the greatest concerns with concurrent software is how to synchronise access to shared information and prevent “data races”. Each thread can have any number of steps from other threads interleaved between two of its own steps, so it needs some way of ensuring consistency of at least some information between steps. For example, if two threads are both attempting to increment a shared variable x , they might both execute $\text{read}(x)$ and receive the value 0, then both execute $\text{write}(x, 1)$, effectively losing the result of one operation.

The most common method for dealing with this issue is to employ *mutual exclusion* through the use of *locks*. Locks grant access to a particular area of memory to only one thread at a time — a thread must acquire the lock before it enters the *critical section* that accesses the memory, and must release it afterwards. Locks do solve the synchronisation problem — each thread is guaranteed that no other will alter the data whilst it holds the lock — but they can introduce additional problems. The system can halt in a *deadlock* when all threads are waiting for a lock to be released, it can exhibit near-sequential behaviour when fast threads are *convoying* through a sequence of locks behind a slow thread, or it can experience *priority inversion* when a high priority thread is forced to wait for a lower priority one.

Avoiding explicit locks does not guarantee nonblocking behaviour, so it is important to formally state the behaviour that is desired of a system. An algorithm is *wait-free* if *every* thread is able to complete an operation within a finite number of its own steps. This property ensures that every thread will make progress, independent of the number and behaviour of other threads. Wait-freedom captures the ideal notion of nonblocking behaviour, but in practice wait-free algorithms are usually expensive to implement, and few algorithms have been proposed that are deemed to be of practical significance. Alternatively, an algorithm is *lock-free*² if *some* thread is able to complete an operation within a finite number of steps of the system. This property ensures that the system will always make progress, independently of the number and behaviour of individual threads. In contrast to wait-freedom, it sacrifices individual guarantees of progress for a system guarantee of progress. This is less than ideal, but is often good enough

² Some authors use the term “nonblocking” to refer specifically to this property, rather than the collection of such properties.

in practice — a thread is infinitely delayed only if other threads complete an infinite number of operations.

2.1 Correctness

The sequential approach to correctness is to observe the states of a data structure at the invocation and response of an operation to determine whether it has been applied ‘correctly’. This is meaningless for concurrent data structures though, as multiple operations can occur at the same time. The intuitive notion of correctness that we adopt in this context is that an operation must appear to take effect atomically at some point between its invocation and response by an outside observer with no knowledge of the internal state, such that the resulting ordering of operations forms a correct sequential execution. This notion is captured by the *linearisability* correctness condition (sometimes referred to as *atomicity*), which was introduced by Herlihy and Wing [HW90].

3 Model Checking Linearisability

3.1 Example

We will use the algorithm presented in Figure 1 as a running example. This is a simplification of Treiber’s lock-free stack implementation [Tre86]. The modified algorithm never frees memory back to the system, meaning that it is inefficient for practical purposes but simpler for explanatory and exploratory purposes. The stack employs a linked list, and operations take “snapshots” of the `Head` pointer (`ss`) and its `next` pointer (`ssn`) before attempting to update the stack using Compare-and-Swap (CAS). CAS is an atomic operation that updates a location only if it contains an expected value — see Figure 2.

The push operation allocates a new node, and enters its value. It then takes a snapshot of `Head`, points its `next` field at the snapshot and then, if `Head` has not changed, makes `Head` point to the node. If `Head` has been modified since the snapshot was taken, CAS returns false and the operation retries. The pop operation is similar. It takes a snapshot of `Head` and checks to see if it is null; if so the stack is empty. It then reads the values of the snapshot node and attempts to point `Head` to the next pointer.

3.2 Modelling the Implementation

In any real deployment of this algorithm the data structure will be part of a much larger system, with threads accessing many other data structures as well. However the other parts of the system will not affect the stack data structure, so without loss of generality we can use a model that contains only one stack. Our model has global variables containing the shared data structure and a collection of threads, each with its own local variables. Each thread is nonterminating and nondeterministically performs an infinite sequence of operations. The threads

```

type Node = {val: T; next: Node};
shared Head: Node := null;

push(v:T)
1. n := new(Node);
2. n.val := v;
3. repeat
4.   ss := Head;
5.   n.next := ss;
6. until CAS(Head,ss,n)

pop(): T
1. repeat
2.   ss := Head;
3.   if ss = null then
4.     return empty;
5.   ssn := ss.next;
6.   lv := ss.val;
7. until CAS(Head,ss,ssn);
8. return lv

```

Fig. 1. Lock-free Stack

```

CAS(*loc: T, old: T, new: T): Boolean
atomic
  if *loc = old then
    *loc := new;
    return true
  else
    return false

```

Fig. 2. Pseudocode for Compare-and-Swap

have an idle state, in which they are not performing any operation, and from which they can choose to perform either a push or a pop. When they reach the end of the operation they return to the idle state and choose another operation. The first step of an operation is called the *invocation* and the last the *response*.

In the initial state of the system the stack is empty (`Head` is `null`) and all threads are idle. Each operation consists of a number of atomic steps, and in any execution of the system the atomic steps of different threads may be non-deterministically *interleaved*, i.e. two sequential atomic steps of one thread may be separated by any number of atomic steps from other threads. Thus the set of all reachable states includes the initial state and any state that results from performing a single atomic step of any thread in a reachable state.

3.3 Linearisability

The stack implementation is linearisable if each operation appears to take effect atomically at some point between its invocation and response. Thus, we need to identify the “linearisation points” at which this occurs. For a push operation this is at line 6 when the CAS returns true, which happens exactly once in each push. It is at this point that the value v can be considered to be “pushed” into the stack, as it is now pointed to by `Head`, and is accessible to other threads. Similarly, for a pop operation that returns a value, the linearisation point is at line 7, when the CAS returns true. It is at this point that the `Head` pointer is

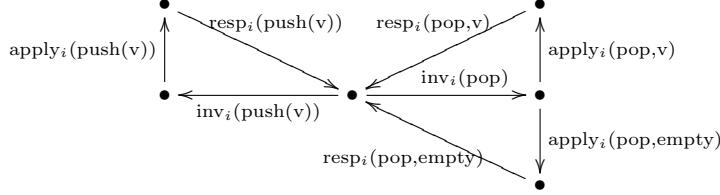


Fig. 3. Thread i of a concurrent stack specification

incremented to the second node in the list (or to null for a one element list) so the first element can be considered ‘‘popped’’. The linearisation point for a pop operation that returns empty is less straightforward. It is at line 2, when the snapshot is taken and `Head` is null, as this is the point where the operation ‘‘observes’’ an empty stack. It is not the subsequent test at line 3 because the stack is definitely empty at line 2 (when `Head` is null) but by line 3, when the thread inspects the snapshot, another thread may have performed a push operation so the stack may not be empty anymore.

Having identified potential linearisation points, we need to verify that they are actually places where the operations can appear to take effect atomically. To do so, we construct another system that is a concurrent specification of the data structure. It contains the same global variables as the implementation, but its threads are different. The threads perform the entire operation in a single atomic step, as a sequential specification does, but it performs the invocation and response as separate steps. Figure 3 displays a state transition diagram for a single thread of the concurrent stack specification. The idle state, where no operation is being performed, is the central dot. In this state, the thread nondeterministically chooses to attempt either a push or a pop operation. If it chooses to push a value v , it first performs the invocation (which has no effect), then applies the full operation (allocating a new node with v , pointing `Head` to it and pointing its next field to the old value of `Head`), and finally it performs the response (also no effect), which returns it to the idle state. These three steps are atomic, and can be interleaved with the atomic steps of other threads. The behaviour is similar if the thread chooses to perform a pop operation, except that after the invocation it may perform either an $\text{apply}_i(\text{pop},v)$ (recording the value v of `Head.val` and reassigning `Head` to its next value) or an $\text{apply}_i(\text{pop},\text{empty})$ (which has no effect) depending on the value of its stack representation.

Now, the linearisability of the implementation is equivalent to *trace inclusion* by the concurrent specification, i.e. every sequence of invocations, linearisation points and responses generated by an execution of the implementation can also be generated by the specification. One way to prove trace inclusion of two systems is to show that there is a *simulation* between them; this is a stronger property than implies trace inclusion. The deductive proofs in PVS used this approach [CDG05], and Smith and Derrick have shown how to model check for

ward simulation, using large formulas in computation tree logic [SD05]. Instead, for these systems, we can check trace inclusion directly and more efficiently by providing the trace of the implementation as input for the specification to see if it is able to generate the same set of traces. We synchronously combine the implementation and the specification into a new system where one atomic step comprises a single atomic step from each component. Additionally, the specification does not make a nondeterministic choice of the available atomic steps, but performs the step that matches the step performed by the implementation. For example, if thread 7 in the implementation performs the step that is the invocation of a $\text{push}(v)$ operation, then thread 7 in the specification performs the step $\text{inv}_7(\text{push}(v))$. If the implementation performs a step that is neither an invocation, linearisation point nor response then the specification performs a “stuttering” step that has no effect. Alternatively, the specification may be unable to perform the step that matches the implementation’s step, e.g. the implementation has performed the step that is the linearisation point for a pop returning value a , but the first element in the specification’s stack is b , allowing $\text{apply}_i(\text{pop}, b)$ but not $\text{apply}_i(\text{pop}, a)$. In this situation, the system enters an error state.

If the specification can perform the appropriate step to match every step performed by the implementation throughout the whole system then we have shown that any execution of the implementation is equivalent to one where each operation is applied atomically at some point between its invocation and response, i.e. it is linearisable.

This property can be easily expressed in a model checker as the invariant that the error state is never reached. I have used this method to examine both finite instantiations of this algorithm and a number of others, using the model checkers Spin³ [Hol04] and SAL⁴. Predictably, it found no bugs in the algorithms that have been proved correct, but it also quickly and easily discovered bugs in the algorithms that were known to be incorrect.

4 Limitations

The method described in the previous section has two general limitations on its applicability. The first is due to linearisation points not always being as nicely behaved as with the stack example. It may be possible that one step of the implementation corresponds to more than one step of the specification, and/or that the linearisation point of an operation may be a step of another operation (by a different thread). This situation can be accommodated by asynchronously, rather than synchronously, composing the implementation and specification.

Additionally, in some algorithms it can be possible that a step may or may not be the linearisation point of an operation, depending on the future behaviour of other threads. In other words, it is not possible to determine at that point in the execution whether the specification should perform a matching operation

³ <http://spinroot.com>

⁴ Symbolic Analysis Laboratory, <http://sal.cs1.sri.com>

step or not. One possible solution is to replace the specification system with an intermediate system that is similar but allows multiple “linearisation point” steps for some operations. To complete this approach we are also required to show that the intermediate system is linearisable, but by traversing the state space in reverse, which is more difficult.

The second limitation is that model checkers can only verify properties of finite-state systems, but these algorithms are in general infinite-state. They fail to be finite because of three factors: the number of threads, the maximum size of the data structure and the cardinality of the data elements. In any real world implementation these will be finite, but there is no theoretical limit on how large they can be, so for verification purposes we consider them to be unbounded.

There are two ways of producing finite-state systems for a model checker. The first is to simply create a finite instantiation by placing bounds on these numbers. This is easy to achieve and is useful for bughunting — any bugs found in the instantiation are also bugs of the general algorithm. However, an absence of bugs in part of the statespace does not guarantee correctness in general. Alternatively, a property-preserving abstraction technique could be employed to generate a finite-state system that represents the infinite-state implementation. This would allow verification of linearisability for the algorithm in general, as any property that is true in the abstract system is also true in the concrete one. Many techniques have been proposed for abstracting infinite-state systems in ways that preserve certain properties, but they mostly deal with only one aspect of unboundedness. The three factors mentioned above make abstraction much more complicated, and to date no satisfactory techniques have been applied to these algorithms. As such, this is currently the main focus of my research.

References

- [AC05] Jean-Raymond Abrial and Dominique Cansell. Formal construction of a non-blocking concurrent queue algorithm. *Journal of Universal Computer Science*, 11(5):744–770, 2005.
- [CDG05] Robert Colvin, Simon Doherty, and Lindsay Groves. Verifying concurrent data structures by simulation. In J. Derrick and E. A. Boiten, editors, *Proceedings of the International Refinement Workshop (REFINE)*, volume 137.2 of *Electronic Notes in Theoretical Computer Science*, pages 93–110. Elsevier, 2005.
- [CG05] Robert Colvin and Lindsay Groves. Formal verification of an array-based nonblocking queue. In C. Ghezzi, Y. Fu, S. Liu, and J. Woodcock, editors, *Proceedings of the 10th International Conference on Engineering of Complex Computer Systems (ICECCS)*, pages 507–516. IEEE Computer Society Press, 2005.
- [CGLM06] Robert Colvin, Lindsay Groves, Victor Luchangco, and Mark Moir. Formal verification of a lazy concurrent list-based set algorithm. In T. Ball and R. B. Jones, editors, *Proceedings of the 18th International Conference on Computer Aided Verification (CAV)*, volume 4144 of *Lecture Notes in Computer Science*, pages 475–488. Springer-Verlag, 2006.

- [CGP99] Edmund M. Clarke, Orna Grumberg, and Doron A. Peled. *Model Checking*. MIT Press, 1999.
- [DFG⁺00] David L. Detlefs, Christine H. Flood, Alexander T. Garthwaite, Paul A. Martin, Nir N. Shavit, and Guy L. Steele, Jr. Even better DCAS-based concurrent deques. In M. P. Herlihy, editor, *Proceedings of the 14th International Conference on Distributed Computing (DISC)*, volume 1914 of *Lecture Notes in Computer Science*, pages 59–73. Springer-Verlag, 2000.
- [DGLM04] Simon Doherty, Lindsay Groves, Victor Luchangco, and Mark Moir. Formal verification of a practical lock-free queue algorithm. In D. de Frutos-Escrig and M. Núñez, editors, *Proceedings of the 24th IFIP WG 6.1 International Conference on Formal Techniques for Networked and Distributed Systems (FORTE)*, volume 3235 of *Lecture Notes in Computer Science*, pages 97–114. Springer-Verlag, 2004.
- [Doh03] Simon Doherty. Modelling and verifying non-blocking algorithms that use dynamically allocated memory. M.Sc. thesis, Victoria University of Wellington, 2003.
- [GC06] Lindsay Groves and Robert Colvin. Derivation of a scalable lock-free stack algorithm. In *Proceedings of the International Refinement Workshop (REFINE)*, to appear in *Electronic Notes in Theoretical Computer Science*. Elsevier, 2006.
- [HW90] Maurice P. Herlihy and Jeannette M. Wing. Linearizability: A correctness condition for concurrent objects. *ACM Transactions on Programming Languages and Systems*, 12(3):463–492, July 1990.
- [Hol04] Gerard J. Holzmann. *The Spin Model Checker: Primer and reference manual*. Addison-Wesley, 2004.
- [LaM94] Anthony LaMarca. A performance evaluation of lock-free synchronization protocols. In J. H. Anderson, D. Peleg, and E. Borowsky, editors, *Proceedings of the 13th Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 130–140. ACM Press, 1994.
- [SD05] Graeme Smith and John Derrick. Model checking downwards simulations. In J. Derrick and E. A. Boiten, editors, *Proceedings of the International Refinement Workshop (REFINE)*, volume 137.2 of *Electronic Notes in Theoretical Computer Science*, pages 205–225. Elsevier, 2005.
- [SHC00] Chien-Hua Shann, Ting-Lu Huang, and Cheng Chen. A practical non-blocking queue algorithm using Compare-and-Swap. In M. Miyazaki and M. Takizawa, editors, *Proceedings of the 7th International Conference on Parallel and Distributed Systems (ICPADS)*, pages 470–475. IEEE Computer Society Press, 2000.
- [Tre86] R. Kent Treiber. Systems programming: Coping with parallelism. Technical Report RJ 5118, IBM Almaden Research Centre, April 1986.

Formal Methods and Refinement for User-Centred Design Artefacts

Judy Bowen and Steve Reeves

Department of Computer Science
University of Waikato
Private Bag 3105
Hamilton
New Zealand
jab34@cs.waikato.ac.nz

Abstract. When we design and implement software systems there are many different approaches we can take. We may choose a formal approach, where we formally specify the system and prove properties about its intended behaviour before refining to an implementation. Conversely, we may take a totally informal approach where we plan our system by jotting ideas down on paper, discussing ideas with users, drawing sketches *etc.* Formal methods are naturally suited to modelling underlying system behaviour while user-centred approaches to user interface design fit comfortably with more informal approaches. In order to develop systems which benefit from both of these approaches and recognise their respective uses for different parts of the design process we need to find ways of integrating user-centred design methods with formal methods. My research addresses this problem, and in particular looks at ways of including informal designs within a formal process and examining these in relation to standard notions of refinement.

1 Introduction

Designing and building software systems, particularly large and complex systems, often requires us to work in a modular fashion. Different parts of the system will be worked on at different times, perhaps by different groups of software engineers, designers and programmers.

Separation of the design and implementation of the user interface (UI) of a system from what we will refer to as the underlying system behaviour is a common and pragmatic approach for many applications. This separation allows us to not only focus on the different concerns which different parts of the system development present, but more importantly, allows for different development methods and design techniques.

When we develop the underlying system functionality for an application we are often concerned with issues such as correctness, reliability, robustness and efficiency *etc.* which lend themselves to the techniques we call “formal”. Such formal techniques include specifying requirements, validating and verifying specifications and refinement methods. When we develop UIs, however, our concerns

are often more human-focussed (this is particularly true if we follow a user-centred design (UCD) approach). The design techniques we adopt reflect this and rely on more informal strategies such as prototyping, scenarios, storyboards, iteration based on user-feedback, usability testing *etc.*

Whilst we can see the benefits of this separation of concerns and design methods in terms of being able to adopt the most suitable development approach to different parts of the task, there are clearly some problems associated with it. If our aim is to use a formal process to develop provably correct software (which it is), then we must ensure that all parts of the system have been designed in a way which satisfies this. That is, when we bring together the different parts of the system in our final implementation we want to be sure that it is still correct and behaves as expected.

This gap between the formal and informal has been identified and discussed many times, notably in 1990 by Thimbleby [12]. Several different approaches have been taken over recent years by different groups of researchers to try and bridge this gap. A small sample of such work includes [7],[13] and[8]. Much of this work is demonstrably a step forward in bringing together formal methods and UI design, however, for the most part, the methods and techniques which have been developed have failed to become mainstream. Partly this is due to the reluctance of any group to change their working practices and adopt newly derived methods. In addition, persuading users of formal methods to adopt less formal methods has proved as unsuccessful as encouraging UI designers to abandon their human-centred approach in favour of more formal approaches.

The approach we are taking with our research is to consider the existing, diverse, methods being used to develop software and find ways of formally linking them together. In particular, because our interests lie in both using formal methods and rigorous development techniques to develop our software, and UCD approaches to UI design, our intention is to find ways of *interpreting* the sorts of informal design artefacts produced in a UCD process within a formal framework.

2 User-Centred Design Artefacts

The purpose of user-centred design (UCD) is to ensure that the software we build, and in particular the interface to that software, meets the expectations of the actual users of the software. UCD design techniques therefore aim to involve the end-users at an early stage of the development to find out about not only the tasks they need to perform with the software but also information about the users themselves.

Techniques used by UCD practitioners may include ethnographic studies, task analysis, scenarios and personas to enhance the task analysis process, prototyping and usability testing. All of these methods involve collaboration with users and are mostly iterative in nature.

The sorts of artefacts that are generated during such processes reflect the collaborative nature of the UCD approach and will include things like whiteboard design sessions with post-it notes used to represent interface elements,

textual narrative descriptions of things like domain information and scenarios, task analysis models and paper-based prototypes.

One of the problems we face when trying to capture UCD processes within a formal software engineering context is that the artefacts produced are intentionally informal. They aim to encourage users to feel able to participate and change the design, and lo-fidelity artefacts, such as paper prototypes for example, have been shown to be very successful for this purpose.

3 Presentation Model

We have developed a model, which we call the presentation model, as a way of formally capturing the meaning of informal design artefacts. It is a deliberately simple model because the informal artefacts it describes are themselves simple. When we talk about the *meaning* of a design artefact we are talking about what the UI described by the informal artefact is supposed to do, *i.e.* if it were transformed into an implementation what its behaviour would be.

A design artefact, such as a paper prototype, has two parts. It consists of a sketch which suggests how the final UI may look, but it also has what we call a *narrative*. That is, when a designer shows the prototype to the user there is a story which accompanies it explaining how the user can interact with it and what the effect is of that interaction. This allows a simulated interaction to take place which enables the user and designer to evaluate the suitability of the proposed design.

The presentation model is a formal model which describes an informal design artefact in terms of the widgets of the design and formally describes the narrative of the design. We state that it is formal because the semantics we have given for the presentation model [3] are a conservative extension of set theory allowing us to derive a sound logic for presentation models. The presentation model is not intended to replace the informal design artefact, rather it acts as a bridge between the meaning intended by the design and the formal design process being used for the system functionality.

The presentation model consists of declarations followed by a description of each element in the design (which we refer to as the widgets) by way of a triple consisting of an identifier, a category (using the category hierarchy from the work described in [2]) and a set of behaviours. The complete UI of a system may consist of a number of different windows and/or dialogues. Each of these is described by a list of its widgets as a component presentation model which allows for modular construction of the overall UI by joining together these components.

As an example, consider the design given in Fig. 1 which is a prototype for a mobile-phone based application to control a home heating system (this example is derived from one originally given by Calvery *et al.* in [6]).

Each of the four different screens in the design is described in its own component model and then these are combined to describe the overall interface, as follows:



Fig. 1. Design for Mobile Phone Application UI

<i>PModel</i>	<i>MPHeat MPMenu MPBed MPLounge MPBath</i>
<i>Widgetname</i>	<i>BathSelect LoungeSelect BedSelect QuitOpt IncBathOpt DecBathOpt IncLoungeOpt DecLoungeOpt IncBedOpt DecBedOpt AcceptOpt CancelOpt BathTempDisp BathRangeDisp LoungeTempDisp LoungeRangeDisp BedTempDisp BedRangeDisp ActCtrl SValSel SValRespdr</i>
<i>Category</i>	<i>ShowBath ShowLounge ShowBed QuitApp IncBathTemp DecBathTemp IncLoungeTemp DecLoungeTemp IncBedTemp DecBedTemp StoreSettings ShowMenuPage DispBathTemp DispBathRange DispBedTemp DispBedRange DispLoungeTemp DispLoungeRange</i>
<i>Behaviour</i>	<i>MPHeat is MPMenu : MPBath : MPLounge : MPBed</i>
<i>MPHeat is</i>	
<i>MPMenu is</i>	<i>(BathSelect, ActCtrl, (ShowBath)) (LoungeSelect, ActCtrl, (ShowLounge)) (BedSelect, ActCtrl, (ShowBed)) (QuitOpt, ActCtrl, (QuitApp))</i>
<i>MPBed is</i>	<i>... the rest is omitted for brevity.</i>

Our first use for the presentation model is to enable us to include the design of the UI in our formal refinement process. We have previously given a detailed account of this process [5] and it is not our intention to repeat these details here. However we will give an outline of the process and direct the interested reader to [5].

At its simplest level we can describe refinement by the principle of substitutivity. If we have an abstract system, *A* and we replace it with a more concrete system, *C*, and *C* provides all of the behaviours of *A* such that any user of *A*

cannot tell that the substitution has occurred, then C is a refinement of A . As part of our refinement process for systems and UIs we need to ensure that all user operations described in the specification have been described in the UI design. From the presentation model of the design we can produce a Z description (using the framework for describing widget categories in Z given in [2]), this then allows us to use standard data refinement and simulation techniques, such as those given in [14], to show that a refinement holds between the UI design and the specification.

We have also derived a notion of equivalence between designs, based again on the presentation model. The intention here is to be able to take different UI designs (for the same system) and using the presentation models of these designs determine if they can be considered in some way equivalent. The functionality of a design is given by the set of behaviours of the presentation model of that design. If we wish to compare two different UI designs to determine whether or not they have the same functionality, we can simply compare the corresponding behaviour sets of their presentation models. We have also considered other types of equivalence which exist between designs, namely *Component Equivalence* and *Isomorphism*. Formal definitions of these types of equivalence and their uses can be found in [5].

A third use of presentation models is in ensuring consistency. Consistency is an important principle of UI design and is included by Shneiderman [11] as one of his eight golden rules. An application may consist of many different screens, menus and dialogues and maintaining consistency throughout the design is not a trivial task. Using the presentation model we can ensure that controls which have the same behaviour have the same name. Conversely we can also check that controls with the same name have the same behaviour.

4 Presentation Interaction Models

In the previous section we introduced the presentation model and gave some examples of its uses. There are, however, other properties we wish to consider when looking at proposed designs which also have an effect on issues such as refinement.

A UI is not static; it generally consists of multiple windows, dialogues *etc.* Part of the behaviour of the UI revolves around the changes in this state. That is, the way in which a user can change the currently visible window and allow movement to different parts of the UI. In our research we refer to this type of behaviour as the UI functionality, this enables us to separate it from the behaviour which relates to underlying system functionality.

When we consider refinement between UI designs and systems we consider the relationship between functionality provided by the UI and the operations described in the system specification. It is not enough, however, to be able to prove that such a correspondence exists, we also need to show that the functionality is reachable within the UI. For this reason we must also consider issues such as reachability and deadlock within UIs.

The presentation model does not contain enough information to examine these types of properties as it provides a static view of the design. The presentation model describes the total functionality of the proposed UI as given by the designs but not the dynamic behaviour required to consider reachability properties. One way of solving this would be to extend the presentation model. However, rather than doing this and making our deliberately simple model more complex, we have decided instead to use the existing presentation model in conjunction with another formalism, finite state machines (FSM).

FSM have been used in UI modelling for both design and evaluation since the late 1960's [9]. A common problem with this approach, however, is that of "state explosion". Given the complexity of UIs, particularly modern GUIs, the number of possible states makes such an approach impractical in many cases. In our research, however, we are already dealing with an abstraction of the UI, *i.e.* the presentation model, and are concerned with the dynamic changes between different parts of the UI rather than with each micro-change of the UI itself. Using these two formalisms together, therefore, avoids the state explosion problem. We produce a FSM which is at a high level of abstraction and describe a relation between states and presentation models to give its lower-level meaning.

Our FSM consists of: a finite set of states, Q ; a finite set of input labels, Σ ; a transition function, δ , which takes a state and an input label and returns a state ($q \rightarrow a \rightarrow q'$); a start state, q_0 , one of the states in Q ; a relation R which relates each state in Q to a presentation model. The FSM is then a five tuple $(Q, \Sigma, \delta, q_0, R)$.

When the FSM is in a particular state the presentation model associated with that state is active, *i.e.* we can consider the application to be in a state where the part of the UI described in that model is visible to the user and available for interaction. The input labels in Σ are themselves the names of behaviours taken from the behaviour sets of the presentation models. So, we can only move between states if the functionality to do so exists in our design. We call the combination of presentation model and FSM in this way a presentation and interaction model (PIM).

If we consider again the example shown in Fig. 1 we can derive a PIM for this which we give in Fig. 2. In order to show that a particular behaviour is reachable, we first need to show that the part of the UI it is in (*i.e.* the component presentation model containing this behaviour) is itself reachable in the FSM. We can do this using standard FSM methods which check for reachability and deadlock.

5 Current Status and Future Work

The work described in sections 3 and 4 of this paper are a first examination of refinement concerns for UIs and systems which are developed using separate processes. This has led to our current research focus which is to examine further implications of this style of integration and to put them in the context of standard, traditional refinement techniques.

MPHeat

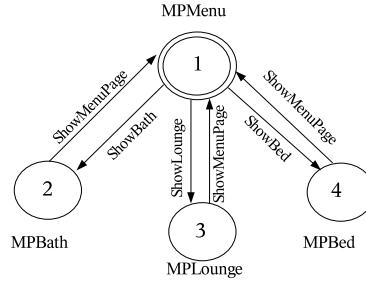


Fig. 2. PIM for MPHeat Presentation Model

In particular we are interested in the nature of the composition of the system and UI at the pre-implementation level. We need to define this composition in order to formally define how the system and UI will be integrated when we produce our final implementation. There are several interesting concerns that arise from this.

Although we now have a way of formally describing UI design artefacts, via the presentation model, the formalisms we use to describe the UI and the underlying system specification are different. We have shown in [4] how we can give a description of a presentation model in the specification language Z [1]. This does give us a common formal language to talk about the system and the UI (assuming we are also using Z as our specification language), but in order to use this to consider the nature of the composition of system and UI there is more that needs to be done. We need to extend and combine our Z descriptions to fully describe all parts of the UI and functionality and explicitly model the composition within the specification. While this is one possible approach, it is not necessarily the most useful, or the most intuitive.

Another consideration which arises from consideration of the composition of the system and UI is that of monotonicity of the refinement. That is we want to be sure that if we refine just the system, or just the UI, the composition is similarly refined.

Given these considerations, we are currently looking at modelling the system and UI as reactive systems using the μ -Charts language. A refinement theory for μ -Charts has been given in [10] which includes a monotonic refinement for μ -charts which are in parallel composition with each other. By describing a system and UI pair as a composed μ -chart we can examine the possibilities of using this theory as the basis for a monotonic refinement theory for systems and UIs.

6 Conclusion

The purpose of our research is to find ways of integrating existing user-centred design techniques with a formal software-development process. So far we have

identified some of the problems associated with this and developed models to enable us to overcome these problems. We have started to examine issues of refinement for UIs and systems and are currently extending this work through the consideration of system and UI composition.

References

1. ISO/IEC 13568. *Information Technology—Z Formal Specification Notation—Syntax, Type System and Semantics*. Prentice-Hall International series in computer science. ISO/IEC, first edition, 2002.
2. Judy Bowen. Formal specification of user interface design guidelines. Masters thesis, Computer Science Department, University of Waikato, 2005.
3. Judy Bowen and Steve Reeve. Formal models for informal GUI designs. In *Proceedings of Formal Methods for Interactive Systems*. Electronic Notes in Theoretical Computer Science, Elsevier, 2006.
4. Judy Bowen and Steve Reeves. Including design guidelines in the formal specification of interfaces in Z. In Helen Treharne, Steve King, Martin Henson, and Steve Schneider, editors, *ZB2005: Formal Specification and Development in Z and B*, volume 3455 of *LNCS*, pages 454–471. Springer-Verlag, 2005.
5. Judy Bowen and Steve Reeves. Formal refinement of informal GUI design artefacts. In *Proceedings of the Australian Software Engineering Conference (ASWEC'06)*, pages 221–230. IEEE, 2006.
6. Gaelle Calvary, Joëlle Coutaz, and David Thevenin. A unifying reference framework for the development of plastic user interfaces. In *EHCI '01: Proceedings of the 8th IFIP International Conference on Engineering for Human-Computer Interaction*, pages 173–192, London, UK, 2001. Springer-Verlag.
7. Antony Courtney. Functionally modeled user interfaces. In *Interactive Systems. Design, Specification, and Verification. 10th International Workshop DSV-IS 2003, Funchal, Madeira Island (Portugal) J. Joaquim, N. Jardim Nunes, J. Falcao e Cunha (ed.)*, pages 107–123. Springer Verlag Lecture Notes in Computer Science LNCS, 2003.
8. A. Hussey, I. MacColl, and D. Carrington. Assessing usability from formal user-interface designs. Technical Report TR00-15, Software Verification Research Centre, The University of Queensland, 2000.
9. David L. Parnas. On the use of transition diagrams in the design of a user interface for an interactive computer system. In *Proceedings of the 1969 24th national conference*, pages 379–385. ACM Press, 1969.
10. G. Reeve. The syntax and semantics of μ -charts. Technical Report 04/2004, Department of Computer Science, University of Waikato, 2004.
11. Ben Shneiderman. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Addison Wesley Longman Inc, 3rd edition, 1998.
12. H. Thimbleby. Design of interactive systems. *The Software Engineer's Reference Book*, 1990.
13. Harold Thimbleby. User interface design with matrix algebra. *ACM Trans. Comput.-Hum. Interact.*, 11(2):181–236, 2004.
14. J. Woodcock and J. Davies. *Using Z: Specification, Refinement and Proof*. Prentice Hall, 1996.

Performance of Evolving IEEE 802.11 Security Architectures

Andrew Gin, Nilufar Baghaei, Ray Hunt

Department of Computer Science and Software Engineering

University of Canterbury, New Zealand

{atg19, ray}@cosc.canterbury.ac.nz, {nilufar.baghaei}@gmail.com

Abstract. Introduction of security in Wireless LANs had always created degradation in performance - depending in part upon the options chosen for encryption, authentication, re-keying and digital certificates. The finalisation of IEEE 802.11i (WPA2) along with its implementation in new hardware platforms has significantly altered such performance characteristics. This paper describes the performance analysis of IEEE 802.11i and compares the results with studies based on WEP and WPA. This research shows that while there are statistically significant differences between the security options, they are small enough to be ignored in practice. This paper provides an analysis which will assist in secure wireless network design which meets specific performance criteria.

Keywords: Network Research, Wireless Network Security

1 Introduction

Wireless networks have increased in use dramatically over recent years. Mobility and the freedom to connect without fixed network connections have contributed to its popularity. However, wireless networks are inherently more vulnerable to attack than wired networks. Where a wired network needs to have its lines tapped, or direct physical access to the hub or switch to have its security compromised, a wireless network broadcasts over the open airwaves. Anyone within transmission range can receive this signal.

Because of this inherent vulnerability, a comprehensive security standard, IEEE 802.11i was developed [1, 2, 3]. Wireless network security encompasses three main security architectures: WEP, WPA and WPA2. These are outlined in section 2. As wireless security evolves, so too has the hardware platforms used to implement it.

This paper investigates the effects of the WPA2 security specification on the performance (throughput, latency and errors) of wireless networks and compares this performance with the existing WEP and WPA architectures. It builds on previous work carried out in [4] and [5] by including the full WPA and WPA2 specifications.

2 Wireless LAN Security Architectures

2.1 The Evolution of WLAN Security

Wired Equivalent Privacy (WEP) was intended to make a wireless connection as secure as an unsecured wired network [1]. WEP attempts to bring confidentiality, access control and data integrity to a wireless connection. Unfortunately, WEP is flawed in each of these security aspects, and these are well published [2].

The problems in WEP led to the creation of the 802.11i standard [3]. The 802.11i standard seeks to address all of the security issues concerning wireless LANs and is split into three main parts: Temporal Key Integrity Protocol (TKIP) and Counter Mode with CBC-MAC Protocol (CCMP) both offer confidentiality and data integrity, and IEEE 802.1x provides authentication. TKIP is designed for legacy devices and hardware that can only support WEP, while CCMP is a more advanced, robust protocol designed for all new devices. Either of these can be combined with 802.1x authentication; when 802.1x is combined with TKIP, it is known as WPA, and when 802.1x is combined with CCMP, it is known as WPA2. Wi-Fi Protected Access (WPA) was introduced in October 2003 as an interim solution, to immediately address the security flaws in WEP while the 802.11i standard was still under development. In order to effectively secure wireless LANs, WPA needed to work on existing hardware and infrastructure, and uses TKIP to achieve this.

802.11i has two modes of operation in terms of authentication: IEEE 802.1x and Pre-Shared Key (PSK). IEEE 802.1x authentication (also known as WPA Enterprise mode) is aimed at corporations with existing authentication infrastructure in place, such as RADIUS. Under the PSK method (also known as WPA Personal mode), a key is manually entered into each device on the wireless network. WPA under PSK authentication is designed for small office/home office (SOHO) wireless networks that do not have authentication servers, but still require the security benefits that WPA has over WEP.

Wi-Fi Protected Access 2 (WPA2) was designed from scratch and represents the complete implementation of the 802.11i standard. WPA2 uses the Counter Mode with CBC-MAC Protocol (CCMP) instead of TKIP and is required for devices claiming Robust Security Network (RSN) compliance. The Advanced Encryption Standard (AES) algorithm is used (instead of WPA's RC4) for both encryption (in Counter Mode (CTR)) and data integrity (CBC-MAC). WPA2 can use the IEEE 802.1x authentication framework (for WPA2 Enterprise mode), or PSK (for WPA2 Personal mode) in the same way as WPA to perform authentication. These wireless network security methods are summarised in Table 1.

Table 1. Summary of wireless network security methods (adapted from [9]).

	WEP	802.11i Methods	
Security Protocol	WEP	WPA	WPA2
Cipher	RC4	RC4	AES
Key Length	40 or 104 bits	128 bits encryption, 64 bits authentication	128 bits
Key Life	24 bit IV	48 bit IV	
Key Generation	Concatenation	Two phase mixing function	Not needed
Data Integrity	CRC-32	Michael	CBC-MAC
Header Integrity	None	Michael	CBC-MAC
Replay Protection	None	Packet Number	
Key Management	None	EAP-based	
Authentication	Open or Shared Key	802.1x or Pre-Shared Key (PSK)	

2.2 Related Work

Wong [4] investigated the effects of Virtual Private Network (VPN) and IEEE 802.1x security frameworks on network performance. The general pattern found was that the stronger the security level, the lower the network performance. While VPN offered end to end security (compared to 802.1x, which provided client to access point security only), it was more complex to implement and had a greater impact on performance than 802.1x did.

Baghaei [5] extended this research by evaluating the effects of packet length on the network throughput with different security architectures. This study showed that WEP encryption significantly reduced network performance when the network was congested.

No previous research has analysed the effects of the IEEE 802.11i (WPA and WPA2) security specifications on network performance. This study builds on Wong's and Baghaei's research by focusing on the effects of this specification.

3 Experimental Design

3.1 Security Levels

The following seven security levels were used to test the performance of the wireless network testbed. Each level represents a newer, more complex degree of security than the one before. The evolution of these security methods were described in section 2.1. The security layers defined are:

1. No Security
2. WEP shared key authentication and 40 bit encryption

- 3. WEP shared key authentication and 104 bit encryption**
- 4. WPA with PSK authentication and RC4 encryption**
- 5. WPA with EAP-TLS authentication and RC4 encryption**
- 6. WPA2 with PSK authentication and AES encryption**
- 7. WPA2 with EAP-TLS authentication and AES encryption**

The first layer (No Security) was tested as a base case in order to ascertain whether the different security layers reduce throughput by adding overhead.

3.2 Network Measurement Methodology

This project's main aim is to ascertain the extent to which the security architectures reduce throughput in a wireless LAN. Two clients and a server were configured as illustrated in Figure 2. A traffic generator (IPTraffic¹) was used to generate traffic on Client2 to send to Client1. Section 3.4 describes this setup in more detail. Each test was performed at each of the security layers.

The first set of experiments were synthetic benchmarks. IPTraffic was configured as described in Section 3.3. This test recorded the throughput of Client2 sending to Client1.

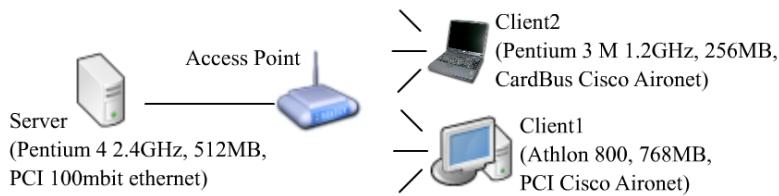


Figure 2: The topology used to perform the experiments.

The second set of experiments were similar to the first set. These experiments consisted of Client2 downloading an 11.2MB video file from Client1 via FTP. Client1 was running Cerberus FTP server (<http://www.cerberusftp.com>). While synthetic benchmarks are useful as they test only a specific aspect (in this case network throughput) and enable full control over the characteristics of the traffic, they are not representative of 'real world' applications. File transfer is a popular network application, and this test gave insight into one aspect of 'real world' throughput and whether the security layers affected it.

3.3 Traffic Generator Configuration

IPTraffic was the traffic generator used in this study and was configured as follows:

- Total Number of Packets sent to Client1 per synthetic test: 20,000.
- Traffic Protocol: TCP and UDP.
- TCP Window: 8192 bytes (default for Windows XP).

¹ <http://www.zti-telecom.com/pages/iptraffic-test-measure.htm>

- **Packet Payloads:** Synthetic throughput tests; 6 and 1460 bytes for TCP, 1472 bytes for UDP. A 6 byte payload for TCP was the minimum that IP Traffic could support, and represents header only TCP acknowledgement packets. A payload of 1460 and 1472 bytes correspond to the maximum payload of TCP and UDP respectively. Synthetic latency and error tests: 1460 bytes for both TCP and UDP. Since throughput was not being measured, the payload was kept constant to limit any effects it may have on latency and error rates.

While the tool used was able to test with duplex connections (where both clients send to each other), for the scope of this project, each connection was configured to send in one direction only. Results were analysed using ANOVA at the 95% confidence interval.

3.4 Experimental Network Testbed

The server was running Microsoft Windows Server 2003 Enterprise Edition with Service Pack 1. This server operated as a RADIUS server via Microsoft's Internet Authentication Service. Both clients were running Windows XP Professional with Service Pack 2.

A Cisco Aironet 1130AG series access point, connected to the server via a 100Mbps Ethernet connection, operated in the 802.11a 5GHz 54Mbps mode. The 5GHz frequency range is less crowded than the 2.4GHz range (for example some cordless phones and Bluetooth devices use this range) and is therefore less susceptible to interference. Noise generated by other devices also tend to affect 2.4GHz frequencies more than 5GHz frequencies. The Cisco 1130AG has hardware acceleration for both AES and TKIP.

As all of the wireless network devices used in the experiment were Cisco based, vendor interoperability problems were avoided. The wireless network adapters in the two clients were based on the Atheros AR5212 and AR5112 chipsets which, like the access point, has hardware accelerated encryption². These configurations are summarised in Figure 2. None of the computers were running any processes which could affect processor or network utilisation.

As the access point and the server were the only devices directly connected to each other, the network was isolated from other LANs. A scanning utility verified there were no other wireless networks close enough to be a direct source of interference. All of the hardware was in a single room to ensure that walls and other obstacles would not be a factor in the measurements. All hardware locations were kept the same for each experiment, and no hardware was moved during the running of the experiments. Antennas were positioned no closer than 1.5m to each other since [6] states some wireless products have low data rates at very close ranges, as they restrict signals that are too strong.

² <http://www.atheros.com/pt/AR5002XBulletin.htm>

4 Performance of Security Architectures

While the focus here is on the performance of the 802.11i security specification, comparisons are also made with previous studies ([4] and [5]). While several experiments were performed, only the results from the 1460 byte TCP synthetic transfer and FTP transfer will be included, as these can be directly comparable to results found in [4] and [5].

In the first set of synthetic benchmarks, Client2 was configured to send data as fast as possible to Client1. Figure 3a shows the mean throughput for each of the security levels defined in section 3.1. The differences are statistically significant ($F_{6,24} = 11.09, p < 0.01$). While statistically significant, in practical terms, the difference is not large; across all security levels, the mean throughput is 8.12Mbps³ (s.d. 0.26). A standard deviation of 0.26Mbps corresponds to approximately 33.28KB/s. This variance is not large when considering the mean is 8.12Mbps, which corresponds to 1015KB/s. This variance is also typical of a wired Ethernet LAN [7].

These results are reproduced in the second experiment, the FTP transfer (Figure 4a), where Client2 downloaded an 11.2 MB file from Client1. There are no statistically significant differences between the security levels ($F_{6,24} = 2.00, p = 0.11$). The mean throughput across all security levels is 9.73Mbps (s.d. 0.29).

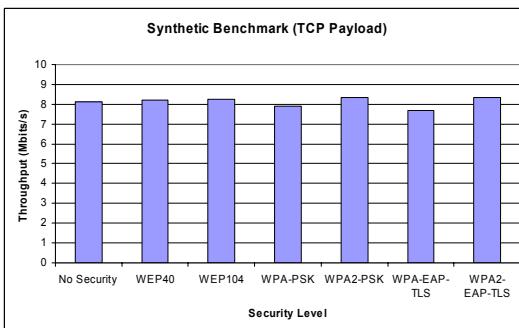


Figure 3a (results from this study)

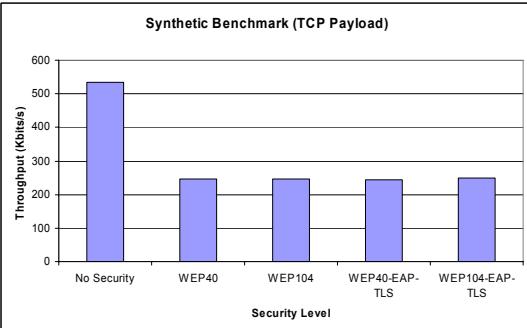


Figure 3b (results from [5])

Figure 3: Throughput with a synthetic TCP payload.

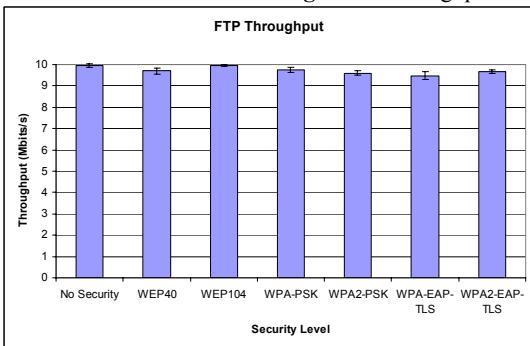


Figure 4a (results from this study)

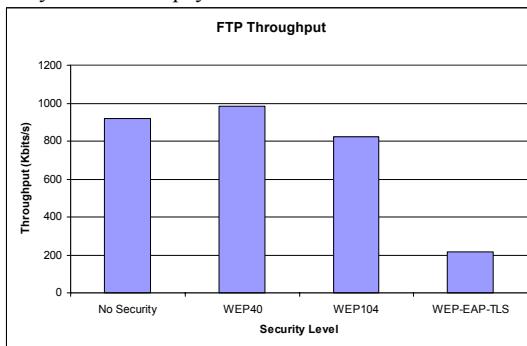


Figure 4b (results from [4])

Figure 4: Throughput with the FTP file transfer.

³ Mbps = megabits per second, KB/s = kilobytes per second.

4.3 Discussion

These experiments show that wireless network throughput is largely unaffected when the various security levels are applied. This is in stark contrast to findings in [4] and [5], where throughput fell as more complex security was introduced. The differences between the security levels are statistically different in the synthetic TCP transfer (Figure 3a). These differences are not considered realistically significant, and the fact that there are no statistically significant differences found between the security levels in the FTP transfer (Figure 4a) is a further indication that any statistical differences between the security levels is not of practical significance.

As [5] identified, the implementation of the security specification has a large impact on the throughput performance. In [5], the wireless LAN client adapters performed the encryption and decryption in the firmware (this was dictated by the current WEP implementations of the time) while in this study, these functions were implemented in the hardware. This explains the minimal performance degradation when security is used in this study.

In terms of authentication type, both [4] and [5] found that the 802.1x authentication methods resulted in a large performance degradation. This is clearly not the case here; as stated previously, the security levels have little real effect on the throughput. Overall, the EAP-TLS security methods have similar throughputs to the other security levels, and in the FTP instance, cannot even be statistically differentiated from the other levels. However, these experiments were performed with dynamic rekeying disabled and this must be taken into consideration. It is expected that as the re-authentication and dynamic rekeying frequency increases during a session, the performance will decrease, due to the additional overhead.

These results show that it is now possible to implement a wireless network with robust security measures in place, without any real compromise in performance.

4.4 Limitations

One limitation is that the clients used to send and receive were not of uniform hardware specification; ensuring the clients were of uniform hardware specification would guarantee that the throughput was independent of client hardware specifications.

These results are also only representative of an ideal base case infrastructure mode set up. Radio and physical interference will have an effect on the performance. The type of client adapter and access point will also have an effect on the throughput, as detailed in [8].

5 Conclusions

This research aimed to discover the effects of the 802.11i security specification on wireless network performance, and compare it to existing security configurations.

The results show that while there are statistically significant differences between the security levels in some situations, it is unlikely that these small differences will

matter in practice. The differences are so small that under some conditions, the various security levels cannot even be statistically differentiated. All of the security levels, ranging from No Security to WPA2-EAP-TLS had realistically similar throughput, latency and error rates under all transfers.

The results also indicate that the WLAN client adapters and access point have an effect on the performance. Both the client adapters and access point used in this study had hardware accelerated encryption and decryption, resulting in minimal network performance degradation when more complex security was employed.

Future work could investigate the effects on throughput with more than two clients, as well as enabling re-authentication and dynamic rekeying. Other 802.1x authentication methods could also be used in combination with this, in order to discover how the re-authentication frequency and number of clients affects network performance. Performing these tests with different hardware as well as software or network applications will also provide further insights into the effects of the security levels on wireless network performance. The finalisation of IEEE 802.11e (Quality of Service) in November 2005 is a valuable step. The combination of a range of security options in conjunction with QoS traffic engineering is a further area of research.

This research shows that it is possible to establish a wireless network with robust security, without any noticeable compromise in performance. Using modern hardware with hardware accelerated security features, there is no reason to use anything less than the complete WPA2 security specification with AES/CCMP encryption.

Acknowledgements. This research was funded in part by Cisco for the hardware and ZTI Telecom for the traffic generator.

References

- [1] "Securing Wi-Fi wireless networks with today's technologies." Wi-Fi Alliance White Paper, http://www.wi-fi.org/files/uploaded_files/wp_4_Securing%20Wireless%20Networks_2-6-03.pdf, February 2003.
- [2] B. Brown, "802.11: the security differences between b and i," *IEEE Potentials*, vol. 22, pp. 23-27, October – November 2003.
- [3] "Amendment 6: Medium Access Control (MAC) Security Enhancements." ANSI/IEEE Std 802.11i, Institute of Electrical and Electronic Engineers, 2004.
- [4] R. Hunt, J. Vargo, J. Wong, "Impact of security architectures on wireless network performance," in *5th IEEE International Conference on Mobile and Wireless Communications Networks (MWCN 2003)*, October 2003.
- [5] N. Baghaei and R. Hunt, "Security Performance of loaded IEEE 802.11b wireless networks," *Computer Communications*, Elsevier, U.K., vol. 27, no. 17, pp. 1746-1756, 2004.
- [6] "Methodology for Testing Wireless LAN Performance." Atheros Communications White Paper, www.atheros.com/pt/atheros_benchmark_whitepaper.pdf, 2003.
- [7] D. R. Boggs, J. C. Mogul and C. A. Kent, "Measured capacity of an ethernet: myths and reality," *SIGCOMM Comput. Commun. Rev.*, vol. 25, no.1, pp. 123-136, 1995.
- [8] A. Vasan and A. U. Shankar, "An empirical characterization of instantaneous throughput in 802.11b WLANS," Department of Computer Science, University of Maryland, September 2002.
- [9] N. Cam-Winget, T. Moore, D. Stanley and J. Walker, "IEEE 802.11i Overview," in *NIST 802.11 Wireless LAN Security Workshop*, http://csrc.nist.gov/wireless/10_802.11i%20Overview-jw1.pdf, December 2002.

Interaction Designers on eXtreme Programming Teams: Two Case Studies from the *Real World*

Jennifer Ferreira¹, James Noble^{1,2} and Robert Biddle³

¹ Victoria University of Wellington, New Zealand

² Microsoft Research Cambridge

{jennifer,kjx}@mcs.vuw.ac.nz

³ Carleton University, Ottawa, Canada

robert_biddle@carleton.ca

Abstract. The interaction designer role is not an acknowledged role on the core eXtreme Programming (XP) team and XP has no explicit process for dealing with interaction design. We interviewed interaction designers and other team members on two real-world XP teams and in this paper we report on how they combined interaction design activities with XP. Initial results show that having interaction designers on the team, resulted in a workflow that differed from the pure XP process in a significant way: up-front design for user interaction, as opposed to up-front code design, was considered necessary. The iterative nature of XP development required that the interaction designers have continual involvement with the development of the product, which inevitably influenced the nature of the relationship between the interaction designers and the developers.

Keywords: Software Engineering, System Usability, eXtreme Programming, Interaction Design

1 Introduction

Extreme Programming (XP) [1] is the most widely adopted agile method today [2], however, it has no explicit process for dealing with interaction design. Studies have shown that companies that produce software with poor usability lose money [3–5]. Anderson connects usability directly with value: “Value is perceived through usage. Without usability there is no value.” [6]. Interaction designers can help the end user perceive that value by enhancing the usability of the product.

We conducted interviews with interaction designers⁴ and other team members on two real-world XP teams to learn about the role of the interaction designer and how interaction design activities are integrated with XP. Our hypothesis was that the role of the interaction designer would differ from that of interaction designers on traditional software development teams and would also be key in the success of the integration of

⁴ Although the interviewees used the terms *interaction designer* and *user interface designer* interchangeably, we use the term *interaction designer* to refer to the member of the development team, whose main responsibility it is to design the user experience and the user interface. The team members involved in mainly coding activities, are referred to as the *developers*.

interaction design and XP. Our preliminary results show that XP teams have different approaches to combining user interaction activities with their development process, but wholeheartedly agree on the valuable contribution of the interaction designers to the development of their product. They see a real need for usability specialists, in the form of interaction designers, to be involved with development because experience has shown that following the XP rules is not enough to ensure usable software. Our research also shows that with the interaction designer role added to the team, the workflow differed from the pure XP process in a significant way: up-front design for user interaction, as opposed to up-front code design, was considered necessary. Further, the iterative nature of XP development required the interaction designers to be continually involved with the development of the product, which influenced the nature of the relationship between the interaction designers and the developers.

In the next section we explain our research method. The teams we interviewed are introduced in section 3 and the interaction designer role is explored in section 4. After a brief discussion of the related work in section 5, we present our conclusions in section 6.

2 Research Method

The data for this paper was obtained from semi-structured in-depth one-on-one interviews with four team members from two different software companies: Greenback Inc. based in the United States and Emerald Inc. in Ireland.⁵ The two main objectives of these interviews were to understand the process and practices relating to interaction design on XP projects and to learn about the interaction designer role in XP teams. The interviewer conducted the interviews with team members from Greenback Inc. on their premises. The interviews with the team members from Emerald Inc. were conducted using phone conferencing facilities. The interviews were voice recorded and transcribed in detail. All persons interviewed were asked to validate the transcriptions, as well as the interpreted findings. We present those findings here and quote the interviewees as illustration.

The interview transcriptions were analysed using the method known as open coding [7]. This method is the first step in grounded theory analysis [8] and is used to identify the different recurring categories present in the data. The next step will be to perform axial coding, where the relationships between the categories are established. In the course of further analysis, more interview data will be incorporated to eventually produce a theory grounded in the data.

3 Team Profiles

Based in the United States, Greenback Inc. is a company that develops and markets web-based software to support IT managers and development professionals. Team Liberty of Greenback Inc. is an XP team and includes ten engineers and one user interface designer/product manager. At the time of the interviews, Team Liberty was working on redesigning and enhancing one of its products. Their User Interface Designer/Product Manager described the project as follows:

⁵ All names have been changed in order to preserve confidentiality.

“There are several features we added and several things that we wanted to do with the product. And one thing we noticed was that performance was really bad, it was built upon a really terrible code base. It was just all hacked together. User interface was terrible and, you know, the user interactions were very cumbersome. So we decided from that generation of product that we were gonna rewrite and start from scratch. We took everything that we wanted out of the old product and built it the way we really wanted the product to end up. [...] During this process we adopted the XP methodology.”
— *User Interface Designer/Product Manager, Team Liberty*

The second team, Team Cláirseach, is employed by Emerald Inc., based in Ireland. This company develops and sells software to support wealth management. Team Cláirseach is also an XP team and includes four engineers, one domain expert/on-site customer and two interaction designers. Their Project Manager and one of the interaction designers described their project and its status at the time of the interviews:

“We’ve been focusing on, sort of, single-user client systems.” — *Project Manager, Team Cláirseach*

“There are smaller projects as well and we have our bigger, our over-riding kind of application building, which is our wealth planner application, which we’ve been working on for two years and we have our first customer for that now and we’re releasing that in a few weeks. And we’re a certain of a way through developing. It’s quite big, so it’s an ongoing project [...]” — *Interaction Designer, Team Cláirseach*

Team Liberty provided the Engineering Manager and User Interface Designer/Product Manager for interviews and Team Cláirseach provided the Project Manager and Interaction Designer. The Engineering Manager and Project Manager were both from a programming background, whereas the User Interface Designer/Product Manager and Interaction Designer were both experienced and qualified user interaction specialists.

4 eXtreme Interaction Designers

During the open coding of the interviews, we became aware that the interaction designer role presents unique challenges in the world of XP. In this section we take a look at how the various up-front activities formed part of the development process and the nature of the interaction designer/developer relationship.

4.1 Up-Front Activities

In XP the development process is an evolutionary process, with little or no significant design up front. Beck states that “it is better to do a simple thing today...than to do a more complicated thing today that may never be used anyway,” [1] that is, “Concentrate on what is scheduled for today only.” [2] Among practitioners, this is known as the YAGNI (You Ain’t Gonna Need It) principle and warns against adding features to the product that are not required in the current development iteration. Therefore, to avoid waste and to remain responsive to changing requirements, XP discourages up-front code design. While neither Team Liberty nor Team Cláirseach performed up-front

code design, both teams found a significant amount of **user interaction** design, after the requirements gathering process and before implementation begins, to be crucial for usability. Both teams created *personas*⁶ to represent the types of users of their systems. The following quotes illustrate the value of the personas:

“[..] as we created the user stories all written from these personas, what that allowed me to do was figure out who was trying to do what [...] and it just helps in the design process.” — *User Interface Designer/Product Manager, Team Liberty*

“[..] that [personas] was for the UI, but also for the user stories, one of the most important things to do. [...]” — *Engineering Manager, Team Liberty*

“[..] We design personas up front for our projects and we identify what their goals are in using the product. We have to make sure that when we’re testing the product we meet all their goals.” — *Interaction Designer, Team Clairseach*

Team Liberty then created a navigation model and a style guide in order to set the interaction standards for the future user interface and to provide some consistency in its behaviour and appearance:

“I’d try to work things out. Just create some upfront consistency, like, what do buttons look like, where are they placed, what do tables look like, how do users interact with tables, what do forms look like, how do you get from a table to a form and then back to the table – basic interaction models. So, kind of like a style guide. [...] having that upfront consistency in designing the behaviour into it, regardless of what the application is, I think, is one way of mitigating risk. And then having a rough, high-level navigation model, I think, is probably a good idea too.” — *User Interface Designer/Product Manager, Team Liberty*

“[..] it’s hard to have a holistic view of the application when you haven’t followed out all its framework: How am I gonna handle breadcrumbs, how am I gonna handle main navigation, how am I gonna handle user profile stuff. [...] So I created a really basic framework in the navigation model [...]” — *User Interface Designer/Product Manager, Team Liberty*

Team Liberty refined and implemented the detailed user interface designs incrementally during the XP development iterations. By contrast, Team Clairseach completed the design of the bulk of the user interface before implementation began:

“Before it gets into development, the user interface is more or less 90% defined. So there wouldn’t be that many changes once it goes into the development iterations [...] because we put so much effort into the [up-front] design.” — *Interaction Designer, Team Clairseach*

4.2 Continuous Involvement

In popular software engineering processes, such as the Rational Unified Process (RUP) [10], the interaction designer is responsible for the complete design of the user interface early on in the development process. Then the design is handed to the developers

⁶ A well-known technique introduced by Alan Cooper [9]

who implement the design, with little or no further involvement from the interaction designers. In the XP teams interviewed, this was not the case. The interaction designers on these XP teams were involved in the up-front activities discussed in section 4.1, but from that point on, as the user interface was implemented during the development iterations, the interaction designer was constantly communicating with the developers regarding changes, testing and clearing up any questions the developers may have had about the design. For the designers it was possible to receive immediate feedback from the developers when something could not be implemented, and the developers could be informed by the designers straight away when there were additions or changes to the user interface.

“[..] the designers test it [the user interface] on a day to day basis, give feedback back to the development team to ensure if anything was missing, that we’d write a card for it and it will be captured.” — *Interaction Designer, Team Cláirseach*

“[..] sometimes even during development people realize, ‘Oh this and this doesn’t work,’ and then they go to the User Interface Designer/Product Manager.” — *Engineering Manager, Team Liberty*

With this continuous involvement the interaction designer could ensure that the developers maintain consistency in the user interface, as specified either by the style-guide or the mock-up.

The Engineering Manager on Team Cláirseach related an incident where the communication between the designers and the developers had broken down, leading to major bottlenecks in the development process:

“[In] London a guy had serious problems [...] he had an interaction expert guy who [would] not talk to the programmers and start changing stuff [...] There was very little communication between the different blocks and it caused serious delays and bottlenecks [...] and not knowing what he’s doing he’s causing them more work.” — *Engineering Manager, Team Cláirseach*

4.3 Designer-Developer Interaction

The relationship the interaction designers have with the developers on the XP team is a direct consequence of their continuous involvement, as discussed in the previous section.

We observed two different approaches to the interaction designer/developer relationship on the XP teams and that difference is clearly marked by the following two quotes. Within Team Liberty the developers were free to give their User Interface Designer/Product Manager feedback on user interface design issues:

“[..] the Product Manager comes to the meeting and says, ‘Ok, here’s how we do this and this feature,’ and there’s ten engineers sitting there and saying, ‘Look, nobody works like this. What are you doing?’ [...] but it’s a fantastic thing; you have these sometimes very opinionated discussions up front and sometimes even during development.” — *Engineering Manager, Team Liberty*

Whereas, in Team Cláirseach, developers did not give feedback on user interface design issues:

“Pretty much design-wise, whatever the interaction designers say goes.” — *Engineering Manager, Team Cláirseach*

Team Liberty valued lively discussions and debate and enjoyed sharing knowledge of their domains and the overall product. They believed this helped them come up with the best solution. Interaction design in this team was clearly seen as a collaborative activity. Unfortunately, the Engineering Manager admitted that it was sometimes difficult to convince the developers to implement a screen that they did not like or did not agree with. This was largely due to the fact that the developers did not take into account how other types of users could use their software. By contrast, their user interface designer is trained to take all the different types of users into account:

“[...] there’s ten engineers sitting there and saying ‘Look, nobody works like this. What are you doing? It’s like, nobody’s doing this,’ and so sometimes it’s really, really hard for him [*User Interface Designer/Product Manager*] to tell them, ‘Look, you might not work like this but I think outside, people who use our application, they will work like this, they will appreciate this feature.’” — *Engineering Manager, Team Liberty*

The designer/developer relationship in Team Cláirseach, is based around respect and trust. The designers show their respect for the developers by not suggesting coding improvements and the developers show their respect for the designers by not suggesting user interface improvements:

“I have to say in our team here, each of us have great respect for each other’s work. We, as designers, have great respect for what the developers do and the developers have great respect for us. Even with great communication going, we’d never assume to make a suggestion about something we don’t know very much about.” — *Interaction Designer, Team Cláirseach*

Consequently, the interaction designers on Team Cláirseach have complete control of the user interface. Further, designers and developers trust each other that each will come up with the best solution for their domain and when issues do arise, both sides make trade-offs until both are happy. This team viewed their approach as a very efficient way of developing the product, as designers and developers do not waste time and effort on things they are not good at. Their view was that their work in their own domain is better when they do not have to think outside the scope of their domains:

“If I don’t have to worry about UI concerns, I can get more work done and get it done better. The UI doesn’t naturally fall inside my domain, it isn’t something I naturally do well.” — *Project Manager, Team Cláirseach*

Although not explicitly mentioned by Team Cláirseach, one disadvantage for this team may be that knowledge is not shared among team members, which would make it difficult for team members to learn from each other.

4.4 Valued Role

Both teams agreed that having interaction designers on the XP team was vital for enabling good interaction design. In their opinion, having the developers doing interaction design is not ideal:

“[...] you need someone who really has a very good idea about the really high level UI design.” — *Engineering Manager, Team Liberty*

“I don’t think the user interface designers should be engineers.” — *User Interface Designer/Product Manager, Team Liberty*

“I can’t imagine why a developer would be designing screens because their training lies in a whole different area to me, and their skill set lies in a different area [...]” — *Interaction Designer, Team Cláirseach*

5 Related Work

The way in which interaction design and agile development should work together has been discussed surprisingly little. One important early exception was the debate between Kent Beck and Alan Cooper [11]. This debate explicitly addressed the issue of when interaction design should occur relative to software development. Jeff Patton describes in several papers and tutorials how interaction design and agile development can work together by using a process where interaction design iterations fit into the iterative structure of agile development [12, 13]. Hodgetts presented an experience report about integrating User Experience Design into agile processes [14] and McInerney and Maurer investigated the role of User-Centered Designers on agile teams [15]. The study that is most closely related to the research presented in this paper is a case study by Rogers, Sharp and Preece [16]. Although their findings are specific to designing a web-based interface, they focus on the work of the graphic designer and how she relates to the XP team — a similar approach to that taken in our research. There are, however, very few independent studies of the interaction designer’s role in XP teams: a gap this research hopes to begin to fill.

6 Conclusion

We set out to investigate the processes and practices of interaction design of two real-world XP teams and uncovered some unique characteristics of the interaction designer role. The approach both teams interviewed have taken, differed in significant ways from the pure XP process, in that both teams performed significant up-front user interaction design — producing user interface mock-ups and style and navigation guides. The iterative nature of XP development, required continual involvement from the interaction designer, which differs from traditional interaction designer roles. Both teams acknowledged the value of having interaction designers on board and believed they helped create a better, more usable product.

References

1. Beck, K.: eXtreme Programming Explained: Embrace Change. Addison Wesley (2000)
2. Pettichord, B.: Deconstructing GUI Test Automation. STQE Magazine (January/February 2003) 28–32

3. Marcus, A.: Return on Investment for Usable User Interface Design: Examples and Statistics. Aaron Marcus and Associates, Inc. Whitepaper (2002) Published at URL: www.amanda.com/resources/ROI/AMA_ROIWhitePaper_28Feb02.pdf.
4. CC Pace: Usability and User Interface Design in XP. Available at URL: www.ccpace.com/Resources/documents/UsabilityinXP.pdf (Last accessed 15 December 2006.)
5. Nielsen Norman Group: Usability Return on Investment. Available at URL: <http://www.nngroup.com/reports/roi/> (Last Accessed 15 December 2006.)
6. Anderson, D.J.: Scheduling UI Design with Critical Chain Project Management. Published at URL: www.agilemanagement.net/Articles/Papers/Scheduling_UI_Design_v1_5.pdf (Revised October 2004. Last Accessed 15 December 2006)
7. Strauss, A., Corbin, C.: Basics of qualitative research: techniques and procedures for developing grounded theory. Thousand Oaks (California): Sage Publications (1998)
8. Glaser, B.G., Strauss, A.L.: The Discovery of Grounded Theory: Strategies for qualitative research. Aldine de Gruyter, Hawthorne, NY, USA (1967)
9. Cooper, A.: The Inmates are Running the Asylum: Why High-Tech Products Drive Us Crazy and How to Restore the Sanity. (Indianapolis): SAMS (1999)
10. International Business Machines Staff: Rational Unified Process: Best Practices for Software Development Teams. Rational Software Corporation Whitepaper (2003) Published at URL: <http://www-128.ibm.com/developerworks/rational/library/253.html>. Updated July 2005.
11. Fawcett, E.: Extreme programming vs. interaction design. FTP Online (2002)
12. Patton, J.: Improving on Agility: Adding Usage-Centered Design to a Typical Agile Software Development Environment. In: ForUse2003: Proceedings of the Second International Conference on Usage-Centered Design. (Portsmouth, NH, USA, 2003)
13. Patton, J.: Usage-Centered Design in Extreme Programming and Agile Software Development Environments. In: *Tutorial* at ForUse2003: Proceedings of the Second International Conference on Usage-Centered Design. (Portsmouth, NH, USA, 2003)
14. Hodgetts, P.: Experiences integrating sophisticated user experience design practices into agile processes. In: Proceedings of Agile 2005 Conference, Denver, CO, IEEE Computer Society (2005) 235–242
15. McInerney, P., Maurer, F.: UCD in agile projects: dream team or odd couple? *interactions* **12**(6) (2005) 19–23
16. Rogers, Y., Sharp, H., Preece, J.: Using XP to Develop Context-Sensitive Adverts for the Web. In: Companion website for the book *Interaction Design: Beyond Human-Computer Interaction*, at URL: http://www.id-book.com/casestudy_xp.htm, Wiley (Last Accessed 15 December 2006)

Congestion Control for Data-limited Flows

Syed Faisal Hasan

Department of Computer Science
University of Otago
Dunedin, New Zealand
`shasan@cs.otago.ac.nz`

Abstract. The best-effort service model of the Internet is unsuitable for multimedia applications which require a smooth and flexible packet transmission rate. TCP is unable to provide such a sending rate due to its strict adherence to congestion control. TCP-friendly-rate-control(TFRC) is a promising end-to-end rate control algorithm for multimedia traffic. TFRC based data-limited applications cannot grab the available bandwidth and are continually beaten at the competition by other TCP flows. We propose a new framework which alleviates this problem by packet prioritization. Depending on the encoded media streaming often requires a sharp rate change which is not permitted by TFRC. We address this problem by keeping an account of the unused share when the flow is sending at a rate below the fair share and use it at a later time when a quick rate increase is required.

Keywords: QoS, TCP congestion control, streaming media, TFRC

1 Introduction

Historically the Internet only provides best-effort delivery of packets. This model is suitable for elastic applications like large file transmission and email where the variation of sending rate have little or no effect on user perception. However this is inappropriate for multimedia applications like audio-video conferencing, streaming media etc. These applications require smooth and flexible packet transmission rate while tolerating few packet losses. For this reason some sort of resource reservation has been well sought after in order to keep the delay variation (jitter) in packet reception within some maximum threshold. A large number of research and development have been made under the framework of Quality-of-Service (QoS) for ensuring guaranteed throughput and delay [1],[2]. Unfortunately due to various reasons these frameworks have not yet seen any widespread deployment in the public Internet. As such, multimedia applications have learnt to adapt their sending rates to various network conditions. The role of the transport protocols like TCP and UDP are utmost important towards providing a satisfactory user experience for these rate adaptive applications.

TCP's congestion control mechanism has played a vital role for the success of todays Internet [3]. By limiting the sending rate of flows, congestion control ensures a fair and efficient sharing of resources for TCP based flows. This

proactive rate control contributes to widespread jitter when the application's required sending rate mismatches with TCP's allowable sending rate. Although techniques like client side buffering helps to a certain extend, this is a major obstacle for the future deployment of TCP based multimedia applications over the Internet. On the other hand, UDP which doesn't have any congestion control, is a viable alternative transport protocol which can be exploited to send packets at the application's desired rate. But in most of the times applications cannot use UDP due to Firewalls and NATs (network address translation). Moreover widespread deployment of UDP traffic is a threat to the fair and efficient sharing of resources in the Internet. The Internet Architecture Board(IAB) has already raised its concern about this issue[4]. As such during the last few years a number of research have been done to find a middle ground between the flexibility of UDP based data transmission and the confinement of TCP based ones.

Today most traffic in the Internet is TCP based. In order to be good network citizens other applications are required to be friendly with TCP flows. TCP friendly rate control protocol(TFRC) [5] has gained much attention and has been incorporated into the newly standardized Datagram Congestion Control Protocol(DCCP) [6]. However TFRC has yet to become an appropriate rate control algorithm for multimedia flows [7]. TFRC focuses on smooth sending rate at the cost of avoiding any sharp rate change which is often required by streaming applications. In this paper we try to illustrate some of the drawback of TFRC and describe a framework which is suitable for streaming applications. Streaming applications are data limited in the sense that they often cannot utilize the available bandwidth. The key idea is to keep track of this unused share and try to reuse it in a TCP friendly manner at times when needed. We use token bucket based packet marking to keep track of the unused share. One particular problem with TFRC is that when a data limited application is sending at a rate which is lower than the fair share, the flow continues to loose its fair share as other greedy flows try to grab the unused share. We use packet prioritization to tackle this problem.

This paper provides an overview of the area of research for my PhD and is organized as follows. In Section 2 a brief discussion on Quality-of-Service (QoS), streaming applications, and the notion of TCP friendliness is presented. In Section 3 the problem statement and proposed framework is described. Since this is an on going project in Section 4 the current status and future work of the project is described.

2 Background

2.1 Quality-of-Service

Quality-of-Service(QoS) in the Internet refers to the ability to deliver packets with a previously agreed delay and throughput variation. This implies having different levels of service for Internet traffic, to support various types of applications. Integrated Services(Intserv) [8] and Differentiated Services (DiffServ) [1] are two well established frameworks for introducing QoS on a best-effort network.

The Intserv architecture allocates resources to individual flows by a resource reservation protocol, whereas the DiffServ divides the traffic into a small number of classes and allocates resources on a per-class basis. In diffserv only boundary nodes (edge routers) at the edge of the network classify traffic and mark packets, the interior nodes (core routers) use the forwarding classes encoded in the packet header to determine the treatment of the packets. In contrast, Integrated services requires all nodes to perform packet classification to identify packets from reserved flows and schedule them with per-flow queueing. Diffserv defines a per-hop behaviour and an overall architecture leaving service definition on the hands of service providers. As such there is no well defined end-to-end services like frame-relay's committed information rate (CIR) with diffserv. Providing end-to-end CIR in diffserv requires a predefined service-level-agreement (SLA) between all the involved ISPs from source to destination, a daunting task indeed [9]. No wonder despite the simplicity and scalability promised by diffserv architecture, deployment of QoS mechanisms in the public Internet remains sparse [10]. Lack of QoS has made the multimedia applications more adaptive to changing resources. But when this adaptability is coined with a transport protocol having its own rate control mechanism, the task becomes more challenging.

2.2 Nature of Streaming

Bandwidth of a connection changes as new flows start and stop on the shared bottleneck link. Streaming servers usually keep multiple levels of encoded media suitable for different sending rates. Upon establishing the connection with the client, the server uses some packet-pair based bandwidth probing technique to calculate the available bandwidth and chooses the appropriate media encoding rate for streaming. The server switches between buffering and streaming mode. It starts with the buffering mode by sending at a higher rate to quickly fillout the client side media buffer upto a threshold. Afterwards the client starts depleting the media buffer for playout while the server switches to streaming mode and keeps sending at the previously chosen encoding rate. At regular intervals the server monitors the available bandwidth and packet loss rate and changes the media encoding rate accordingly. At times of congestion the client side media buffer may get emptied below the threshold level and eventually the client will stop playing. In order to refill the media buffer quickly, the server switches to buffering mode again. This event of rebuffering is undesirable and directly affects the user perception of streaming.

Unlike large file transmission applications streaming applications cannot grab the available bandwidth and send at a variable rate. Often these applications are happy sending at a fixed unfluctuating lower rate than a possible much higher but abrupt rate. We call these sort of applications as data-limited as they don't have enough data to saturate the available bandwidth.

The media codec may encode the media as a variable rate bit stream. Voice codec often use silence suppression, where the media goes completely idle for some time while one side listens to what other side has to say. In video codecs when there is a major motion or change of scene the codec encodes more data and

the stream bit rate may vary upto a factor of ten. These abrupt changes in the transmission rate is problematic for any congestion control algorithm which only promotes gradual increase in rate.

An ideal framework for streaming traffic should ensure the following two QoS:

- *Smooth Transmission Rate*: Regardless of the sending rate the application must be able to sustain that rate as long as it is consuming a fair share of resources.
- *Allow Quick Rate Change*: Variable rate bit stream requires the ability to change rates in short time. This must be allowed without overshooting the network's bottleneck bandwidth in an unfair manner.

2.3 Transport Protocol for Streaming

The predominant transport protocol for streaming media is Real-Time Transport Protocol(RTP) [11]. RTP is not a complete protocol like TCP and UDP and it usually runs on top of UDP, and is not required to respond to network congestion like TCP. Applications using RTP have some freedom to adjust their response to variations in network transit delay, packet loss and congestion based on their needs. RTP exposes the underlying packet loss behaviour of the network to the applications, allowing them to choose how to response to loss. Moreover the RTP framework provides framing, sequencing, timestamping, payload and user identitiy, and reception quality feedback by means of the accompanied RTP Control Protocol (RTCP). But as the amount of applications using UDP is growing, concern has been raised that the primitive congestion control algorithms employed in these applications are not sufficient and may lead to a 'congestion collapse' in the Internet [12].

On the other hand streaming applications very often use TCP when UDP cannot be used due to firewalls and network address translation (NAT) employed at end user routers, as is the case in most broadband networks. A robust congestion control algorithm is used in TCP to adapt the transmission rate to available bandwidth and to ensure that bandwidth is shared approximately equally among flows. The additive-increase-multiplicative-decrease (AIMD) [13] based rate control is appropriate for large file transmission as the application doesn't care about the rate fluctuation while the data has been transmitted. The rapid changes in sending rate enforced by AIMD congestion control algorithm of TCP causes significant disruption to the media playout. Moreover the in-order delivery of packets by TCP stalls the delivery of other data packets to the application until the gap of the missing packet has been filled by retransmission at a slower rate.

There have been many proposals for new congestion control algorithm to provide smooth sending rate for multimedia applications. Among these, TCP-Friendly rate control(TFRC) [5] is the most mature rate control algorithm taken seriously by the research community and has been adopted as an alternative congestion control algorithm in the newly standardized Datagram Congestion Control Protocol(DCCP) [14]. TFRC uses the TCP equation [15] to limit the

upper bound of sending rate but unlike TCP, maintains smoothness by responding slowly on average loss rather than a single loss. TFRC requires four to eight round-trip-times(RTT) to halve its sending rate in response to persistent congestion whereas TCP requires only one RTT. This slow responsiveness is balanced by a considerable slower increase in sending rate than that of TCP, of about 0.14 packets per RTT [16]. This situation is at odds with the variable bit rate stream which occasionally needs a sharp increase in sending rate. Moreover when a TFRC flow is sending below it's share of the link it will be subject to loss more than it induces as TCP is more aggressive than TFRC. The implication is that data-limited flows which are less bandwidth hungry will be penalized, and this is unfortunate, since it provides a disincentive for saving bandwidth.

3 A New Framework: Streaming Congestion Control

3.1 Problem Statement

TFRC lacks Quality-of-Service (QoS). It only makes sure that TFRC flows remain friendly to TCP flows and provide smooth sending rate by reacting slowly on packet loss. On the other hand when flows are sending at a rate which is below the fair share, it is often the victim of 'use it or lose it' principle of Internet as greedy flows grab the unused bandwidth and push the TCP friendly flow to retransmit.

3.2 Solution: Streaming Congestion Control Framework

Fair share of bandwidth means an equal share of the bandwidth. Data limited applications cannot utilize the fair share when there is not enough data to send. We use differentiated services assured forwarding framework [17] to make sure that the flow gets its required share when sending at a rate which is below the fair share and can often reuse the unused share. Leaky bucket tagger is used to keep track of the unused bandwidth as tokens. Leaky bucket is a well known algorithm for rate shaping and it is configured with a token generation rate and bucket depth. Each packet passing through the tagger must collect a token for transmission. If the token generation rate is faster than the packet arrival rate at the tagger then tokens get accumulated up to the bucket depth level. We calculate the fair share of the flow in bytes/s using TFRC and generate the tagger's tokens at this rate. When the application is sending at a rate below the fair share, tokens are accumulated in the bucket and we mark the packets as *Green* which has high priority. These high priority packets have lower drop probability in the core routers and/or receiver's loss rate calculation procedure. When the flow is sending above the fair share, provided there are tokens in the bucket, the packets are marked as *Yellow* which has bit higher drop precedence than the *Green* ones. All other flows are marked as *Red* which has the highest drop precedence. The algorithm is presented below:

```

if(send_rate < fair_share )
    mark packets Green
else if(send_rate > fair_share && token >0 )
    mark packets Yellow
else
    mark packets Red

```

We use this framework for flows based on TFRC. All other flows are not considered.

One difficulty with token bucket tagger is in setting the depth of the bucket. If we set the bucket depth too large then the flow might overshoot the network by sending more packets while it has been already sending at a rate above its fair share. On the other hand if we set the bucket depth too small then we might not utilize the unused share properly. We decided to calculate the bucket depth, D in bytes whenever a packet is received using the following formula:

```

D = fair_share * (k*RTT),
where k = 1,2,3,4 or 5, and
      RTT = Round-Trip-Time estimation

```

We have done experiments with different values of k and decided to choose the right value between 1 to 5 based upon the measured RTT range.

Another component of this framework is the handling of the marked packets. The design permits using diffserv(DS) enabled core routers where packets are droped/delayed probabilistically based on their tags. This require deployment of DS-enabled routers at the core which might not be possible immediately over the global Internet. In that case the core router's dropping machanisms can be placed at the receiver's end.

4 Current Status and Future Work

The framework is being implemented using ns-2 network simulator. The TFRC sender module has been modified to interact with the token bucket tagger. The core router's packet dropping algorithm is being tuned to meet the desired objective of the framework. This allows differential treatment of packets based upon the marked tag. This algorithm drops more RED marked packets than the GREEN ones. Early experiments show that the approach is effective when the flow is data-limited like most streaming applications. The future work is to implement router algorithms to send explicit feedback to the end host so that the sender can change rate much more aggressively without causing congestion. A java based prototype streaming system is being designed to test the framework by streaming audio-video files.

References

1. S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated services". RFC 2475, December 1998.
2. L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala. "RSVP: A new resource reservation protocol". IEEE Network, pages 8-18, September 1993.
3. V. Jacobson, "Congestion avoidance and control", Symposium proceedings on Communications architectures and protocols, p.314-329, August 16-18, 1988, Stanford, California, United States.
4. S. Floyd, Ed., J. Kempf, Ed., "IAB Concerns Regarding Congestion Control for Voice Traffic in the Internet", RFC 3714, March 2004.
5. Mark Handley, Sally Floyd, Jitendra Padhye, and Joerg C. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification", Internet Engineering Task Force, RFC 3448, January 2003.
6. Eddie Kohler, Mark Handley and Sally Floyd, "Designing DCCP: congestion control without reliability", SIGCOMM '06: Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications, pp.27-38, Pisa, Italy, 2006.
7. Vlad Balan, Lars Eggert, Saverio Niccolini and Marcus Brunner "An Experimental Evaluation of Voice Quality over the Datagram Congestion Control Protocol." To appear: IEEE Infocom, Anchorage, AL, USA, May 6-12, 2007.
8. R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, "Resource reservation protocol (RSVP) - version 1 functional specification". RFC 2205, September 1997.
9. L. Burgstahler, K. Dolzer, C. Hauser, J. Jhnert, S. Junghans, C. Macin, W. Payer, "Beyond technology: the missing pieces for QoS success", Proceedings of the ACM SIGCOMM workshop on Revisiting IP QoS: What have we learned, why do we care?, August 25-27, 2003, Karlsruhe, Germany.
10. Gregory Bell, "Failure to thrive: QoS and the culture of operational networking", Proceedings of the ACM SIGCOMM workshop on Revisiting IP QoS: What have we learned, why do we care?, August 25-27, 2003, Karlsruhe, Germany.
11. H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications", RFC 3550, Internet Engineering Task Force, July 2003.
12. Sally Floyd ,Kevin Fall, "Promoting the use of end-to-end congestion control in the Internet", IEEE/ACM Transactions on Networking (TON), vol.7 no.4, pp.458-472, Aug. 1999.
13. D.-M. Chiu, R. Jain, "Analysis of the increase and decrease algorithms for congestion avoidance in computer networks", Computer Networks and ISDN Systems, vol.17 no.1, pp.1-14, June 10, 1989.
14. E. Kohler, M. Handley and S. Floyd, "Datagram Congestion Control Protocol (DCCP)", RFC 4336, Internet Engineering Task Force, Mar. 2006.
15. Jitendra Padhye, Victor Firoiu, Don Towsley, Jim Kurose, "Modeling TCP throughput: a simple model and its empirical validation", Proceedings of the ACM SIGCOMM '98 conference on Applications, technologies, architectures, and protocols for computer communication, pp.303-314, August 31-September 04, 1998, Vancouver, British Columbia, Canada.
16. Sally Floyd and Mark Handley and Jitendra Padhye and Jorg Widmer, "Equation-based congestion control for unicast applications", Proceedings of ACM SIGCOMM 2000, pp.43-56, Aug. 2000, Stockholm, Sweden.
17. J. Heinanen, et al., "Assured Forwarding PHB Group," IETF RFC 2597, Jan. 1999. <http://www.ietf.org/rfc/rfc2597.txt>.

Collaborative Profiles for Event Notification in Healthcare: Lessons Learned from Healthcare Staff Interviews

Doris Jung

University of Waikato, Hamilton, New Zealand
d.jung@cs.waikato.ac.nz

Abstract. In their treatment, patients with chronic conditions must remember to perform recurring tasks such as taking their medication or measuring their bodily parameters. They have to document the progression of their condition as well as supply this data to their healthcare providers. To target these problems, we suggest applying a mobile alerting system. To fully exploit the potential of such a system, it has to support collaboration of the parties involved in the treatment of the patient. Furthermore, it should be adaptable to the particular context of each individual patient. For this purpose, we have proposed the concept of collaborative profiles. This paper describes interviews we have undertaken to explore the stance of healthcare staff towards our idea and to explore their individual requirements regarding such a concept as well as to collect first hand application scenarios from healthcare staff.

1 Introduction

Our research targets the development and evaluation of a computerised system that reminds its users of issues relevant for them and supports the collaboration between them in defining these reminders.

Initially, we became aware of our research problems due to our previous experience and work in the area of healthcare. In the treatment of patients with chronic conditions, we had to observe more than once how repeated examinations were undertaken, important analysis forgotten and treatment-related actions delayed. This increases cost of treatment, multiplies the workload of healthcare providers, causes discomfort for patients and undermines the outcome of their treatment. In particular, we have focused on the following problematic situations, which can benefit from the application of event notification services:

Problem 1: Alerting. Patients with chronic conditions have issues they have to deal with on a recurring basis. These may be to remember to take their medication, to keep their doctor appointments, to measure some physiological parameters or to document these measurements and other information such as pain patterns. It is cumbersome to keep track of all of these chores next to one's daily routine. Therefore, the compliance of these patients is not always optimal and could be improved by an appropriate support. To date, this support has been targeted only marginally.

Problem 2: Collaboration. Moreover, in the treatment of patients with chronic conditions it is essential for the contributing healthcare providers to jointly work on their patients' treatment regime. This avoids repeated examinations and thereby saves time, money and hassle. In addition, this enforces that relevant results found by one party are considered by other parties as well. It is vital to exploit the knowledge of all treating parties in order to optimise the treatment outcome. Patients themselves should contribute to this process, as they are the best experts on their own lives.

Problem 3: Context. The treatment of patients with chronic conditions continues on over extremely long patches of time - possibly lifelong. As circumstances in patients' daily lives vary, so does their treatment. For example, patients might change jobs and therefore develop a new pattern of their daily routine. Also, they gradually learn more about their condition and the treatment required for their particular condition will change with time. Therefore, as patients' lives and therefore their conditions' treatment changes, so should the support of the management of the treatment.

To target these problems, we suggest a mobile alerting system (MAS), which is aiming to support patients with chronic conditions in the management of their recurrent treatment-related issues [1].

We have identified the three problems mentioned above in a first cut requirements analysis we have presented in [1]. This analysis is based on several sources: We used results from interviews and experiences at a university clinic in Berlin. Furthermore, we drew from participant-evidence of condition-related newsgroups. Out of these sources, we then built up a use case development. The next step was to verify the requirements we have found. For this, we have run an online survey which was targeted at patients, doctors, nurses as well as computer scientists employed at IT departments of clinics [2]. Due to the limited number of clinical participants and computer scientists, we focused on the evaluation from the patients' point of view. Overall, they verified and extended the assumptions we had made over the requirements for a MAS for patients with chronic conditions.

Consequently, we decided to undertake another study aiming at healthcare staff. We chose to lead interviews with doctors and nurses. Into these interviews we have incorporated our extended original assumptions which have an emphasis on problems involved with collaboration between the various treating parties of a patient. In this paper, we present the results of the interviews we led.

In Sect. 2, the body of this paper commences with describing our suggested solution towards solving the presented problems. That section also comprises concept definitions. Section 3 describes our interviews, in particular the demographics of our participants, our research procedure and lessons learned from the interviews. We conclude this paper with a summary of our findings and a presentation of future plans.

2 Suggested Solution and Concepts

As already outlined in the previous section, we suggest a mobile alerting system (MAS) in order to target problem 1 (cf. Sect. 1): Patients or potentially their healthcare providers have to remember recurrent treatment-related issues at the appropriate

time. A MAS could be taken along similarly to a watch or a mobile phone. It could be set to remind of relevant tasks when required. This would be done by defining alerts using so-called profiles. One requirement on the system is to support collaboration between patient and treating healthcare staff (cf. Sect. 1, problem 2). For this, we have introduced the novel concept of collaborative profiles [3]. By using this concept, users are enabled to define their alerts and information needs jointly. This way, they help each other with specialised expert knowledge or general knowledge the other party does not have. We realised that the concept we found to support collaboration, at the same time also can help to express context. Users can simply define varying context and depending on the circumstances different alerts are sent to the user. Thereby, this approach can be employed to solve our third problem (cf. Sect. 1, problem 3).

As we realised that the concept of collaborative profiles we have proposed also is applicable for representing varying context [3], in our further research we focus on the realisation of the concept of collaborative profiles in combination with a MAS. It is based on our novel collaborative publish/subscribe model, which will be further developed in our future research (cf. Sect. 4).

In the following section we introduce definitions of the concepts used repeatedly in our research.

Context. Following [4], we regard any information that can be used to characterise the situation of an entity as context. For example, in our application scenario, this is, among others, the knowledge level of a patient regarding their condition.

Collaboration. Relating to the area of CSCW [5], our understanding of collaboration is that of work which is undertaken jointly with others. This may include collaborating partners which are not immediately connected, i.e. they might be situated remotely or co-located. Collaboration may occur either synchronously or asynchronously.

Event Notification Service. Event Notification Services are systems that inform their subscribers about certain facts they are interested in (cf. Fig. 1). The knowledge about these facts is provided by publishers, which send information about them to the event notification service. In the following these facts are referred to as events. Events are changes of the state of an object (e.g. a sensor). The interests of subscribers are defined in profiles using a profile definition language and registered with the event notification service. Whenever the event notification service receives an event which can be matched by one or several profiles, the respective subscribers are notified by a notification.

Collaborative Profiles. As identified in [1, 2], in several application domains profiles have to take into account the expert-knowledge and the specific contexts of several users. We therefore have proposed for several parties to collaboratively define profiles [3]. Profiles determine the filter criteria for the alerting system. Thus, collaborative profile definition leads to a collaborative specification of the filter functionality to ensure a context-based profile evaluation. It allows for an incorporation of the expert-knowledge of several parties into the filtering process.

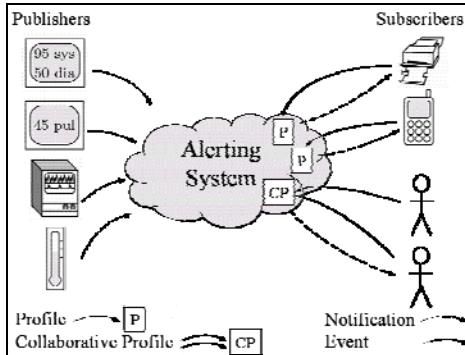


Fig. 1. Overview event notification service with collaborative profile

Figure 1 also shows the idea of several users collaborating for the profile definition: Several users share their knowledge and collaboratively specify and refine the profile. For example, after an initial definition of a patient's profile, other doctors and nurses subsequently refine the profile when the patient visits them. Depending on the patient's current condition (i.e. changing context), the patient may be alerted to take specific medicine, or doctor and patient may be alerted to a serious change in the patient's condition. Collaborative profiles may have to undergo several refinement steps as they may contain uncertain or vague specifications which have to be refined by other health practitioners.

3 Interviews

We undertook interviews with doctors and nurses in New Zealand and Germany. The aim of these interviews was to find out about healthcare staff's position towards our approach of supporting their work with an alerting system offering collaborative profiles. Additionally, we intended to gain an insight into further application scenarios deduced from their daily work with patients with chronic conditions. The interviews in Germany were conducted over a one week period in June 2005. Most of them took place outside of healthcare staff's work premises. The interviews in New Zealand were carried out at a university healthcare centre over a one week period in January 2006. We interviewed five people (three doctors and two nurses).

3.1 Participants

Our participants range in age from 31 to 61. In our survey we captured age, gender, nationality, experience with technology as well as their position/field of practice. Nevertheless, not all of these posed to be relevant for our evaluation. To be significant we found position, gender and nationality. In Table 1 we present a brief overview of our participants' background. The participants were recruited through direct contact. An attempt to recruit participants via notices on bill-boards failed.

Table 1. Overview of Participant Details

Position/Field	Hours Per Week Computer Use	Years of Computer Use	Experience Computer Use ¹	Gender	Age	Nationality
Haematologist / Oncologist	10	10	(0)(x)()	Male	61	German
ENT Doctor	0-10	11 work 15 private	(0)(x)()	Male	50	German
Nurse for Heart Disease	25	7	(0)(0)(x) work (0)(x)()	Female	34	German
Practice Nurse at Student Health Service	20	15	(0)(0)(x)()	Female	31	NZ
Medical Officer at Student Health Service	40-50	10	(0)(x)()	Female	55	NZ / Ex-South African

3.2 Study Method

Two of the interviews were conducted outside of the participants work premises and thereby gained a more informal character than the other interviews. Nevertheless, the direction of the results did not seem to be different than those, which took place in the participants' individual offices. During the interview only the interviewee and the researcher were present with the researcher taking notes of the participant's replies.

We briefly introduced our participants to the idea of a MAS and collaborative profiles using a figure to illustrate its use. Next, the participants were shown examples for collaborative profiles. Then, we went to the main part of the interview, asking the participants for their opinion and their expectations regarding collaborative profiles and the usefulness of MASs. Finally, they were asked to think about examples of their daily work in which a MAS and collaborative profiles would be beneficial for them.

3.3 Findings

All of our participants have taken up a positive stance on our suggested MAS as well as on the concept of collaborative profiles. As a general trend, we can state that women, New Zealanders and nurses were more open towards our ideas. Nevertheless, due to the small number of interviewees this can only be an impressionistic assertion.

Usefulness of Mobile Alerting System. All of our participants agreed that the idea to have a system that reminds patients and healthcare staff of important condition-related issues is very feasible and that there is a need for this kind of system. Yet, they varied in their opinion of which kind of scenarios they would like to employ that system for.

¹ The scale denoting the experience of computer use ranges from inexperienced (on the far left) to very experienced (on the far right).

The nurses we interviewed were most happy about the prospect of such a help, whereas doctors were also concerned about the consequences of potential errors made by the users of that system. One doctor preferred using the system only for simple applications. One of the nurses appreciated our idea for application in monitoring tasks because in opposition to humans the alerting system does not tire out (e.g. important for intensive care units). Another nurse stated that a MAS would be extremely suitable for supporting patients in homecare, who otherwise would forget their medication when they are on their own. She also put forward that it would be very helpful if the stored data would automatically be transferred to the patients' EHRs. One issue that arose was the fact that such a system lacks the intuition experienced personnel have acquired. Moreover, they wanted to be assured that the system sends reminders rather than to function as a decision help taking over decisions from healthcare staff.

All participants of the study agreed that if the system would be used correctly, i.e. if patients are compliant and data is input correctly, it would be very helpful.

Usefulness of Collaborative Profiles. Most of our interviewees highly favoured our idea of supporting collaboration between patients and different parties of healthcare providers by the help of collaborative profiles.

All of the participants were open to use the concept for simple cases. Most of them were positively inclined towards the idea for general applications without too many concerns. They stated that it is excellent to support collaboration because currently in the New Zealand healthcare system, A does not know what B is doing and vice versa. They assessed that this concept would be helpful because of the support of multidisciplinary involvement and communication processes. Another good use would be the incorporation of lab results into decisions over a patient's medication regime. They appreciated the system's usefulness for supporting the work in a nursing home, presuming it was ensured that it will be kept track of who is manipulating alerts. This is to avoid that settings are erroneously changed.

All of this put together, shows that the idea is very good as it would save time and money. This way, extra resources would be available to look after the patients.

Expectations of Healthcare Staff regarding Collaborative Profiles. We had to learn that our participants even though they highly appreciated our ideas had trouble relating information about their daily work directly to our idea of collaborative profiles. Some of their replies concerned the general idea of a mobile alerting service rather than the support of collaboration. Nevertheless, the first two issues we can pinpoint were concerned with collaboration:

Communication. In particular, the nurses that participated in our interviews were extremely interested in a support of communication in all directions. Also, one of the doctors pointed out the interest in communicating with other doctors.

Liability. This issue was the most controversial among our participants. German healthcare staff seemed to take on a different stance than healthcare staff from New Zealand. This might originate in their different cultural backgrounds. Germans were deeply concerned about liability problems regarding collaboration, whereas New Zealanders did worry less about this matter.

Adaptability. Since every patient is different, adaptability was of importance to a lot of participants. The system should be adaptable to various needs of their patients. However, they welcomed the idea to offer default profiles so that healthcare providers have a basis on which grounds to start working (e.g. based on the guidelines of the NZ Guideline Group (e.g. [6] for the management of diabetes)).

General. Other issues that arose covered concerns about handling the system and its consequences. Essential information should not be filtered out, data has to be entered correctly and an automatic transfer of the data to patient EHRs is indispensable. One doctor was calling attention to the requirement to support patients in developing their independence regarding the management of their condition, rather than being driven into lethargy concerning their condition management.

Further Application Scenarios for Collaborative Profiles. Clarifying the question of further application scenarios for the concept of collaborative profiles posed the most challenging. Our participants confused applications for the MAS with simple profiles with the suggested collaborative approach. However, some remarks were related to our concept. On the one hand, they suggested the use for organisational matters in a clinic, e.g. in order to coordinate x-ray examinations with nurses and patients. On the other hand, they proposed to support collaboration between different medical establishments with our system.

4 Conclusion and Future Work

In the beginning of this paper, we identified three problems repeatedly occurring in the treatment of patients with chronic conditions: the need of alerting patients and healthcare staff about condition-related issues, supporting their collaboration and considering their changing context for the management of a patient's treatment (cf. Sect. 1). To target the first problem, we have suggested employing a MAS. To solve problem 2, this system has to support collaboration between different parties. It does so by using the proposed concept of collaborative profiles, which is based on the novel collaborative publish/subscribe model and is also suitable for capturing different contexts, solving problem 3.

We undertook a study with healthcare providers to analyse their position towards our suggested solution. In the course of this study, we undertook interviews with doctors and nurses in New Zealand as well as in Germany. From the findings of these interviews we have learned several lessons:

Lessons Learned. The first three conclusions we can draw from the findings of our user study. The last lesson is deduced from the research involved with our study:

Lesson 1. Healthcare providers appreciate the idea of an alerting system for patients with chronic conditions provided the guarantee of an appropriate use of the system.

Lesson 2. Our participants positively accepted the concept of collaborative profiles. Most important for the realisation of this concept they found support of a multi-directional communication, a well-defined approach for liability issues and the possibility to adapt the system to their patients' individual needs.

Lesson 3. Even though healthcare providers are affirmative of our idea, they have difficulties to immerse themselves into the concept of collaborative profiles and cannot properly relate it to their work in a short interview. Thus, there is a need to realise our ideas and to evaluate them in a real-world scenario rather than to merely discuss them.

Lesson 4. The suggested concepts are not only applicable to the area of healthcare but also to other domains which require a collaboration of several people. This occurs e.g. in e-commerce, facility management or tourism. We therefore have concluded to continue our research on a more abstract level in order to gain universally valid results.

Future Work. According to lesson 3 and 4, we will focus on the realisation of collaborative profiles. This will involve both, HCI aspects as well as IS-related research.

In particular, we will look into the development of an effective interface for the definition of profiles for event notification. Based on this research, we will develop a profile definition language which can express the concepts required for collaborative profiles. In order to find this language we will develop a conceptual framework for the subsumption of collaborative profiles. To enable us to evaluate these mechanisms, we will build up an effective interface for the collaborative definition of profiles.

Finally, this research will enable us to test our concept of mobile alerting and collaborative profiles. According to lesson 4, this evaluation will be undertaken independently of the application area healthcare to lead to universally valid results.

Acknowledgements. We would like to thank our study participants in both Germany and New Zealand for sharing their thoughts and experiences with us. Furthermore, I would like to thank my late father for setting up some of the interviews in Germany.

5 References

1. Jung, D., Hinze, A.: A Mobile Alerting System for the Support of Patients with Chronic Conditions. European Conference on Mobile Government (EURO mGOV), Brighton, UK (2005)
2. Jung, D., Hinze, A.: Patient-Based Mobile Alerting Systems - Requirements and Expectations. Health Informatics in New Zealand Conference (HINZ 2005), Auckland, New Zealand (2005)
3. Jung, D., Hinze, A.: Capturing Context in Collaborative Profiles. OTM 2005 Workshop on Context-Aware Mobile Systems (CAMS 2005), in conjunction with OnTheMove Federated Conferences (OTM 2005), Agia Napa, Cyprus (2005)
4. Dey, A.K.: Understanding and Using Context. Personal and Ubiquitous Computing 5 (2001) 4-7
5. Dix, A., Finlay, J., Abowd, G.D., Beale, R.: Human-Computer Interaction. Pearson/Prentice-Hall, Harlow, England; New York (2004)
6. Evidence-based Best Practice Guideline: Management of Type 2 Diabetes. New Zealand Guidelines Group (NZGG), Wellington

A Comparison of XML and Relational Database Technology

Christof Lutteroth

Department of Computer Science
The University of Auckland
38 Princes Street, Auckland 1020, New Zealand
lutteroth@cs.auckland.ac.nz

Abstract. This paper describes differences between XML and relational database technology. In our project we had to make a choice between them which led us to examine their functional as well as non-functional differences. We found that XML, although widely used, is bound to a particular set of requirements and limitations. It cannot replace the relational data model, which ended up being the one we chose.

1 Introduction

This paper describes some of the work that has been done in the context of a project to develop a platform for model-based software engineering. This platform, the AP1 system [1], consists of a repository for storing models and data instances of models, a common user interface for the integration of CASE functionality, and a model-based generator technology. AP1 is essentially a set of components that can be reused by CASE tools for typical tasks, such as data management, data manipulation and data generation. The components are designed in a way so that tools using them can be integrated easier. The common data repository, for example, lets tools exchange data and communicate through an event notification mechanism. That is, if one tool changes a particular data set, then another tool can be notified. A generic editor lets users change the data in the repository, no matter what tool created the data originally. Tools can also be integrated with the generic editor as plug-ins, so that they can reuse its GUI and its data modification capabilities. Because different tools may use different models for similar data, AP1 also offers a way of specifying generators that transform data of one model into data of another one. This helps to bridge the differences between tools and make them more interoperable.

For the repository we had to make a choice between relational database and XML technology. What we need is a flexible and efficient way of storing and managing the data that is used in CASE technology, e.g. source code or software models such as class or user interface diagrams. The data of CASE tools can be very complex and voluminous. Note that the decision about the underlying database technology is a very important one and should not be made unmindfully. It has a strong impact on all the parts of the project because data

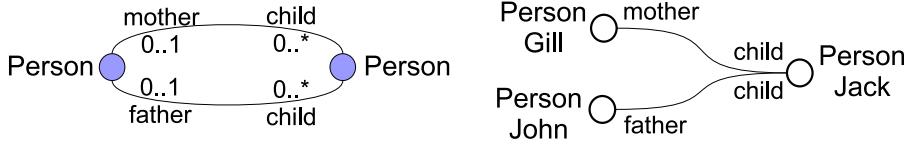


Fig. 1. A simple PD model (left) and one of its data instances (right).

management plays such an essential role. Different database technologies do not only affect how data is represented internally but also how it can be accessed and modified. They can have a significant impact on the ease with which CASE tools can be implemented. While many similar projects use XML in their data backend, several reasons made us rethink the suitability of XML for these purposes. Regarding the popularity of XML and the way it is pushed by its advocates, we thought it to be informative to describe some of these reasons in this paper.

Section 2 describes the relational data model that we ended up using for our project. Section 3 discusses many aspects of the differences between XML and the relational data model (RDM). In Sect. 4 we describe some of the shortcomings of XML. The paper concludes with Sect. 5.

2 The Parsimonious Data Model

For our modeling platform we use the parsimonious data model (PDM) [2], which is a very simple relational data model. It abstracts from some of the details necessary in relational database schemas. The left side of Fig. 1 shows a concrete PDM data model describing two recursive relations between instances of a type Person, the mother-child and the father-child relations. The filled circles represent entity types and the lines between them represent relation types. The only entity type in this model, Person, is represented twice in order to disentangle the recursive relation types and thus make the diagram clearer. At the end of each relation type we can see role names and multiplicities. The 0..1 multiplicity at the mother role, for example, signifies that in each instance of the model a Person can have at most one mother associated with them. This is different to relational database schemas, which reflect the multiplicities of relation types in the structure of their relations and not just in additional constraints. A change in the multiplicities of a relational schema likely changes its structure, e.g. adding an additional relation, while the topology of a PDM is invariant with regard to changes of multiplicities. The PDM also allows us to model inheritance. Its underlying relational structure allows for a clean mapping onto a relational schema, so that RDBMS technology can be efficiently leveraged for storage and retrieval. The right part of the figure shows a model instance of the model on the left side. The empty circles are entity type instances and the lines between them links of relations.

3 XML and the Relational Data Model

The XML and the relational data models are logically equivalent. For each XML schema a logically equivalent relational schema can be created and vice versa. However, many XML-relational mappings are possible. A focus of research has been the storage and retrieval of XML data on the basis of relational database technology, e.g. [3–6]. The simplest approaches store XML data by creating edge tables that contain all elements and attributes of XML documents and link them with foreign keys. When processing XML queries, which are based on path expressions, such approaches require many joins along the path denoted in such a query. This is not performance efficient, therefore more advanced approaches exist that combine larger parts of an XML schema into single tables. Such tables use several columns to store multiple edges in a single row, an approach known as inlining, so that less joins are necessary. But it is usually not entirely possible to avoid the “shredding” of XML data into many tables and rows, and the resulting necessity for joins. However, by using specialized indexes queries based on path expressions can be evaluated very efficiently on relational database management systems (RDBMS) [7], and most big commercial RDBMS provide special support for XML data. In the following sections we want to discuss more of the differences between XML and relational data.

3.1 XML versus Relational Data Representation in a DBMS

In general, using the physical structure of an XML document for storage and retrieval of that document has advantages and disadvantages. If the structure of a document is left as is, storage is fast because no restructuring is needed. The queries on XML data follow the tree structure of the document, so having the document stored in its original structure offers performance benefits for query processing as well. Data that are accessed together because they are closely related in the tree structure are likely to be stored on the same memory pages, resulting in a good locality and less pages to be read. However, if the stored documents are modified, having to preserve the document’s original structure in memory may not be possible without a loss of performance. If, for example, a big subtree is inserted into the tree of the document, the memory structure has to be reorganized in most cases to make space for the insertion. The mismatch between the linear structure of conventional computer memory and the tree structure of XML results potentially in memory collisions and fragmentation. In contrast to this, the tables of a RDB are inherently linear and unordered so that memory management is more flexible and such problems can be avoided, possibly at the cost of a reduced locality. Only the technique of clustering, which is sometimes used in RDBMS, results in hierarchical physical structures similar to those of XML, having the advantages and disadvantages similar to the aforementioned ones. If an XML document makes use of element references, such references have to be resolved during query processing, similar to relational joins. In such cases having the XML tree structure in memory does not provide performance benefits. DBMS that use XML as the basis of internal data representation may

be suitable for information storage and retrieval with path expressions, but not for data that is updated frequently.

3.2 Data Integration

XML is frequently introduced as the de facto standard for exchanging and integrating data on the Web. It is important to understand what makes XML suitable for data integration and how it is actually related to the Web. First of all, one has to note that, like any multi-purpose data model, XML is inherently only capable of syntactic data integration. By storing data in XML, we can read and access its structure, using an XML parser. This, however, does not mean that we understand the meaning of the data and can relate it to other data in a meaningful way. Syntactic integration is a prerequisite for, but does not include, semantic integration. As described, for example, in [8], semantic integration can be achieved by translating between different concrete data models. XML comes with accompanying technologies like the Extensible Stylesheet Language (XSL), that can be used to define transformations between different XML schemas or from an XML schema to a different representation. This, however, is not a unique property of XML. There exist many other methods for data transformation.

3.3 Self-Descriptiveness

XML is often described as being self-descriptive. It is important to be aware of the degree to which XML documents describe the data they contain and how this differs from other data models. XML is a textual format, and thus it can be read with every text editor. It is therefore often referred to as human-readable, but this alone only contributes to accessibility but not to its self-descriptiveness. The fact that data is human-readable does not imply that it is human-understandable. XML data can be typed, which means that an XML document can optionally specify a schema that describes its structure. Apart from the possibility of a well-specified structure, self-descriptiveness of XML documents comes down to the fact that XML data elements have textual labels which serve as their names. If these names are chosen well then they may betray the semantics of the respective data elements to a human reader. In general, a machine operates only on a syntactic level. It may infer a schema for an XML document, if it is not already given.

Is XML more self-descriptive than other data models? We want to make a comparison with the relational data model as it is used in SQL. First of all, relational data always has a schema, and a relational schema is well-defined, similar to an XML schema. All tables and columns of such a schema have names, similar to the element and attribute names of an XML document. SQL is a textual language and can be used for data as well as for metadata. It incorporates a data manipulation language (DML) for data modification and retrieval and a data definition language (DDL) for the specification of data types and constraints. With SQL one can thus incorporate a relational schema as well as data in a

single document. Relation and attribute names would be just as much part of such a document as element and attribute names in XML.

3.4 Usability

XML emerged in the late 90's, which was the time of the big bubble of web-based e-commerce. In spite of the damage the web-hype has done to the reputation of Internet technology, the World Wide Web has established itself firmly as the primary medium of the digital world. HTML has gained huge popularity, and its simple textual markup structure has fostered a lot of end-user development. This trend can be seen as an important historical factor for the development and proliferation of XML. HTML is a user interface description language and therefore tailored to the representation of data. XML is a general-purpose data model because it does not imply any particular semantics. Users are free to, or rather compelled to, specify the semantics on their own. This is what is meant by XML's separation of content and layout.

When comparing XML and textual relational notations of data it becomes clear that the nested, hierarchical way used in markup languages is often easier to handle for humans than the relational graph representations consisting of sets of edges. The reason is that the hierarchical representation groups together related data elements so that logical association is reflected in the physical locality of the representation. In contrast to this, the elements in a set of edges do not possess such a hierarchical structure but exist usually in a plainly linear representation. An example for this are SQL insert statements, which are logically equivalent to sets of tuples.

A lot of data have a mostly hierarchical structure, that is why the hierarchical markup representation is adequate. However, if data are not mostly tree-structured but have, for example, merely the structure of a directed acyclic graph, then the ease of use of such a notation diminishes and tool support becomes more important. Hierarchical textual representation becomes less and less suitable with the degree to which the data deviates from a purely tree-like topology because the degree to which associated data can be grouped locally decreases.

In XML we have to make use of node references using node identities in order to specify associations that do not fit into a tree, or specify data redundantly. Unfortunately there is a disparity between the identity of XML data by element identity and the identity by data value [9]. Element identity is representation dependent because data can be represented redundantly, i.e. the same data can be represented in several duplicate XML elements. Value identity is purely on a logical level, and the only choice in SQL.

One has to note that with proper tool support, e.g. a structural editor, the question about the underlying physical representation of data is irrelevant. The user only sees the data through the user interface of the tool, so what matters is the representation of the data in the user interface. A good tool offers several representations, so that the user can choose if, e.g., the data are represented as a tree or in a graph-like visualization. On this level the physical representation may

at most affect non-functional requirements. XML is known to have shortcomings with regard to performance, space consumption and security [10].

A property that certainly encourages the use of XML is the possibility to define ad hoc structures very easily. The syntax of XML is relatively simple and data can be stored without previously specifying its schema. While this may be an advantage if the aim is to store and retrieve small amounts of data relatively easily, it can become a serious problem on a larger scale. In many organizations XML data is used without having a proper schema definition, and in the long run this leads to confusion and additional cost. Furthermore, myriads of different competing XML-based standards have emerged over the years in nearly all domains, with only few of them being widely accepted. In the domain of e-commerce, for example, there are many incompatible XML-based “standards” [8], and even more in-house standards used by individual companies. XML has not solved the interoperability problem caused by the presence of too many competing standards, but rather lifted it onto the level of different XML schemata and semantics.

4 Problems of XML

There are cases where XML distinguishes between different representations of data although they are logically equivalent. This significantly drags down the level of abstraction and complicates data integration and the language as a tool itself. For example, XML introduces the notion of attributes, although they are equivalent to leaf elements, i.e. elements without further subelements. It means that schema developers frequently have to make irrelevant decisions whether a data value should be stored as attribute or leaf element, but it is exactly those arbitrary decisions that can render schemas incompatible. Another distinction is that between subelements and references: XML distinguishes between nested elements and references to equivalent elements that are defined somewhere else. In general, distinctions in the representation that are not founded in the logical structure of data and are not made transparent to users are an unnecessary source of incompatibility. In the following sections we want to discuss briefly some of the shortcomings of XML.

4.1 Ordered Data Elements

In XML elements are always ordered, and this is in many cases an overspecification. Even if the order of elements is not relevant for an application, it has to be maintained by a DBMS, creating an overhead. Hence, the DBMS loses flexibility when managing the data: either it has to use the order in memory or maintain additional ordinal data. Furthermore, having order on data when it is not part of a system’s specification violates a rigorous information hiding principle: a user may exploit document order to create behavior that does not conform to the specification. RDBMS do not have to care about the order of records and can therefore manage storage space more easily. Order is only sometimes needed for

real data, and if so, often naturally given by ordinal attributes, e.g. a timestamp. If order of data is important, it can be modeled in the relational schema, see for example [11].

4.2 The Root Problem

XML not only requires data to have a mostly hierarchical structure, but also requires it to have a single natural hierarchy in order to be suitable as data representation. The standard ways to query elements in XML, XPath and XQuery, rely on path expressions that identify an element in the tree by the path from the root. Path expressions as a way of addressing elements can have a severe impact on the way XML data is stored, e.g. see [5]. Unfortunately, for many data models the choice of the root is rather arbitrary, i.e. many different roots would be possible. Let us look at two different XML representation of the model instance on the right side of Fig. 1:

```
<person> <name>Jack</name>
    <mother> <person> <name>Gill</name> </person> </mother>
        <father> <person> <name>John</name> </person> </father>
    </person>

<person> <name>Gill</name>
    <child> <person> <name>Jack</name>
        <father> <person> <name>John</name> </person> </father>
    </person> </child>
</person>
```

The two variants look very different and have very different schemas, but essentially contain the same information. Child is simply the role on the other end of the mother-child association that can exist between two Person instances. And we could produce many more such variants for the same data. The problem of having to choose a root element may recursively reappear for several subtrees of an XML document. This makes the representation of data in XML highly ambiguous and the identification of data elements using path expressions very much dependent on the particular representation that is chosen for a data model. This can, in fact, be a strong hindrance to data integration, since we have to be unnecessarily strict and enforce the use of one particular, arbitrary representation. The relational data model does not have this anomaly.

5 Conclusion

Considering the different arguments it became clear for us that XML does not match the requirements of our model repository. XML is not a guarantee for successful data integration but just a tool, and like any other tool just as good as its user. There are many XML-based standards, but many such “standards” sprout up like mushrooms, which can be a source of instability. XML mingles issues of representation into the main purpose of a data model, i.e. storage,

manipulation and retrieval of data on a logical level. Therefore, XML cannot achieve the same level of abstraction as the RDM, which relies on a very simple, well-defined, mathematical notion.

References

1. Lutteroth, C.: AP1: A platform for model-based software engineering. In: TEAA '06: Proceedings of the 2nd International Conference on Trends in Enterprise Application Architecture, Springer (2006)
2. Draheim, D., Weber, G.: Form-Oriented Analysis - A New Methodology to Model Form-Based Applications. Springer (2004)
3. Florescu, D., Kossmann, D.: Storing and Querying XML Data using an RDMBS. *IEEE Data Engineering Bulletin* **22**(3) (1999) 27–34
4. Shanmugasundaram, J., Tufte, K., He, G., Zhang, C., DeWitt, D., Naughton, J.: Relational Databases for Querying XML Documents: Limitations and Opportunities. *Proceedings of the 25th International Conference on Very Large Data Bases* (1999) 7–10
5. Khan, L., Rao, Y.: A performance evaluation of storing XML data in relational database management systems. *Workshop On Web Information And Data Management* (2001) 31–38
6. Amer-Yahia, S., Du, F., Freire, J.: A comprehensive solution to the XML-to-relational mapping problem. In: WIDM '04: Proceedings of the 6th annual ACM international workshop on Web information and data management, New York, NY, USA, ACM Press (2004) 31–38
7. Weigel, F., Schulz, K.U., Meuss, H.: Exploiting native XML indexing techniques for XML retrieval in relational database systems. In: WIDM '05: Proceedings of the 7th annual ACM international workshop on Web information and data management, New York, NY, USA, ACM Press (2005) 23–30
8. Díaz, L., Wüstner, E., Buxmann, P.: Inter-organizational document exchange: facing the conversion problem with xml. In: SAC '02: Proceedings of the 2002 ACM symposium on Applied computing, New York, NY, USA, ACM Press (2002) 1043–1047
9. Krishnamurthy, R., Kaushik, R., Naughton, J.: Unraveling the duplicate-elimination problem in XML-to-SQL query translation. *Proceedings of the 7th International Workshop on the Web and Databases: colocated with ACM SIGMOD/PODS 2004* (2004) 49–54
10. Vaughan-Nichols, S.: XML Raises Concerns as It Gains Prominence. *IEEE Computer* **36**(5) (2003) 14–16
11. Tatarinov, I., Viglas, S.D., Beyer, K., Shanmugasundaram, J., Shekita, E., Zhang, C.: Storing and querying ordered XML using a relational database system. In: SIGMOD '02: Proceedings of the 2002 ACM SIGMOD international conference on Management of data, New York, NY, USA, ACM Press (2002) 204–215

Designing Ethical Investigation on Password and Online Account Management Strategies

Gilbert Notoatmodjo

Department of Computer Science, The University of Auckland, New Zealand
gnot002@ec.auckland.ac.nz

Abstract. At the present time, password is arguably the most common authentication method across the internet. In this paper we present some background information on digital identity and current issues with password authentication scheme. We provide a brief overview of our research, which involves a survey-based study on password management strategies. We also describe our ethical concerns regarding our study and how they affected our survey design. Finally, we discuss current progress as well as future directions of our research.

Keywords: Internet Security, Identity Management, Password, Survey, Ethics

1 Introduction

The concept of identity in the digital context is mostly related to the digital representation of entities, including both human and non-human, which consists of one or more attributes. Some of these attributes of digital representations only exist as a result of storing information in digital format and are only used for recording purposes, whereas some are used to distinguish different users of a system. This is done to satisfy the security objectives of the system. Most security requirements can be generalized into *confidentiality* (resources can only be read by authorized parties), *integrity* (resources can only be modified by authorized parties), and *availability* (all authorized parties are always able to perform the operations for which they have the rights to) [1].

Two steps are involved before a user is granted access to a resource. The process of verifying the identity of an entity is known as *authentication*, whereas the process of granting privileges or access to a resource is known as *authorization* [1]. There are different types of attributes which are normally used to authenticate individuals. Most of them can be classified into these categories [2, 3]:

1. *Knowledge-based authentication* (what the person knows)

In knowledge-based authentication, the identity of a person is verified by demonstrating a piece of information which is expected to only be known by that person. For examples passwords and PINs (Personal Identification Number).

2. *Token-based authentication* (what the person has)

In token based authentication, the identity of a person is verified using physical evidence or token that he/she possesses, such as smart cards, RFID tags, and credit cards.

3. *Biometrics* (what the person is)

In this type of authentication, measurable physical traits of a person are used to verify his/her identity. Examples of this are fingerprints, face recognition and iris scanning.

There are occasions in which combinations of these attributes are used in authentication process, such as bankcard and PIN. This is known as *multi-factor authentication* [3]. In the next section, we discuss current issues with password authentication, a widely used knowledge-based authentication scheme, which is also the main focus of our research. The rest of the paper is structured as follows Section 3 presents a brief description of our research. Section 4 describes our survey-based study on passwords and online accounts, including our ethical concerns and considerations that brought a large impact on our survey design. Finally, section 5 concludes the paper.

2 Current Issues with Password Authentication

The origin of password authentication could perhaps be traced back to the ancient secret societies, when secret phrases are used as one of the means of identification among their members [4]. At the present time, passwords are arguably the most common authentication method across the internet.

The security of password authentication mechanism hinges on the secrecy of a single phrase. If an adversary obtains knowledge of a victim's secret phrase, the adversary will be able to impersonate the victim and gain access to the resources which the victim is entitled to. Although cryptographic means and protocols offer a degree of protection during the transmission and storage of the secret phrase in the system end, the user end is often left unprotected by nothing but security policies and guidelines which are often neglected [5, 6].

There are various kinds of attacks which can be attempted by an attacker in order to obtain a victim's password. Some of these attacks work by targeting password during transmission, such as eavesdropping, replay and man-in-the-middle attacks [7]. Other attacks, such as dictionary attacks are directed to stored passwords in the server end [8]. Some tools required to launch these attacks are readily available on the internet, although most of them require some level of technical knowledge. Examples include network sniffer (such as *tcpdump* [9] and *wireshark* [10]), rogue Wi-Fi access point (such as *airsnarf* [11]) and password cracking utilities (such as *John the Ripper* [12] and *RainbowCrack* [13]). Service providers usually attempt to prevent these attacks by encrypting the communication between users and server (using SSL, for example), encrypting and limiting access to stored passwords, and blocking accounts which have too many incorrect login attempts [7, 8].

Whereas these attacks are reasonably dangerous and have the potential to yield serious damage if successful, there is another type of attacks which is even more frightening. This type of attacks is known as *social engineering*, and is directed to

users, the least protected part in the mechanism. Most social engineering attacks require minimum level of technical knowledge and yet have relatively large chance of success. Phishing is an example of social engineering attacks in which the attacker attempts to obtain user information such as passwords and credit card details by masquerading as a legitimate party, usually through e-mail communication [14, 15]. This is sometimes used in conjunction with URL obfuscation and graphical tricks to mislead unsuspecting victims to bogus websites, which often look very similar to their legitimate counterparts [16]. Pharming refers to a similar, yet more sophisticated type of attack, whereby the attacker manipulates victim's local DNS address in order to redirect user to a fraudulent website [14, 15]. Based on a survey conducted in 2004, Gartner Research estimated 1.78 million Americans had given sensitive or financial information to phishers [17].

Malicious server attack is defined in [18, 19] as another type of social engineering attack, in which the attacker sets up a malicious server, and lures the victim to open an account in order to gather victim's passwords. If the victim uses the same password for more valuable accounts, attacker will be able to impersonate the victim with ease and gain access those accounts. In addition, since users tend to base their passwords on permutations around the same word [5, 20], it would be logical to combine this attack with dictionary attack. The attacker could also attempt to gather as much information as possible from the victim by asking the victim to provide personal details during the registration process. As if these were not enough, the attacker could also disable victim's access after a period of time, leading the victim to reveal other passwords upon login failures.

It is clear that this attack benefits from reckless password reuse practices. Although as far as we are concerned there is no live data or statistics indicating damages caused by this attack, we believe that this attack is highly plausible, and would be very logical to attempt by someone motivated by financial gains. Ives et al [21] highlighted that careless password reuse also has the potential to add 'domino effect' to successful attacks of any kind. While human memory capacity is very unlikely to increase significantly over the next few years, the rising number of online services will force most users to reuse their passwords.

Federated identity management attempts to address this problem by enabling different service providers within a federation to recognize user's credentials. This eliminates the need to create a different password for each service provider, which lessens the number of passwords that have to be maintained by users. There are different approaches by which this can be achieved, but we will not discuss the details in this paper. Examples of federated identity management systems include, Windows Live ID (formerly Passport) [22] and Liberty architecture developed by The Liberty Alliance Project [23].

There are currently varying opinions on federated identity management. On one side, federated identity management provides allows users to manage fewer passwords, promoting more secure practices and better awareness [14]. On the other side, this would mean that a successful attack to an account would almost ensure that other accounts within the same federation are also compromised. Recent report has shown how vulnerability in MSN Passport allows attacker to steal credit card details by exploiting the single sign on mechanism [24]. Another consequence of federated identity management would be that users have less control over which credentials

should be associated with which services and virtually no control over which services should be grouped together. Furthermore, although this approach should reduce the number of passwords that have to be maintained, users experience will not improve by much as the number of service providers increases significantly [25].

3 Overview of Our Research

We believe that before we could review the existing solution, or work towards an appropriate solution to the aforementioned password management problems, it would be essential to study how people actually manage their passwords. We agree with Gaw and Felten [20] who stated that many projects focus on developing solutions to solve poor password practices without studying the actual problems.

As we have discussed, it is a known fact that most people reuse their passwords and that it would be unrealistic to expect users to create a completely unique password for each of their accounts. However, there is one intriguing question that in our opinion has not been thoroughly explored: how do people actually reuse their passwords? The main focus of our research is to investigate the strategies by which users manage the associations between their accounts and passwords by conducting of a survey.

There are many survey-based studies that have been done on the subject on passwords but most of them have specific objectives which are different from ours. A survey conducted by CentralNic, a London-based internet domain name registry, which involved 1200 people from 30 companies throughout the UK [26]. Their study revealed four different categories of users based on the inspirations from which they created their passwords; 48% were family oriented, 32% were based on famous characters, 11% were based on fantasy, while 10% used combinations of letters and numbers based on security considerations. Adams and Sasse [5] used web based questionnaire and in depth interview among employees of several organisations, and found that many users devised their own methods to increase password memorability by linking multiple passwords through some common elements (such as name1, name2, name3 and so on).

Brown et al. [27] conducted a survey among 218 university students. Their study indicated that their respondents had on average 4.45 ($N=218$, $SD=1.63$) unique passwords to be used with in average 8.18 ($N=218$, $SD=2.81$) different items. A study by Riley [28] also reported similar findings; their respondents on average had 3.1 ($N=232$, $SD=2.028$) passwords that were used with 8.5 accounts. Riley also reported that around 40% of the respondents varied the complexity of their password depending on the nature of accounts. Another recent study conducted by Gaw and Felten [20] suggested that there are different reasons why people decided to reuse their passwords for several websites, but use unique passwords for several others, such as ease of recall, importance of the information stored, security and so on.

In our study, we aim to explore the relationship between accounts and passwords by looking at their similarities from users' perspectives, investigate the mapping patterns of accounts and passwords by interviewing a number of users, and to finally create a framework of these mapping patterns based on the interview results.

Before we could proceed with our survey procedures, our university regulation [29] requires us to obtain an approval from the University of Auckland Ethics Committee. During the application process, we realized that there are many ethical issues that need to be considered. In the next section, we discuss our views and ethical concerns and how they affected our survey design.

4 Designing an Ethical Survey

4.1 Ethical Issues, Concerns, and Considerations

There is a belief in the society which considers science to be ethically neutral. This argument is likely to be based around Hume's classical proposition that no 'ought' can be correctly inferred from 'is', implying that no moral proposition can be derived from non-moral propositions [30]. Many people believe that since ethics is primarily about moral judgment, it cannot be inferred from science, which is often perceived to only deal with facts. In contrary to this belief, Bronowski [31] stated, "Those who think science is ethically neutral confuse the findings of science, which are, with the activity of science, which is not". According to his view, science is affected by human views and interpretations, and thus cannot be considered ethically neutral.

Despite these arguments, we realized that in any study that involves human participation, there are often trade offs and ethical considerations that have to be made concerning the amount of information that we could collect without violating the rights or endangering the safety and well being of the participants. It is still fresh in our memory how unethical experiment at National Women's Hospital in Auckland resulted in severe physical consequence to a number of participants involved [32]. We also realized that although our study imposes almost no risk of physical injury, it involves very sensitive information, which if exposed, could result in both privacy and financial loss to the participants.

The University of Auckland Human Participants Ethics Committee Guidelines [29] defines 8 ethical principles which govern research involving human participants, which are:

1. Informed and voluntary consent
2. Respect for the privacy rights of participants
3. Social and cultural sensitivity
4. Acknowledgement of the Treaty of Waitangi
5. Soundness of research methodology
6. Transparency, and the avoidance of conflict of interest
7. Minimization of harm
8. Principles relating to human remains, tissue, and body fluids

We consider points 3, 4, and 8 to be irrelevant, because our study involves neither socio-culturally sensitive materials nor human remains. Among the rest, there are two principles that we believe to have largely impacted our survey design. The second principle implies that we have to respect the privacy rights of research participants. As we have discussed, ideally, we would need to gather real passwords and accounts from participants. However, asking participants to disclose such materials would be

impossible without compromising their privacy rights, and even if we somehow managed to collect such information, we would have to handle our data with extreme care, as exposure of these data would result in major loss of privacy. The seventh principle implies that we have to ensure the risk of the participants being harmed is minimized. In our study, the most likely cause of harm is financial damage caused by the leakage of participants' passwords and accounts information. If we decided to collect real passwords and accounts from participants, we still could end up in a very difficult situation if participants' accounts were compromised by other means completely unrelated to our study, even though we tried our best to minimize this risk.

This is a complicated situation. We need to collect information about password and accounts from the participants in order to achieve our research goals, whereas after considering the ethical issues, it seems that asking participants to entrust their actual passwords and accounts to us would be prohibitive. The only way to do this would be to collect descriptions about participants' passwords and accounts without asking them to disclose their actual passwords and accounts, but even this has to be done in such a manner to ensure the associations of their passwords and accounts could not be directly inferred from the descriptions they provide. Thus a trade off has to be made between the amount of information we could gather and the potential risks exposed to participants.

4.2 Survey Design

We decided to use a numbering scheme to allow participants to describe their passwords, accounts and associations between them without revealing their passwords and accounts. A potential weakness of this approach is that the procedure would be mentally exhausting, as participants would have to describe their passwords and accounts without writing them down. A possible way to overcome this would be to ask them to write down this information separately as an aid, and to ask them to destroy this information at the end.

Our survey design comprises of a one-to-one interview, consisting of two main parts. In the first part, the participants are asked a few general questions about their computing experience and how they record their passwords and online accounts. The second part consists of a guided exercise, which is divided into five steps:

1. Participants are asked to write all their passwords in a piece of paper.
2. Participants are asked to complete a table by assigning numbers and codes to their passwords based on the way they group their passwords.
3. Participants are instructed to complete a table to describe their passwords, without writing down their actual passwords, by using the numbering scheme from the previous step as an aid.
4. Participants are told to write their online accounts in a table and assign numbers and codes to them, based on the way they group their accounts.
5. Finally, participants are asked to complete a table to describe their online accounts and the associations between their online accounts and passwords, by using the numbering scheme assigned in previous steps.

After the procedures are completed, participants are instructed to separate the sheets containing their actual passwords and accounts from step 1, 2, and 4, and destroy them using a commercial grade strip-cut paper shredder.

5 Conclusion and Future Work

We have described the motivation and focus of our research, which intends to investigate password management strategies employed by users. Our analysis suggests that although various improvements have been proposed, we believe that most users would still have to rely on their own password management strategies, at least in the foreseeable future. Thus, it becomes necessary to study the actual problems with these strategies before making further improvements or working towards an appropriate solution. We also discussed some of our ethical concerns regarding our survey-based study, which involves sensitive information and how these concerns affect our survey design from a layperson's perspective. We believe that this could be of help to researchers who are considering to conduct studies involving human participants in the future.

This project was undertaken as a requirement for the degree of Master of Science at the University of Auckland. At present, we have collected data from 26 respondents from different faculties across our university in our pilot study. In the future, we hope to improve our survey design based on the results of our pilot run and possibly extend our target population to include participants from a wider range of background and professions in both New Zealand and overseas.

Acknowledgments. The author would like to first and foremost acknowledge Clark Thomborson as the project supervisor. The author would also like to thank Stephen Drape, Anirban Majumdar and David Leung for their support and constructive comments.

References

1. Lampson, B.W.: Computer Security in the Real World. *IEEE Computer* **37** (2004) 37 - 46
2. Clarke, R.: Human Identification in Information Systems:Management Challenges and Public Policy Issues. *Information Technology & People* **7** (1994) 6-37
3. O'Gorman, L.: Comparing Passwords, Tokens, and Biometrics for User Authentication. *Proceedings of the IEEE* **91** (2003) 2019 - 2020
4. Secret Societies. Microsoft® Encarta® Online Encyclopedia (2006)
5. Adams, A., Sasse, M.A.: Users Are Not The Enemy. *Communications of the ACM*, Vol. 42 (1999) 40-46
6. Summers, W.C., Bosworth, E.: Password Policy: The Good, The Bad, and The Ugly *Proceedings of The Winter International Symposium on Information and Communication Technologies Cancun, Mexico* (2004) 1-6
7. Xia, H.: Hardening Web browsers against man-in-the-middle and eavesdropping attacks *Proceedings of the 14th international conference on World Wide Web Chiba, Japan* (2005) 489-498
8. Pinkas, B., Sander, T.: Securing passwords against dictionary attacks *Proceedings of the 9th ACM conference on Computer and communications security Washington, DC, USA* (2002) 161-170
9. tcpdump. <http://www.tcpdump.org/>
10. wireshark. <http://www.wireshark.org/>

11. airsnarf. <http://airsnarf.shmoo.com/>
12. John the Ripper. <http://www.openwall.com/john/>
13. RainbowCrack. <http://www.antsight.com/zsl/rainbowcrack/>
14. Madsen, P., Koga, Y., Takahashi, K.: Federated identity management for protecting users from ID theft Proceedings of the 2005 workshop on Digital identity management Fairfax, VA, USA (2005) 77-83
15. Berghel, H.: Phishing mongers and posers Communications of the ACM, Vol. 49 (2006) 21-25
16. Jakobsson, M., Ratkiewicz, J.: Designing ethical phishing experiments: a study of (ROT13) rOnl query features. Proceedings of the 15th international conference on World Wide Web Edinburgh, Scotland (2006) 513-522
17. Litan, A.: Phishing Attack Victims Likely Targets for Identity Theft. (2004)
18. Gouda, M.G., Liu, A.X., Leung, L.M., Alam, M.A.: Single Password, Multiple Accounts. Proceedings of 3rd Applied Cryptography and Network Security Conference (industry track), New York City, New York (2005)
19. Luo, H., Henry, P.: A common password method for protection of multiple accounts. 14th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, Vol. 3 (2003) 2749 - 2754
20. Gaw, S., Felten, E.W.: Password management strategies for online accounts. Proceedings of the second symposium on Usable privacy and security ACM Press, Pittsburgh, Pennsylvania (2006) 44-55
21. Ives, B., Walsh, K.R., Schneider, H.: The Domino Effect of Password Reuse Communications of the ACM **47** (2004) 75-78
22. Microsoft: Windows Live ID.
<https://accountservices.passport.net/ppnetworkhome.srf?lc=1033>
23. The Liberty Alliance. <http://www.projectliberty.org/>
24. Slemko, M.:Microsoft Passport to Trouble. (2001)
25. Jøsang, A., Pope, S.: User Centric Identity Management. Proceedings of AusCERT 2005, Australia (2005)
26. CentralNic: Password Clues. <http://www.centralnic.com/news/research>
27. Brown, A.S., Bracken, E., Zoccoli, S., Douglas, K.: Generating and Remembering Passwords. Applied Cognitive Psychology **18** (2004) 641-651
28. Riley, S.: Password Security: What Users Know and What They Actually Do. Usability News, Vol. 2006. Software Usability Research Laboratory, Department of Psychology, Wichita State University, Wichita (2006)
29. The University of Auckland Human Participants Ethics Committee Guidelines 2003. (2003)
30. Cohen, R.: Hume's Moral Philosophy. The Stanford Encyclopedia of Philosophy, Stanford University (2004). <http://plato.stanford.edu/entries/hume-moral/>
31. Bronowski, J.: Science and Human Values. Harper & Row, New York (1965)
32. Coney, S. (ed.): Unfinished business: what happened to the Cartwright Report? Women's Health Action, Auckland (1993)

Augmented Memory for Conference Attendees

Andrea Schweer and Annika Hinze

Department of Computer Science
The University of Waikato
Private Bag 3105
Hamilton 3240, New Zealand
{schweer,hinze}@cs.waikato.ac.nz

Abstract. Human memory at its best can perform astonishing feats – the tiniest snippet of information can trigger whole chains of associations, ending at an item long-believed forgotten. While modern information systems excel at systematic manipulation of structured or semi-structured information or even vast repositories of unstructured textual information, they are still far from these capabilities.

Unfortunately, human memory is also prone to failure. Thus, a personal information system that augments human memory through suitable means to store and access information could have huge benefits.

In this paper, we introduce a problem domain for research on such a system: researchers who attend academic conferences and need to remember names, affiliations and research interests of fellow conference attendees and many other types of information. Academic interest in wearable, context-aware systems for information display and capture has strongly increased in the last few years. We present an overview of existing systems and point out unresolved issues. We show that resolving these issues will not only make it possible to build improved augmented memory systems but will also contribute to research in a wider context. Finally, we outline our agenda for research in this area.

Keywords: Context-Awareness, Personal Information Management, Mobile Information Systems, Wearable Computing

1 Introduction

The idea of using computers to augment a person’s memory capabilities – to increase the amount of information that can be kept in one’s memory and recalled at will – is almost as old as computers themselves.

In 1945, Vannevar Bush envisioned how technology could support researchers in managing their documents, notes and other information [1]. One component of his vision was a storage device built into a researcher’s desk that would hold all documents encountered by the researcher. The user of this system would be able to easily add new documents and retrieve those he or she has already seen. In addition to that, the system would allow for the creation of connections between documents. This part of Bush’s vision has partially come true in modern hypertext systems.

Another component of Bush’s vision was a wearable device, wirelessly connected to the main system. This device would record photographs, voice comments and timestamps while the researcher is working in the field or in the laboratory. Today, it is possible to build such a device at reasonable costs. However, there are still many open issues that need to be resolved to make the device valuable for its users.

This paper outlines research on some of these issues and shows how research on such a device can contribute to more general research areas in Computer Science.

Section 2 introduces the problem domain we are concentrating on in this paper: researchers who visit academic conferences. Section 3 first gives an overview of related work and then points out a number of issues that are still unresolved. Section 4 describes how we intend to address these issues and places this research in a wider context. This includes background information about our project, our research questions, the current status and future steps. Section 5 summarises the contributions of this paper.

2 Attending an Academic Conference

This section describes an application domain for augmented memory systems: supporting researchers who attend academic conferences. The focus lies on the activities that occur during the conference (Sect. 2.1), on the kinds of information that are typically acquired and recalled during or after participating in a conference (Sect. 2.2) and on existing, low-tech memory aids that might be used to improve recall (Sect. 2.3).

2.1 Activities

When a researcher attends an academic conference, this typically includes most of these activities:

- travelling to and from the conference city.
- commuting between the hotel and the conference venue.
- checking in at the conference reception.
- meeting other conference attendees – some of them for the first time.
- attending presentations, demonstrations, keynotes, panel discussions, poster sessions and other events scheduled in the conference programme – alone or as part of a group.
- talking to other conference attendees about a wide range of topics, for example discussions about professional and social events at the conference, research ideas, other professional topics, plans for the evening, travel advice for the conference city and its surroundings, or other personal topics.
- taking part in excursions or social events with other conference attendees.
- exploring the conference city, alone or with other conference attendees.

2.2 Information

After or during the conference, the researcher might want to remember certain information related to his or her experiences at the conference. Here are some example questions that the researcher might ask him- or herself:

- At which place/time/event did I meet this person?
- Which topics have I talked about with this person (at all/last time we met/at a given event/...)?
- Who was it I talked to (about a given topic/at a given event)?
- What happened in February 2006?
- What did I do in Paris?
- Who did I tell about this place/person/conversation?
- Who was it a given colleague introduced me to at a given event?
- At the conference lunch on Thursday of this given conference, there was someone sitting at my table, two seats to my right. Who was that?

From these example questions, a number of information types can be identified (see Fig. 1). Primary entities in these questions are *persons*, *places* and *time intervals*. *Events* are combinations of one or more places, one or more time intervals and a sense of purpose: “While I was attending NZ CSRSC 2007” is not the same as “While I was in Hamilton from 10 until 13 April 2007”. Persons have a wide range of possible attributes – from their name and contact details over their research interests to the colour of the clothes they wore at a specific time/event. *Conversations* are links between two or more persons; they occur at a specific time interval and at a specific (set of) place(s). Each conversation has one or more *topics* – which may in turn be related to persons, places, time intervals, events or conversations. Places, time intervals, events and conversations can be seen as hierarchical. For example, the conference opening is part of the first day of NZ CSRSC 2007 which is part of the whole conference.

All information types, as well as their attributes or other information associated with them, can be used for the subject and for the object of a question.

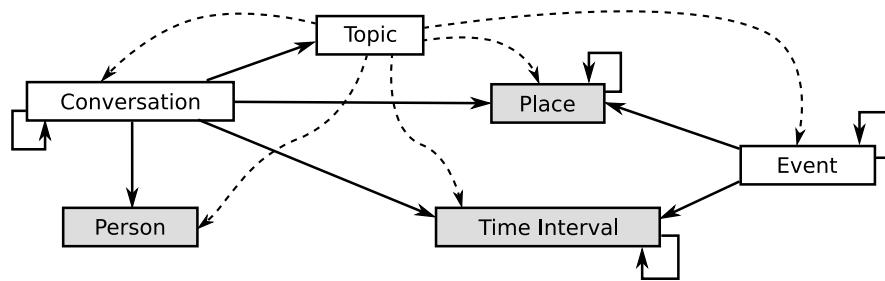


Fig. 1. Information types and their relationships. Primary types are highlighted in grey. Solid arrows stand for the possibility of has-a relationships between entities of the source and target types, dashed arrows for the possibility of is-a relationships. Attributes and relationship cardinalities have been omitted.

2.3 Low-Tech Memory Aids

Without the use of dedicated hardware and software, questions like those presented in Sect. 2.2 can only be answered by referring to one's memory or, in case that fails, by

- looking through the printed or electronic conference programme and proceedings to help remember names and affiliations of persons, presentations, the temporal order of events, etc.
- using business cards collected at the conference to remember names and affiliations of persons.
- using handwritten or electronic notes taken while at the conference or just afterwards to recall persons met there, topics that were discussed, etc.
- looking at photographs to remember persons, events and places.

None of these easily allow for associations between the entities. For example, each business card has to be explicitly annotated with the event at which it was received for this information to be associated with the card. Another drawback is that as most of these memory aids are paper-based, they are not easily portable in large amounts.

3 Existing Systems

This section gives an overview of existing memory aid systems and identifies a number of open questions in this area.

In the Forget-me-not project [2], a PDA was used to collect information about its user's activities (e.g., location of the user, encounters with other people, workstation usage) throughout the day. This information was presented to the user as a biographical log to enable the user to remember details about events.

Jimminy [3, 4], also called the Wearable Remembrance Agent, is an extension of the Emacs text-editor running on a wearable computer with chorded keyboard and heads-up display. It is a note-based system that automatically suggests relevant notes based on the text of the currently edited note and also, if available, on the physical context of the wearer.

Dey et al. developed a dedicated system for the conference domain: Their conference assistant [5] is a portable system that relies on a central conference server to provide conference attendees with information about the conference programme, activities of colleagues at the conference and other conference-related information.

All systems described so far belong to the research area of context-aware computing. According to a commonly-used classification, context-aware applications can either passively display context or actively adapt their behaviour according to the user's current context [6], with the second type being the more common by far. Thus, in these applications context is usually not seen as interesting in its own right. Instead, it is mostly used as metadata for the actual information.

In the last few years, several research projects have been started that aim to use multimedia and possibly dedicated context data recorded through wearable devices during most of the wearer's daily activities.

Arguably the most well-known of these projects is MyLifeBits [7]. Its goal is to store all of a person's digital media, from documents on his/her computer to video recordings of everyday experiences.

The iRemember system [8] is a wearable device that records conversations. To help its user when his/her memory fails, the system makes the recorded data available for searching and browsing through automated speech recognition. This system is typical for a whole range of projects that use machine learning techniques to make recorded multimedia data available for later retrieval (compare [9]).

3.1 Open Questions

Even though augmented memory systems are of interest to an increasing number of researchers, there are still many open questions in this area.

Decreasing size and costs for computer parts, especially storage, make it ever more feasible to build small, powerful wearable devices that capture sensor data and make multimedia recordings of a user's experiences. However, the more important part of augmented memory – improved recall of previously encountered information – still remains difficult. Most existing systems use relational databases with relatively simple data types (timestamps, location, textual data). Associations between items are generally established based on proximity of their timestamps. It is an open question whether data models dedicated to the problem domain can lead to more powerful systems. These specialised data models would require the development of suitable storage models, query languages and retrieval mechanisms. This makes augmented memory systems an interesting field for research in information systems.

Storing new and retrieving previously seen information are the only operations supported by most systems. Especially with highly associative data, the user might find it helpful to edit and reorganise stored information as his or her understanding of a field evolves. To the best of our knowledge, this issue has not been addressed in research on augmented memory systems.

Another open question is how to create an augmented memory system that would be acceptable for the average person to use in his/her everyday life. It is still impossible to use a wearable augmented memory system that is completely concealed from everyone but its user. This makes it especially important to consider social norms in the construction of such a system to ensure it can actually be used outside the laboratory.

4 Our Project

The research described in this paper is part of the first author's PhD project as a member of the Information Systems and Databases Group at the Computer

Science Department, University of Waikato, New Zealand. The second author is the chief supervisor of this PhD project.

This section places the previous sections in the wider context of the whole project. We then introduce our research questions and indicate the project's current status and the next steps.

Our project explores new challenges and opportunities for information systems research – information modelling, storage models, query languages and retrieval methods – that arise in mobile, context-aware systems and specifically in augmented memory systems. Thus, we aim to address the first open question pointed out in Sect. 3.1. In augmented memory systems, context can be both metadata for documents and data in its own right. This makes them more interesting and more challenging than context-aware systems that only display context or adapt their behaviour according to the user's context, and also than systems which treat context merely like any other type of metadata.

There are several reasons why we have chosen to initially focus on the conference domain as described in Sect. 2. Firstly, it is of a semi-structured nature with a limited number of primary information entities, see Sect. 2.2. The conference setting also offers a wide range of context and information types (compare [5]), which increases its usefulness as a testbed. We are confident that the results gained from the conference setting will be transferable to other problem domains. Lastly, our choice of problem domain leads to a good availability of domain experts and possible test users.

4.1 Research Questions

We have refined the goal of our project into the following research questions:

How to Model the Problem Domain. The first step in the development of an information system is to determine which data will be dealt with. This includes the identification of information entities, their attributes and their relationships. We have presented a first cut analysis of the conference domain in Sect. 2.2.

The questions given in Sect. 2.2 can be used to validate the domain model, following the use of competency questions in ontology design [10]: If the list of questions is complete in regards to the intended use of the domain model, the domain model is valid if it can be used to answer all questions. This requires that the list of questions, in turn, has been verified in terms of correctness and completeness.

How to Acquire Context and Other Information. The primary activity of the envisioned users of our augmented memory system is something else than using the system. In the conference setting, the users' focus is – and should remain on – the activities described in Sect. 2.1. To minimise interruptions, our device has to automate the capture of context information as much as possible.

The system should also allow its user to add annotations to captured information. Automatic transcription of audio comments is still error-prone and requires

computational resources often not found on wearable devices. Also, users in the conference setting might be reluctant to record audio comments when they can be overheard. For this reason, it is likely that the system will have to allow the user to input written annotations.

How to Store Data. Once the domain model is complete, we will investigate which systems and mechanisms can be used to store the information described by the model. We will evaluate “classical” database paradigms (e.g. relational and object-oriented approaches) as well as recent developments, for example for RDF data.

How to Query and Retrieve Data. Closely tied to the storage system are the means to access the stored information. Classical strategies for information access are searching and browsing – the first allowing the user to find items that match the supplied criteria, the second to navigate along predefined structures (for example, classification hierarchies). We will decide on how best to support the information needs of the users of our system. Given the associative nature of the human memory, one important access mechanism is likely to be navigation between items that are related in some way.

How to Design the Overall System. Most of the existing systems described in Sect. 3 use a centralised system architecture: While information is captured on small, mobile devices, it is eventually stored on one central, more powerful computer. We will investigate whether this also is a suitable architecture for our system or, if not, find one that is.

How to Improve the User Experience. There are a number of smaller issues not central to our project that we will still have to take into account while developing an augmented memory system. Examples for these are suitable user interfaces for information input, display and manipulation; privacy issues; and sharing of information between two or more users of our system.

4.2 Current Status and Next Steps

The project was started in September 2006. The first step in our project was to find a suitable problem domain for our research. This step has been completed: We have decided to concentrate on the domain of researchers who attend academic conferences, for reasons outlined at the start of this section. As a second application domain (not reported on in this paper), we will consider travel diaries that record tourists experiences.

The next steps are to further review the related work and to refine the data model. We will then make the final decision about the project’s main focus and start addressing the research questions given in Sect. 4.1. Completion of the project is expected for the second half of 2009.

5 Summary

In this paper, we have motivated research on augmented memory systems. We have defined the conference domain of supporting attendees of academic conferences, presented a first requirements analysis and argued why this domain is suitable for further research. Based on an overview of related work, we have identified several unresolved issues in research on augmented memory systems. We have then introduced our project that has set out to address some of these issues while contributing to more general research topics in information systems.

References

1. Bush, V.: As we may think. *The Atlantic Monthly* **176**(1) (July 1945) 101–108
2. Lamming, M., Flynn, M.: “Forget-me-not”: Intimate computing in support of human memory. In: Proceedings of FRIEND21, 1994 International Symposium on Next Generation Human Interface. (February 1994)
3. Rhodes, B.J.: The wearable remembrance agent: a system for augmented memory. In: ISWC 1997: Proceedings of the First International Symposium on Wearable Computers, Los Alamitos, CA, USA, IEEE Computer Society (1997) 123–128
4. Rhodes, B.: Using physical context for just-in-time information retrieval. *IEEE Transactions on Computers* **52**(8) (August 2003) 1011–1014
5. Dey, A.K., Salber, D., Abowd, G.D., Futakawa, M.: The conference assistant: Combining context-awareness with wearable computing. In: ISWC 1999: Proceedings of the Third International Symposium on Wearable Computers, Los Alamitos, CA, USA, IEEE Computer Society (1999) 21–28
6. Abowd, G.D., Dey, A.K., Brown, P.J., Davies, N., Smith, M., Steggles, P.: Towards a better understanding of context and context-awareness. In Gellersen, H.W., ed.: HUC’99: Proceedings of the First International Symposium on Handheld and Ubiquitous Computing. Volume 1707 of Lecture Notes in Computer Science., Springer (1999) 304–307
7. Gemmell, J., Bell, G., Lueder, R.: MyLifeBits: a personal database for everything. *Communications of the ACM* **49**(1) (2006) 88–95
8. Vemuri, S., Schmandt, C., Bender, W.: iRemember: a personal, long-term memory prosthesis. In: CARPE ’06: Proceedings of the 3rd ACM workshop on Continuous archival and retrieval of personal experiences, New York, NY, USA, ACM Press (2006) 65–74
9. Gemmell, J., Sundaram, H.: Guest editors’ introduction: Capture, archival, and retrieval of personal experience. *IEEE MultiMedia* **13**(4) (2006) 18–19
10. Noy, N.F., McGuinness, D.L.: Ontology development 101: A guide to creating your first ontology. Technical Report KSL-01-05, Stanford Knowledge Systems Laboratory (March 2001) Also published as Stanford Medical Informatics Technical Report SMI-2001-0880.

HoneyC - The Low-Interaction Client Honeypot

Christian Seifert, Ian Welch, Peter Komisarczuk

School of Mathematics, Statistics & Computer Science - Te Kura Tatau
Victoria University of Wellington - Te Whare Wānanga o te Ūpoko o te Ika a Māui
P. O. Box 600, Wellington 6140, New Zealand
`{cseifert, ian.welch, peter.komisarczuk}@mcs.vuw.ac.nz`

HoneyC - The Low-Interaction Client Honeypot

Abstract. Client honeypots crawl the Internet to find and identify servers that exploit client-side vulnerabilities. Traditionally, these servers are identified by the client honeypot monitoring state changes that result from a server interaction. These, so called high-interaction, client honeypots are slow and expensive to operate because they require an entire operating system to be hosted. We have developed a component-based low-interaction client honeypot that emulates only the essential features of our target clients and that applies signature matching to allow fast static analysis of server responses. Performance measurements of a prototype implementation targeting clients using the HTTP 1.1 protocol indicate that low-interaction client honeypots are faster and cheaper than high-interaction client honeypots. The difference in false negatives suggests that these technologies may be complementary rather than competitive in nature.

Keywords: Security, Client Honeypots, Intrusion Detection

1 Introduction

A honeypot is a security device that is designed to lure malicious activity to itself. Capturing such malicious activity allows for studying it to understand the operations and motivation of attackers, and subsequently helps to better secure computers and networks. A honeypot does not have any production value. "It's a security resource whose value lies in being probed, attacked, or compromised" [1]. Because it does not have any production value, any new activities or network traffic that comes from the honeypot indicates that it has been successfully compromised. As such, a compromise is easy to detect on honeypots. False positives, as commonly found on traditional intrusion detection systems, do not exist on honeypots.

Honeypots' origins can be traced far back to military concepts and usage, but first appeared in the area of computer security in the 1980s. Stoll describes the hunt of a hacker in 1986 [2]. In order to monitor the intruder on a live system, Stoll and his colleagues provided "bait", fake military reports, to lure the attacker into a particular area of their system. While this was not the honeypot that we

know today, it was the first attempt of "catching flies with honey". The initial honeypot that made use of a simulated environment was described by Cheswick in his account of tracking the Dutch hacker Berferd in 1991 [3]. In the late 90s, attempts to lure and observe attackers moved to the mainstream with the introduction of various tools [4, 5] and commercial products [6, 7].

In our taxonomy of honeypots [8], we classified the majority of these systems according to the taxonomy's developed classification scheme. The main class identified of honeypots was the interaction level. Possible values of the interaction level are high and low. The high-interaction level denotes that the honeypot system allows for full functional interaction. An example of such a honeypot is the Honeynet [5]. A low-interaction level signifies that the functionality is limited, for example, by using emulated services. This strategy is followed by Honeyd [9].

Pouget et al compared the interaction levels of honeypots [10] and concluded they are complementary in nature and allow for more accuracy, depending on the circumstances of deployment and goals of data collection. For example, it might be unnecessary to deploy a high-interaction honeypot on a global scale as global data is likely to be similar; low-interaction honeypots are more suited for this situation. On the other hand, low-interaction honeypots are not suited for an in-depth investigation of attackers actions once a honeypot has been successfully compromised. High-interaction honeypots are required to meet these goals as they expose the full functional spectrum of a computer system for the attacker to interact with and therefore allow for collection of the desired data.

Typically when discussing honeypots we reference *server* honeypots, such as the ones mentioned above that expose server services and wait to be attacked. The focus of this paper, however, is a newer technology called *client* honeypots that deals with a different attack vector. First, in section 2, we introduce client honeypots and review existing client honeypot technology. In section 3, we introduce the notion of low-interaction client honeypots and introduce the first known implementation of such a tool, called HoneyC.

2 Traditional Client Honeypots

Client honeypots are necessary because they are able to detect client side attacks, an attack vector that server honeypots are unable to detect. Client side attacks are assaults of clients that originate from malicious servers. This could be a seemingly harmless visit to a website with a browser. As part of a server's response to a client request, the malicious website might serve code that is targeted at exploiting a vulnerability of the browser as shown in figure 1. As a result, a mere visit to the website might leave a machine exploited with malware. Client honeypots are designed to interact with servers and detect the attacks of servers.

The idea of client honeypots was first articulated in June 2004 by honeypot pioneer Lance Spitzner. Fewer than a handful of client honeypots exist today: HoneyClient [11]; Honeymonkey [12]; and the client honeypot of the University of Washington (UW) [13]. These client honeypots focus on malicious web servers,

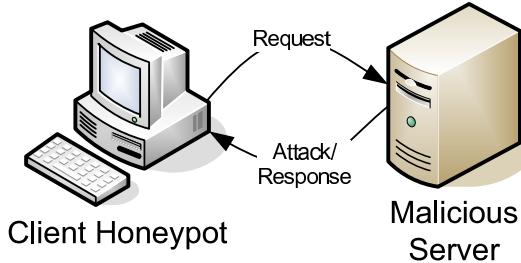


Fig. 1. Client Honeypot Architecture

which they interact with by driving a web browser on the dedicated honeypot system. HoneyClient detects successful attacks by monitoring changes to a list of files, directories, and system configuration after the HoneyClient has interacted with a server. Honeymonkey also detects intrusions by monitoring changes to a list of executable files and registry entries, but Honeymonkey goes a step further by adding monitoring of the child processes to its repertoire to detect client side attacks. The UW client honeypot uses event triggers of file system activity, process creation, registry activity, and browser crashes to identify client side attacks.

All these client honeypots can be classified as high-interaction client honeypots because they make use of a real browser within a real operating system environment and monitor the state of the entire system. With this interaction level come a few disadvantages. First, operation of high-interaction client honeypots is complex and costly as they exclusively occupy an entire host operating system. The inability to detect attacks, so-called false negatives, is another issue encountered by these implementations. While the authors do not provide metrics around detection effectiveness, they do acknowledge that certain events are likely to cause false negatives, such as user interaction to trigger an attack or time bombs that delay an attack. Performance is another shortcoming of these implementations. Monitoring of state, which is employed by all these implementations, is an expensive operation. The performance of Honeymonkey in identification of exploits was about two minutes per URL, whereas the UW client honeypot averaged about 6.3 seconds per URL. HoneyClient has worse performance characteristics in the area of several minutes per URL.

Additional high-interaction client honeypot projects have commenced [14, 15]. These projects acknowledge that web browsers are not the only clients that can be attacked. They also aim at using state-based knowledge to detect attacks, but plan to include a framework that is able to handle different types of clients (P2P, email, media players, etc).

3 Low-interaction Client Honeypots

The concept of low-interaction client honeypot was first identified by us in our taxonomy of honeypots [8]. A low-interaction client honeypot is a client honeypot that uses simulated clients instead of a real system to interact with servers. The subsequent analysis of the response can be based on static analysis, such as signature matching and/or heuristics, which should lead to increased performance and the ability to detect some malicious responses that often elude traditional high-interaction client honeypots, such as time bombs. Since the low-interaction client honeypot makes use of a simulated client, containment of attacks is not a major concern and therefore simplifies deployment of the tool. However, at the same time, a low-interaction client honeypot is likely to miss some unknown exploits that would be identified correctly by a high-interaction client honeypot. The relationship of the low-interaction client honeypot to the high-interaction client honeypot is very similar to the interaction level of server honeypots. Trade-offs exist, but the technologies are likely to complement each other.

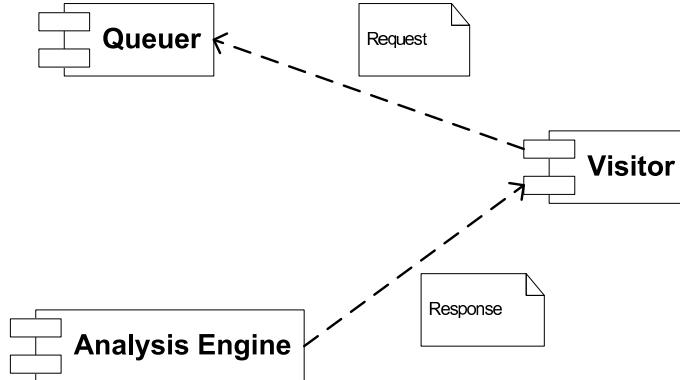


Fig. 2. HoneyC Component Diagram

We identified three tasks that a client honeypot has to fulfill. The client honeypot needs to interact with the server. This usually entails making a request to the server, consuming and processing the response. One portion of the client honeypot needs to create a queue of server requests for the tool to execute. One could employ several algorithms to create such a queue of server requests, such as crawling or search engine integration. In the end, the client honeypot needs to analyze the system or the server response for violation of the system's security policy after the client honeypot has interacted with the server. The first implementation of a low-interaction client honeypot framework, called HoneyC [16], implements these requirements.

HoneyC is a platform independent framework and consists of three components that map to the functional requirements outlined above: Queuer, Visitor,

and Analysis Engine as shown in Figure 2. Each of these components supports pluggable modules to suit specific needs. This is achieved by loosely coupling the components via a command redirection operator (pipe) for passing a serialized representation of the request and response objects. As long as the components agree on the serialized representation of the request and response object, this makes the components implementation independent and interchangeable. This technique is commonly used in Unix for a flexible approach to perform complex tasks. In our instance, this allows us to create a Queue component that generates request objects via integration with a particular search engine API written in Ruby, or to implement a Queue component that crawls a network in C.

In the initial version of HoneyC, we are concentrating on the HTTP 1.1 protocol [17] and have created a serialized representation of HTTP requests and HTTP responses for the components to interact with. We have chosen XML representation [18] as an industry-wide standard for exchanging content in a machine readable format. The HTTP requests that are generated by the Queue component are retrieved by the Visitor component for consumption. The corresponding HTTP responses that are generated by the Visitor are analyzed by the Analysis Engine that obtains the responses from the Visitor once they are available. Since we are working with a command redirection operator to pass the request and response from component to component, the component simply needs to read from standard input and write to standard output. To ensure high performance, threads are used to simultaneously read from standard input and write to standard output. The HoneyC framework automatically links each component to ensure appropriate flow of information.

Implementation components that work with the HTTP objects are provided with the initial version of HoneyC. The Yahoo Search API component is an implementation of a Queue. It is responsible for retrieving URLs by querying the Yahoo Search API with a set of given search parameters. The web browser is a simple implementation of a Visitor component that makes the HTTP request and passes on the response to the Analysis Engine. We implemented a Snort Rules Analysis Engine component, which is able to analyze the response against a set of Snort signatures, a de facto standard of intrusion detection signatures, originating from the lightweight intrusion detection system Snort [19].

4 Preliminary Results of HoneyC

In this section, we present preliminary results of working with HoneyC. We examine performance figures first. With the initially provided HTTP components, we executed HoneyC to query the Yahoo Search API for 10 distinct keywords and subsequently had HoneyC visit 25000 web resources (html pages, JavaScript, images, etc). The responses of the websites were analyzed against a set of 376 Snort rules. We used a notebook with 512MB RAM and a Pentium M 1100MHz processor on a $350 \frac{Kbytes}{sec}$ DSL broadband connection. The duration of HoneyC execution was averaged resulting in a visit and analysis of one web resource in 1.2 seconds, a performance improvement compared to traditional high-interaction

client honeypots. No direct comparison between a high-interaction client honeypot and HoneyC has been undertaken at this time as experiments need to involve identical hardware and visitation schemes, and is left for future work.

We investigate how long it would take to scan the entire Internet with HoneyC. According to a study in January 2005, 11.5 billion publicly indexable web pages exist [20] with an average size estimates per page of approximately 20000 bytes [21]. According to Netcraft, approximately 9800000 web sites existed at that time resulting in about 1173 pages/site on average. Since January 2005, the Internet has grown significantly. Netcraft reports 44 million web sites in October 2006 [22]. Assuming the number of pages per sites has not changed, the Internet consists of approx 52 billion indexable web pages of 945 petabytes (1 thousand terabytes). In order to scan the html pages with HoneyC within a week (604800 seconds), a bandwidth of 1,719,576,720 bytes per second (approx 13.75 GigaBit) is required. Considering that $\frac{\text{bandwidth}}{\text{analysisTime}} \times \frac{\text{sizeofresponse}}{\text{sizeofpage}}$ allows us to determine the number of instances that would be required to process such data, a total of 103174 instances would be needed. As the duration exceeds one week, the bandwidth and number of instances requirements that fall in more feasible ranges.

```
alert tcp any any <> any any (msg: "Web site contains code to modify your home page
(IE)"; reference:url,http://honeyc.sourceforge.net/signatureReferences.php; sid:3400001;
rev:1; classtype:trojan-activity; content:" javascript"; nocase; content:" setHomePage";
nocase; flow:to_client,established; )
```

Fig. 3. Snort Signature JavaScript Example

```
alert tcp any any <> any any (msg: "Sony CD First4Internet XCP uninstallation ActiveX
control identified on web page.";
reference:urlhttp://www.frsirt.com/english/advisories/2005/2454; sid:3400000; rev:1;
classtype:trojan-activity; content:" clsid:80E8743E-8AC5-46F1-96A0-59FA30740C51"; nocase;
flow:to_client,established; )
```

Fig. 4. Snort Signature ActiveX Example

We examined HoneyC's ability to detect malicious servers with the Snort Rule Analysis Engine. Figure 4 and 3 depict two examples of Snort signatures. Snort signatures are composed of pattern matching elements and informational elements that are included in the alert that is being generated once a match has been determined. The first section *tcp any any ij any any* states that any *tcp* traffic should be considered. *flow:to_client,established* filters this traffic further into consideration of traffic that only flows to the client, aka the response of the server. Finally, the content specifies the specific string that should be contained in the response for the rule to fire. Optionally, content matching can be case insensitive. Once the rule fires, an alert is generated that contains the informational elements of the rule, such as the message, the unique identifier SID, the

reference URL with additional information about the alert, and the classification of the alert.

The two signatures depicted support identification of malicious content in a web server response. The first signature identifies a malicious ActiveX control [23] using the class id as an identifier of the ActiveX component, whereas the second signature identifies EMACScript code [24], commonly referred to as JavaScript, that attempts to modify a user’s home page setting of their browser. The latter is a good example of how HoneyC can identify malicious code that is difficult to identify by current high-interaction client honeypots, since user interaction might be required to trigger the JavaScript code. However, depending on the signature quality, false alerts could result and this is a tradeoff to the higher performance obtained with low-interaction client honeypots.

5 Conclusion and Future Work

In this paper, we have introduced a new type of client honeypot, a low-interaction client honeypot, which is designed to address some of the shortcomings of traditional high-interaction client honeypots. We have provided and described an initial implementation of this new technology with HoneyC. While HoneyC is still in its initial stages, preliminary testing has shown some promising results in performance and detection capabilities of this new technology. We acknowledge that signature-based detection as currently implemented by HoneyC is likely to result in some false alerts that remain to be quantified. However, similar to server honeypots, we believe the performance capabilities of low-interaction client honeypots will complement the detection rates of high-interaction client honeypots when used in tandem.

Future work is targeted improving the existing implementation of HoneyC itself, such as performance improvements and creation of additional static detection algorithms. Once implemented, they would allow us to perform some direct performance and detection capability comparisons between the two technologies. Further, there are additional opportunities for future work in the area of Queueer algorithms and Visitor components. We would like to investigate what Queueer algorithms are most effective in the selection of potentially malicious servers. Also, we would like to include different Visitors and compare distributions and trends of malicious web servers vs. other types of servers. If you are interested in joining the project and bringing HoneyC forward, please contact Christian Seifert at cseifert@mcs.vuw.ac.nz.

References

1. Spitzner, L.: *Honeypots: Tracking Hackers*. Addison-Wesley, Boston (2002)
2. Stoll, C.: Stalking the Wily Hacker. *Communications of the ACM* **31**(5) (1988) 484–497
3. Cheswick, B.: An Evening with Berferd in which a cracker is Lured, Endured, and Studied. In: Winter 1992 USENIX Conference, San Francisco, USENIX (1992) 163–174

4. Cohen, F.: Deception Toolkit (1999) Available from <http://all.net/dtk/dtk.html>; accessed on 6 July 2006.
5. : Honeywall CDROM Eyeore (2003) Available from <http://project.honey.net.org/tools/cdrom/eyore/download.html>; accessed 6 July 2006.
6. : CyberCop Sting (1999)
7. : NetFacade (1998) Available from <http://www22.verizon.com/fns/solutions/netsec/>; accessed on 2 June 2006.
8. Seifert, C., Welch, I., Komisarczuk, P.: Taxonomy of Honeypots (2006) Available from <http://www.mcs.vuw.ac.nz/comp/Publications/index-byyear-06.html>; accessed on 14 July 2006.
9. Provos, N.: Honeyd Virtual Honeypot (2003) Available from <http://www.honeyd.org/>; accessed on 6 July 2006.
10. Pouget, F., Holz, T.: A Pointillist Approach for Comparing Honeypots (2005)
11. Wang, K.: HoneyClient, Version 0.1.1 (2005) Available from <http://www.honeyclient.org/>; accessed on 6 July 2006.
12. Wang, Y.M., Beck, D., Jiang, X., Roussev, R., Verbowski, C., Chen, S., King, S.: Automated Web Patrol with Strider HoneyMonkeys: Finding Web Sites That Exploit Browser Vulnerabilities . In: 13th Annual Network and Distributed System Security Symposium, San Diego, Internet Society (2006)
13. Moshchuk, A., Bragin, T., Gribble, S.D., Levy, H.M.: A Crawler-based Study of Spyware on the Web. In: 13th Annual Network and Distributed System Security Symposium, San Diego, The Internet Society (2006)
14. Yuan, B., Holz, T.: Client-Side Honeypots (2006) Available from <http://pi1.informatik.uni-mannheim.de/diplomas/show/27>; accessed on 12 August 2006.
15. Mara, F., Tang, Y., Steenson, R., Seifert, C.: Capture - Honeypot Client (2006) Available from <http://capture-hpc.sourceforge.net/>; accessed on 12 August 2006.
16. Seifert, C.: HoneyC - The Low-Interaction Client Honeypot (2006) Available from <http://honeyc.sourceforge.net>; accessed on 12 August 2006.
17. Fielding, R., Gettys, J., Mogul, J.C., Frystyk, H., Leach, P., Berners-Lee, T.: Hypertext Transfer Protocol – HTTP/1.1 (1999) Available from <http://tools.ietf.org/html/rfc2616>; accessed on 10 August 2006.
18. Consortium, W.W.W.W.: Extensible Markup Language (XML) (1998) Available from <http://www.w3.org/XML/>; accessed on 12 August 2006.
19. Roesch, M.: Snort-Lightweight Intrusion Detection for Networks . In: 13th Large Systems Administration Conference, Seattle, Usenix (1999) 229–238
20. Gulli, A., Signorini, A.: The Indexable Web is more than 11.5 billion pages (2005) Available from <http://www.cs.uiowa.edu/~asignori/web-size/>; accessed on 16 October 2006.
21. Systems, S.o.I.M.: How Much Information? 2003 (2003) Available from <http://www2.sims.berkeley.edu/research/projects/how-much-info-2003/>; accessed on 16 October 2006.
22. Netcraft: October 2006 Web Server Survey (2006) Available from http://news.netcraft.com/archives/2006/10/06/october_2006_web_server_survey.html; accessed on 16 October 2006.
23. Corp, M.: ActiveX Controls (1996) Available from http://msdn.microsoft.com/library/default.asp?url=/workshop/components/activex/activex_node_entry.asp, accessed on 10 August 2006.
24. International, E.C.M.A.: Standard ECMA-262 - ECMAScript Language Specification (1999) Available from <http://www.ecma-international.org/publications/standards/Ecma-262.htm>; accessed on 12 August 2006.

Accent Modification: Approaches and Issues

Jonathan Teutenberg

University of Auckland

Abstract. Three necessary components of a system for the automatic transformation of the accent of speech are introduced. The lexicon that provides a phoneme sequence, functions for transforming the pronunciation of vowels and the construction of prosodic models are discussed. A worked example highlights a number of issues, and possible approaches outlined.

1 Introduction

The acceptability of synthetic speech is conditional on both the quality of synthesis and a perceived rapport with its audience. When a synthesis system is applied in a new location or with a new target audience, the ‘foreignness’ due to accent differences greatly affects this rapport.

With existing systems, the creation of a new voice is a time consuming method for localisation [1]. We propose to adapt existing voices by automatically modifying their accent using minimal time and expert knowledge. In this paper we describe three critical components of an accent modification system: the lexicon that produces a phoneme sequence to synthesise, the pronunciation of the sound produced for each phoneme, and the prosody of the utterance produced.

On our journey through the modification of accent we shall be escorted by the sentence *Pet cats ate Pooh Bear*, initially spoken by a British Received Pronunciation (RP) speaker but which, by the time we reach our references, shall have graduated to a New Zealand English (NZE) accent.

2 Lexicon

The lexicon, also referred to as the pronunciation dictionary, has the enviable task of producing a sequence of phonemes from the text of the input sentence. A phoneme is a distinct unit of speech as perceived by the speaker. English is usually said to have around 42 distinct phonemes, though there are still many who put this number slightly higher or lower. Using a standard RP lexicon our sentence will be rendered as the sequence:

/p e t k a t s e t p uu b e@/

where the ‘/’ indicate a sequence of phonemes.

2.1 Issue: Pronunciation Variation

Even at this early stage a number of problems have arisen. For example, the word *ate* has been transcribed as /e t/, a pronunciation unheard of on the better side of the equator. It is because of these sorts of differences that separate lexicons exist for US and RP synthetic voices, and in general each accent of English requires its own editions to the ‘standard’ lexicon. Ongoing research in the area of speech recognition is attempting to adapt lexicons to individual’s unique pronunciation styles using limited recorded data [2–4], an approach we intend to apply to the adaptation of accented lexicons in the future.

2.2 Issue: Phonetic Differences

A second problem with our phoneme sequence is the final diphthong /e@/, which does not even exist in NZE! As mentioned earlier, a phoneme represents a distinct sound according to the perception of the speaker, and Godzone is blessed with what is known as the air-ear merger [5]. Thus all the diphthongs in *We hear a bare bear in the air with a beer at the fair I fear* are represented by the same phoneme /i@/. While perceptible differences may exist in the pronunciation of these words it has been found that NZE speakers are unable to reliably categorise the diphthongs into two distinct groups, thus eliminating the possibility of a pair of distinct phonemes. In this case the phonemic difference between the accents can be repaired by simply using /i@/ in place of /e@/ wherever it appears, but other accents are not let off so easily. Scottish English for example, can distinguish the words *worn* and *warn*, so there needs to be some system for splitting the RP phoneme into separate classes for every entry in the lexicon.

Comprehensive research into the differences between accents of English have been made[6], and a core accent independent English lexicon produced known only as UNISYN [7]. This core lexicon also provides a set of transformation rules that can produce a number of accent specific lexicons, including such varieties as NZE and Scottish English. Methods of automatically determining the splitting of phonemes based on a large corpus of accented speech is an exciting proposal, but unfortunately lies outside the scope of our work. Using an existing NZE-specific lexicon, our friendly sentence can be transcribed as:

/p e t k a t s ei t p uu b i@/

3 Pronunciation

Armed with an accent-adjusted sequence of phonemes, we allow our synthesis system to create the sounds of speech that make up the output waveform. To our dismay, despite the tweaking of a few phonemes, the output speech still retains the foul taint of an RP speaker. This is due to differences in the phonetic realisation of each phoneme between accents, in particular in the realisations of vowels. The pronunciation of

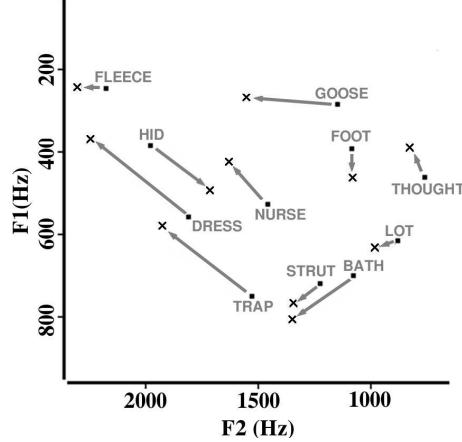


Fig. 1. Difference between RP (squares) and NZE (crosses) vowel spaces.

vowels can be represented on a two dimensional plane, with dimensions based on the positions of the first and second resonant frequencies of the mouth/nasal cavity (the formants). Figure 1 shows the vowels of RP and NZE plotted on the F1/F2 plane, and the distance between them. Our task, should we choose to accept it, is to shift the first and second formants of speech for each vowel so that they move up the grey arrows until they match the locations of the NZE vowels.

3.1 Approach: Baseline System

For a first attempt at correcting the pronunciation of our sentence, we shall not attempt to reach the exact location of the NZE vowels in the F1/F2 plane but merely attempt to reduce the distance to their location. This is achieved by simply replacing each vowel phoneme with the one closest to the NZE target. Thus our phoneme sequence is transformed:

```
Before /p e t k a t s ei t p uu b i@/
After  /p i t k e t s ei t p uu b i@/
```

An earlier experiment has assessed the effectiveness of this crude method of transformation[8]. Providing native NZE speakers with a number of synthetic voices, the transformed voices were placed at 2.7 on a Mean Opinion Score (MOS) scale from 1 (Very British) to 5 (Very NZ). An RP speaker was rated at only 1.9, so the transformed voice was significantly ‘more NZ’.

To place this in perspective, a synthetic voice using recordings of a native NZE speaker to synthesise the phonemes was rated 3.8 on the same scale. This can be considered a perfect transformation of pronunciation, and provides an upper bound for the performance of this component in an accent modification system.

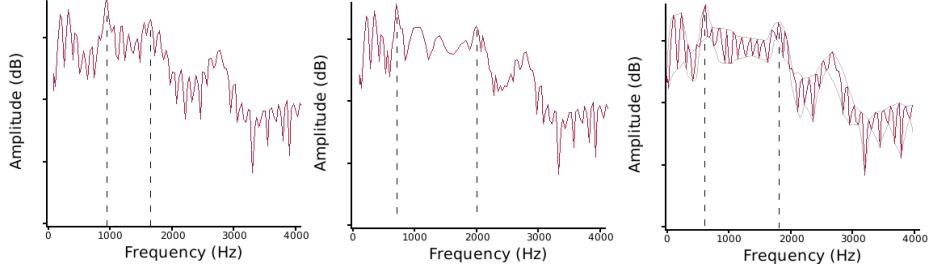


Fig. 2. Frequencies of an RP /ah/ vowel a) in the original speech b) warped towards NZE and c) warped towards NZE retaining local minima and maxima

3.2 Approach: Frequency Warping

If trivial methods can get so far, surely signal processing can do even better. For this, a leaf can be taken from the book of voice conversion. Frequency warping has proved an effective method for transforming acoustic properties of vowels in voice conversion systems [9]. The goal in voice conversion is to stretch or squeeze all the points in the F1/F2 plane so that the overall acoustic space fits that of the target speaker. The intra-speaker differences in acoustic spaces are due to individual's physical characteristics such as vocal tract length and size. The conversion is usually performed by generating a single warping function that is applied across all sounds of speech (phones) of the source speaker, whereas accent modification requires phone-dependent transformations.

Yan et al.[10] adapted the frequency warping method to the shifting of formants and applied this to accent modification. For each phone in the source speech the formants were tracked and a warping function produced that only shifted those frequencies near the first and second formants. Each warping function was applied to a linear predictive (LP) representation of speech, essentially a smoothed approximation of the frequencies of the speech wave. Our experiments in warping the LP representation have degraded the quality of synthetic speech to an unacceptable extent. Instead we apply a warping function to the unsmoothed spectrum obtained by a Fourier transform of the sound wave.

3.3 Issue: Spectral Texture

Figure 2 (a) shows the frequencies of a typical slice of the RP /ah/ vowel in our sentence that needs to be shifted toward NZE. The first two formant frequencies are indicated by a pair of vertical dashed lines. Perceptive readers may have noticed that the spectrum is not a smooth function, but in fact has a number of undifferentiable local maxima at integral multiples of the pitch (the pitch harmonics).

The result of simply warping the frequencies so that the formants lie at the correct frequencies can be seen in Figure 2 (b). Once more, those who are observant will note that the maxima between F1 and F2 are no longer at harmonic frequencies as the spectrum was stretched out

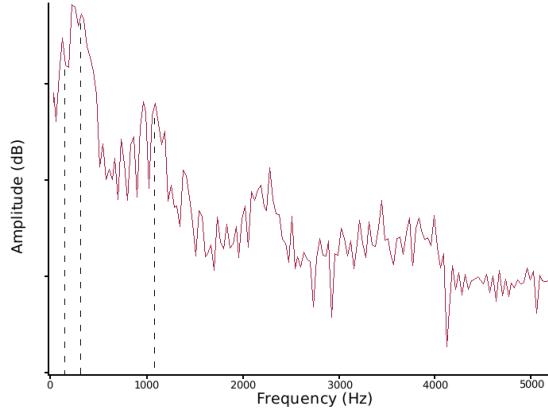


Fig. 3. F0, F1 and F2 of /uu/ vowel in *Pooh*.

between F1 and F2 to accommodate their new positions. This produces a sound containing a fairly reasonable distribution of energy between regions of the spectrum, but the resulting sound can only be described as weak and having little character.

While it can be seen why working with an LP spectrum is so attractive, stupidity is the better part of valour and we shall soldier on for the good of our sentence. An attempt is made to retain the spectral texture by warping a line through the maxima separately from a line through the minima of the spectrum. The spectrum is then reproduced using the original frequencies for local minima and maxima, but using the warped upper and lower bounds to determine their amplitude. An example of this method is shown in Figure 2 (c), with the grey lines showing the warped bounds. The resulting speech still retains artifacts due to slightly unusual distribution of frequencies, but the loss of ‘character’ observed when directly warping the spectrum has been eliminated.

3.4 Issue: Conjoined Formants

A further issue arises when we inspect the *Pooh* /uu/ vowel. Figure 3 shows a spectral slice of this vowel with F0 (essentially the pitch), F1 and F2 marked. Compare this spectrum with that of Figure 2 (a) and it can be seen that the peaks at F0 and F1 have merged to form a single, broad peak. This is due to the low F1 value associated with /uu/ which leaves it only a few hundred hertz off the F0 value. A warping function that increases the separation between these peaks will only spread it across a greater range of frequencies. When this happens there is a marked increase in frequencies around the formants which can be heard as a loud buzzing sound in the resulting speech. What is required is a method that detects these conjoined formants and takes appropriate action to introduce a ‘gap’ when they are separated.

Turn your mind back to Figure 2 (c) once more and consider the gap between the altered F1 and F2 peaks. While some gap exists, a larger

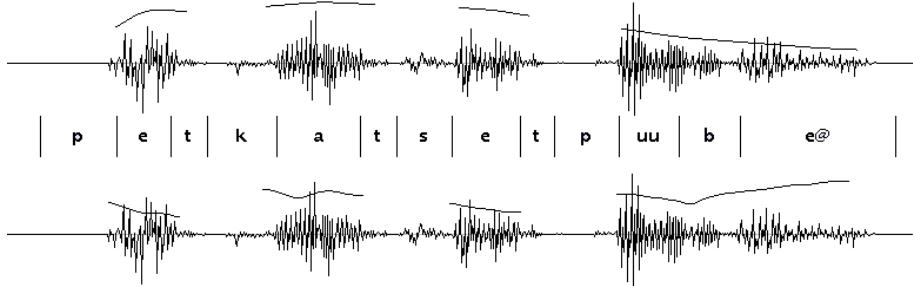


Fig. 4. The pitch contour and phone durations for the RP sentence (top) and a pitch contour from natural NZE speech (bottom)

trough between them would be more natural, and manually adding this dip has been found to remove many of the remaining artifacts in the resulting speech. Part of our current work is looking into simple methods for ensuring natural separations in adjusted formants.

3.5 Approach: Approximate vs Exact Warping

Before the transformation of vowels is left behind, we had best take a brief look at how the warping functions have been determined. Yan et al relied on a formant tracker to obtain exact positions of F1 and F2 in their source speakers. As yet, formant tracking is an inexact science, and any incorrect labelling would result in potentially catastrophic results. Entire frequency bands could be displaced, introducing audible artifacts throughout the spectrum.

Once more we consider a crude, yet more robust approach. Given that we know the accent of our source speaker, reasonable approximations of the locations of formants can be made. Furthermore, as this is sourced from a synthetic voice, we can run an initial poll of various key words and adjust our expectations for this particular speaker. Preliminary tests with hand-drawn warping functions that apply a very broad brush to the spectrum have been run. These functions push the formants and surrounding spectral features toward the target accent. Informal listening tests show no real difference between a broad warping of the spectrum and one that is very specific to the formant peaks.

4 Prosody

A playback of the sentence indicates a rather kiwi-like quality to the voice. It is, however, still missing a certain something, and this something is the intonation and pitch of the speech. Figure 4 shows the current pitch contour and phone durations (referred to as the prosody) of the sentence. There are two possible approaches to NZEifying the sentence. Since we are aware that this sentence is a statement, we can simply alter the pitch

contour so that it rises on the final syllable - a characteristic of NZE that is evident in Figure 4. In general, when creating a transformation for a new accent a number of such characteristic features would need to be discovered and applied when appropriate. The alternative is to build a whole new model and to generate our own pitch contour, essentially replacing this component of the synthesis system.

4.1 Issue: Data Limits

The primary issue with creating models of prosody is the quantity of data available. Models of prosody do not deal directly with the pitch contour, but instead describe the contour on an abstract level such as Tones and Break Indices[11], Rise/Fall/Connect[12] or Fujisaki's model[13]. Labelling data at an abstract level requires a significant investment of expert time for mark up, and to compete with recording whole new voices we must keep this time to a minimum.

While it is important to work with minimal quantities of labelled data, we do have access to near infinite supplies of unlabelled data. Unlabelled data can be found in the form of sentences extracted from suitable websites such as online encyclopaedias, chat rooms, news sites or sites with domain specific content. How to make use of this unlabelled data to complement the labelled data is an important question in the area of prosody modelling.

4.2 Approach: Model Adaptation

A method we are considering that utilises unlabelled data for the generation of prosodic models is a form of active learning that relies on the fact that we have existing models for our source accent. Model generation begins by learning a new prosodic model from a very small corpus of labelled, accented data. The predictions of this model on a large amount of unlabelled data are then compared to the predictions of the existing model for the source accent. Those sentences on which the two models disagree are of interest. One possibility is that the new model has made an error, in which case this item should be labelled against a native speaker and added to the training data. The other possibility is that the model describes a useful deviation from that of the source accent, in which case this data point should be added to the training data and given special significance. This limits queries to a human expert to those instances that discriminate the accents or on which the model is making errors.

5 Evaluation

To our dismay we are short on both space and time to evaluate the success of the transformation of our sentence. There are no concrete examples of this sentence in the target accent to compare against, so only through perceptual tests can we assess its performance.

How the quality of the synthetic voice has held up through the transformations can be asked. Certainly this is important. How strong the accent sounds. Not a bad question. But whether we have fulfilled the larger goal of making the synthetic voice more acceptable to its target audience must be resolved. Under what environment is it possible to accurately assess change in acceptability? What sort of questions can participants be asked that do not bias their answers? Tough questions, and there are more where they came from. Perhaps it would be best if we just gave up now...

References

1. Black, A., Lenzo, K.: Building Voices in the Festival Speech Synthesis System. Retrieved on 11/05 from <http://festvox.org> (2000)
2. Humphries, J., Woodland, P., Pearce, D.: Using Accent-Specific Pronunciation Modelling for Robust Speech Recognition. International Conference on Spoken Language Processing (1996)
3. Tjalve, M., Huckvale, M.: Pronunciation variation modelling using accent features. In: Proceedings of INTERSPEECH. (2005) 1341–1344
4. Bennett, C.L., Black, A.W.: Prediction of pronunciation variations for speech synthesis: A data-driven approach. In: International Conference on Acoustics, Speech, and Signal Processing. (2005)
5. Batterham, M.: The apparent merger of the front centering diphthongs - *ear* and *air* - in new zealand english. In: New Zealand English, Allan Bell and Koenraad Kuiper eds. 111–145
6. Wells, J.: Accents of English. Cambridge University Press (1982)
7. Fitt, S., Isard, S.: Synthesis of Regional English Using a Keyword Lexicon. In: Proceedings of Eurospeech 99. Volume 2. (1999) 823–826
8. Teutenberg, J., Watson, C.: Vowel Quality in Accent Modification. In: Eleventh Australasian International Conference on Speech Science and Technology. (2006)
9. Toda, T.: Voice Conversion Algorithm Based on Gaussian Mixture Model with Dynamic Frequency Warping of STRAIGHT Spectrum. In: Proceedings of International Conference on Acoustics, Speech and Signal Processing. (2001)
10. Yan, Q., Vaseghi, S., Rentzos, D., Ho, C.: Analysis by Synthesis of Acoustic Correlates of British, Australian and American Accents. In: International Conference on Acoustics, Speech and Signal Processing Proceedings. Volume 1. (2004) 637–640
11. Silverman, K., Beckman, M., Pitrelli, J., Ostendorf, M., Wightman, C., Price, P., Pierrehumbert, J., Hirschberg, J.: ToBI: a standard for labeling English prosody. In: Proceedings of International Conference on Spoken Language Processing. Volume 2. (1994) 867–870
12. Taylor, P.: The Rise/Fall/Connection Model of Intonation. Speech Communication (1994)
13. Fujisaki, H., Sudo, H.: Synthesis by rule of prosodic features of connected Japanese. In: Proceedings of International Congress of Acoustics. (1971)

Improving the Stability of Transmission in Mobile Ad Hoc Networks

Yifan Zhang and Peter Komisarczuk

Distributed Systems Research Group
School of Mathematics, Statistics & Computer Science
Victoria University of Wellington
Kelburn Parade, Wellington, New Zealand
yifan.zhang, peter.komisarczuk@mcs.vuw.ac.nz
<http://www.mcs.vuw.ac.nz/research/dsrg/>

Abstract. Protocol development in Mobile Ad hoc NETworks (MANET) has been a hot topic for many years, yet the performance exhibited from recent deployments [1] is still not optimal and not suitable for large scale practical use. Stability of transmission is a critical aspect in mobile ad hoc networks, as by improving the stability of end-to-end transmission, the network could deliver more consistent service and achieve efficiency gains. We propose a protocol architecture which aims to improve the stability of transmission inside MANET using limited additional information based on the location of wireless nodes. The draft architecture and the initial progress in this project are also presented in this paper.

Key words: MANET, Stability, Location, FPGA

1 Introduction

Mobile ad hoc networks are a combination of mobile terminal nodes equipped with wireless network capability that does not rely on additional infrastructure (e.g. base station, access point) to operate. This flexibility makes ad hoc networks suitable for scenarios such as disaster recovery e.g. in an earthquake scenario where the existing infrastructure is knocked out. Nodes in ad hoc networks need to act as routers when they are on the path between distant node pairs that cannot reach each other directly. This greatly increased the complexity and transmission cost for packet transmission in ad hoc networks and raise more challenges compared to cellular wireless networks. MANET nodes need to provide some quality of service to their neighbourhood if we are to enable effective end-to-end transmission, especially for real time traffic such as voice. With over a decade of research in this field, the current ad hoc protocols still have some key problems that prevent them from true success. One critical problem is the reliability of transmission in ad hoc network, which is affected for example by routing stability, node mobility, miss-behaviour of protocols at different layers of the stack and more.

Our research is focused on improving the stability of transmission in MANETs, as by improving stability we could deliver more consistent service for an end-to-end transmission and improve the overall efficiency of the system by reducing the overhead of retransmissions. We will present the draft proposal of a multi-layer architecture that makes use of node location information that should improve the stability of ad hoc transmission. The proposed architecture includes a neighbourhood aware MAC, a location aided routing protocol through transport protocols such as the SCTP (Stream Control Transmission Protocol) with cross-layer functionality extensions. We are also looking into network feedback mechanisms and protection mechanisms to reduce the network costs when certain end-to-end and link connectivity feedback is lost during packet transmission.

This paper is organized as follows: In Section 2, we analyse the MANET stability problem. Then, we in Section 3 we introduce some details of the proposed protocols in our architecture. We present our progress and future plan for our project in Section 4.

2 Problems in Ad Hoc Networks

The experiments conducted by MobileMAN [1] have shown that the instability of transmission is one of the major causes of the performance degradation in their mobile ad hoc environments. In this section we identify some of the causes of the instability in MANET and analyse some possible solutions to reduce this instability.

2.1 Cause of Instability

The root of instability lies in the dynamic nature of the MANET, which are caused by the mobility of nodes and the change in wireless link characteristics. Links between nodes change when nodes move. When nodes move away from each other, the link quality drops and eventually the link breaks when the pair is out of transmission range of each other. As we cannot limit the movement of the nodes inside the network, what we can do is to estimate when the link between a pair of nodes is likely to break. To perform such a prediction inside the network, we need additional information to be transmitted or shared by the nodes.

We are investigating using the location information of the nodes in MANET to predict node connectivity in order to minimise network instability. Location information can be obtained for example from GPS, time of transmission techniques or received signal strength techniques. GPS provides a good self contained modular solution for location information with its accuracy depended on satellite visibility. Time of flight method somehow requires access to a high accuracy clock at sender and receiver, the clock accuracy and triangulation determines the location of the sender. Received signal strength technique are the most ubiquitous but suffers from uncertainty in location as the signal strength varies with many factors. Low overhead is required to transmit location information around

the network, and by having location information we could predict a node's future movement using historical data. This information could then be used at a number of protocol stacks in order to increase network stability as discussed in section 3 (e.g. at the MAC layer and the network layer in the routing protocol).

The second cause of instability can be classified as the over-reaction of protocols to various network events. Wireless ad hoc protocols, in general, can only judge the network status from various feedback mechanisms such as MAC beacon frames, ACK frames and TCP ACK packets. If this critical feedback is lost, the protocol will assume that there is a problem in the network and perform recovery even when the cause is not rooted in the network. The protocol layer in isolation doesn't necessarily have enough information to judge the end-to-end conditions on the loss event that it detects, the provision of messages from other protocol layers could help the protocol to judge the cause of the event and perform more effective actions at the protocol layer. The provision of feedback messages from other layers of the protocol stack has been investigated in a number research projects like the Explicit Link Failure Notification (ELFN) [2] which provide information from network layer to transport layer to help TCP in ad hoc environments.

An alternative way of reducing the effect of instability is to provide protection on the critical transmission components, optimizing network feedback, and reducing loss. We aim to investigate some enhancements to the feedback behaviour of some of the protocols in our system to test this approach. We expect to use multiple feedback which has been investigated by Greensten et. al at [3] or standard feedback mechanisms plus additional piggy back data in our system.

3 Proposed Solutions

More details of our proposed system are presented in this section, including new protocols at different layers, enhancements on existing protocols and effective messaging architecture between protocol layers.

3.1 The Implementation Platform

WAG Platform WAG stands for Wireless Analysis and Generation. It is a wireless protocol experiment platform developed by the WAND [4] research group of Computer Science Department at the University of Waikato. WAG is a complete system with a wireless NIC (Network Interface Card) using FPGA, a Soekris Engineering [5] x86 based compact computer and an embedded Linux OS with drivers, protocol implements and measurement tools. We have used this platform as the initial basis of the protocol development in our project.

The wireless NIC is the major component in the system. It is equipped with a Xilinx FPGA, 802.11b radio and modem, on-board RAM and PCI and serial interface. By using an FPGA, we have the flexibility of implementing any MAC protocol here, not just limited to IEEE 802.11. In addition, we can put functions

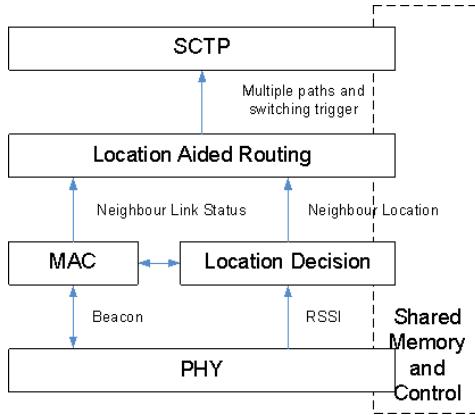


Fig. 1. Controlling Flow of the Proposed System

(e.g. local table management, node separation prediction) inside the FPGA to take advantage of fast calculation and close integration with the MAC protocol.

There are also development tools available which translate the system level design in SDL (Specification and Description Language) into C/C++ codes that could run on FPGA. A simulation framework also exists that allows the use of the C/C++ code directly in the NS2 network simulator.

Cross-layer Architecture A specific header is used to exchange information (frame size, receiver parameters, etc.) between the FPGA and device driver in the current WAG framework. It is 32-byte long and not large enough to carrying neighbourhood information in our project. Two cross-layer messaging methods will be evaluated on the platform before deciding on the architecture. One method is using a shared memory space for these informations which is similar to the architecture proposed in MobileMAN project [6]. Another method is to introduce special types of management packets between device driver and the MAC, which carry these information around the protocol stacks driven by different events.

MADWifi Platform MADWifi or Multi-band Atheros Driver for Wi-fi [7], is a set of Linux device drivers developed for wireless LAN cards using the Atheros chips. The unique feature of this driver is its ability of controlling some of the hardware functionality via HAL (Hardware Abstraction Layer, a binary interface provided by Atheros). This ability makes the MADWifi platform a possible alternative protocol implementation platform for our proposed system. We are still looking at the platform and evaluating this possibility which promise to be more cost effective but not as flexible at the MAC layer.

3.2 Neighbourhood Aware MAC

This MAC protocol is a simplified contention MAC based on similar resolution mechanism that we used in the DCF (Distributed Coordination Function) in the

IEEE 802.11 standard [8]. The novel element in this MAC is the neighbourhood management function. This function is composed of three parts: neighbourhood detection, neighbourhood information management and neighbourhood information exchange.

The neighbourhood of a node is detected via a listening unit attached to the receiving block in the MAC. It processes all frames received by the node and tries to extract as much information from these frames as possible, e.g. source MAC address, source IP address, RSSI of the frame, etc. All this information will then be passed to the neighbour information management unit, which is basically a table management function. We have three tables related to neighbour information inside the MAC, one for neighbourhood link, one for neighbourhood location and one for on-going TCP/UDP flows on each neighbour. The management function is in charge of inserting new entry or update existing entry using the information from the hearing unit, deleting stale entries after they are kept for certain time and responding to queuing request from MAC or other layers. The last unit, is the information exchange unit. It is the most complex part in the neighbourhood function. It controls the exchange of the three tables among a node's one-hop neighbourhood, enabling some important functions in the system. We will use a MAC beacon frame to carry these tables, however, as the size of the tables grows (location table will include some previous location information for mobility prediction), it is not efficient to broadcast these big frames frequently. On the other hand, we need fresh location information carried by the beacon to perform mobility prediction for the neighbourhood. To balance between the overhead caused by the exchange and the requirement of timely update, we propose an adaptation algorithm that controls the interval and content of each beacon. This algorithm takes feedbacks from mobility prediction and table management units to obtain a view of network situation in the neighbourhood area. It then selects different intervals for each table according to the calculation from this feedback, and manages the beacon mechanism accordingly.

An algorithm of judging node's location using different techniques (GPS plus received RSSI of different nodes) will be introduced to the data link layer. We will test several algorithms from [9] and [10] and the system developed at Victoria University [11] [12] to check whether they could operate in our environment. A few nodes with existing knowledge of their location will act as the reference nodes inside the network, and other nodes will calculate their relative location from these reference nodes to obtain a geographical view of the network. Also we will implement the enhancement of contention resolution proposed in the paper by Durvy and Thiran [13] in this MAC to improve its performance.

3.3 Location Aided Routing

Geographical or location based routing protocols use node location information to help the routing process. Based on how location information affects the routing process, we divide location based routing into two sub categories: location routing and location aided routing. Location routing protocols like [14] use only

location of nodes to perform the end-to-end transmission without having an end-to-end path while location aided routing protocols like "dream" [15] use location information together with other metrics (hop count, etc.) to help reduce the overhead or increase the speed in finding new routes. We need the end-to-end view of the path to perform some operations (e.g. switching) thus we need a location aided protocol in our system.

We will use proactive routing in this project for two major reasons: first, we are going to introduce transport layer switching in this project, and proactive routing is more efficient as it should have many routes available at any one time. Second, we need a method to spread the three tables from the neighbourhood function throughout the network to obtain an overall view of the network which then could aid end-to-end routing and transport protocols. This could be done together with the exchange of routing information and saves the effort of developing a separate function. There are two main concerns in this protocol, how to select the route and how to exchange routing information (and the tables) with proper costs.

As discussed in the previous section, we need a long lasting route between source and destination to improve the stability of the transmission between the pair of nodes. This is achieved by selecting the route that has the longest estimated existence time between the end-to-end pair. This end-to-end estimation comes from the calculation of all one-hop link existing time along the path using local prediction values inside the location tables. One problem with this decision algorithm is that it tends to select the links between fixed nodes or nodes with stable links, which might cause bottlenecks in the network. In order to overcome this we decided to add more elements in the routing decision metrics to help satisfy route requirement while maintain certain level of fairness among on-going sections in the network. These elements includes: path collision status, traffic requirement, path length and geographical topology information.

Some of the critical elements identified in several protocols are also obtained from the global picture formed from local tables. How to effectively exchange these tables in the network is the next issue to be determined. Here we propose an algorithm that combines the flooding and remote queuing (push and pull). In the initial stage, nodes will flood the tables to the network to form the overall network picture. Then nodes will try to queue and process tables from other nodes in the neighbourhood which will update the nodes information of that area and prevent its state becoming stale. The queuing consequence is similar to the route discovery process in reactive routing with geographical constraints.

3.4 Other Protocol Stack Aspects

Two specific protocol stack aspects will be discussed in order to complete our introduction to stability in MANETs. These are developments in transport protocols and in cross layer feedback mechanisms. One of the key developments in transport protocols is a message oriented protocol called SCTP which can be used in MANETs in order to switch between streams. Secondly feedback mech-

anisms can be envisaged between protocol layers in order to reduce protocol sensitivity.

SCTP The Stream Control Transmission Protocol (SCTP) [16] is a reliable transport protocol originally designed for telephony signalling. It is a connection oriented protocol much like TCP and also borrowing some beneficial features from UDP. The most interesting feature of this protocol to us is multi-homing, which allows the end-to-end pair to establish multiple paths using different interfaces and switch to alternative path if one path fails. SCTP will be introduced in transport layer in our system to provide protection on end-to-end transmission via switching between two alternative paths, the switching would take effect on a link outage event, or quality impairment on the primary path.

Some modifications will be done on SCTP to make it work better in our system. SCTP will try to switch between disjoint paths on the same wireless interface now and the status of selected paths will be maintained by routing protocol instead of the heartbeat packets in the SCTP proposal.

Also some enhancements for wireless environment will also be implemented, e.g. the Explicit Link Failure Notification (ELFN) for TCP could also be implemented, as SCTP uses identical congestion control method as TCP and the enhancement proposed by Fracchia et. al at [17].

Feedback Mechanism To reduce the over reaction of the protocol, we can introduce feedback information from different protocols (such as the ELFN mentioned before) which could help to gain a better understanding of what is actually happening in the network. We can also tweak the parameters of the protocol as defined by the MobileMAN project [1] to reach a balance point of sensitivity.

We will also look into the protection methods for the critical feedbacks in order to minimize the cost of loss of these feedbacks. One possible solution is the packet salvation proposed by Yu, Shin and Song [18], where neighbourhood nodes hear the feedback could help to reproduce the feedback if it is lost at the destination. One other way is using redundant/multiple packets for critical feedbacks like in [3], and we can piggyback these feedbacks to data or other packets to reduce the chance of collision.

4 Current Progress and Future Plans

We are working on the detail proposal of each protocol inside the system and also collaborating with WAND workgroup at the University of Waikato on the WAG hardware platform. We are also investigating the possibility of the alternative implementation platform of MADWifi now. We plan to implement the multi-layer architecture in the NS2 network simulator and verify its functionality. Then this system will be implemented using the WAG platform or MADWifi and tested in a real world test-bed.

References

1. ANASTASI, G., ANCILLOTTI, E., BORGIA, E., BRUNO, R., ET AL. Mobileman technical evaluation. Deliverable D16. Tech. rep., MOBILEMAN, (2005).
2. HOLLAND, G., AND VAIDYA, N. Analysis of tcp performance over mobile ad hoc networks. In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking* (1999), pp. 219–230.
3. FU, Z., GREENSTEIN, B., ET AL. Design and implementation of a tcp-friendly transport protocol for ad hoc wireless networks. In *ICNP '02: Proceedings of the 10th IEEE International Conference on Network Protocols* (2002).
4. WAND network research group. <http://wand.cs.waikato.ac.nz>, (2006).
5. Soekris Engineering. <http://www.soekris.com/>, (2006).
6. BORGIA, E., BRUNO, R., ET AL. Mobileman architecture, protocols and services, deliverable d10. Tech. rep., MOBILEMAN, (2004).
7. Multi-band Atheros Driver for Wi-fi. <http://madwifi.org>, (2006).
8. IEEE COMPUTER SOCIETY LAN MAN STANDARDS COMMITTEE. Wireless LAN Medium Access Protocol (MAC) and Physical Layer (PHY) Specification. IEEE Std 802.11-1999. The Institute of Electrical and Electronics Engineers, New York, NY,(1999).
9. CHENG, Y.-C., CHAWATHE, Y., LAMARCA, A., AND KRUMM, J. Accuracy characterization for metropolitan-scalewi-fi localization. Tech. rep., Intel. Research, (2005).
10. KOTANEN, A., HANNIKAINEN, M., ET AL. Positioning with ieee 802.11b wireless lan. In *PIMRC 2003. 14th IEEE Proceedings on Personal, Indoor and Mobile Radio Communications* (2003), vol. 3, pp. 2218–2222.
11. WIERENGA, J., KOMISARCZUK, P. SIMPLE - Developing a LBS positioning solution. In *MUM 2005. 4th International Conference on Mobile Ubiquitous Multimedia* (2005).
12. YANG, X.-P. Wireless Location Determination using Particle Filters. Master thesis, School of Mathematics, Statistics and Computer Science, Victoria University of Wellington, (2007).
13. DURVY, M., AND THIRAN, P. Understanding the gap between the ieee 802.11 protocol performance and the theoretical limits. In *Third Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks* (2006).
14. BASAGNI, S., CHLAMTAC, I., SYROTIUK, V. R., AND WOODWARD, B. A. A distance routing effect algorithm for mobility (dream). In *ACM/IEEE Mobicom* (1998), pp. 76–84.
15. Ko, Y.-B., AND VAIDYA, N. H. Location-aided routing (lar) in mobile ad hoc networks. In *ACM/IEEE Mobicom* (1998), pp. 66–75.
16. R. STEWART., Q. XIE., ET AL. Stream Control Transmission Protocol RFC 2960, (2000).
17. FRACCIA, R., CASETTI, C., CHIASSERINI, C.-F., AND MEO, M. A wise extension of sctp for wireless networks. In *IEEE International Conference on Communications* (2005), vol. 3, pp. 1448–1453.
18. YU, C., SHIN, K. G., AND SONG, L. Link-layer salvaging for making routing progress in mobile ad hoc networks. In *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking* (2005)

***Ad Hoc* Visual Documentation**

Isaac Freeman

University of Canterbury

Abstract. This paper outlines proposed research into software documentation created by users in isolation from the original designer. I discuss research questions I intend to investigate, and a possible model for producing such *ad hoc* documentation, with elements drawn from minimal manuals, programming by example, and visual documentation.

1 Introduction

Documentation for software applications has traditionally been created by designers, or technical writers working closely with them, as separate manuals for each tool. However, real users work with multiple applications, in their own environments, to achieve their own goals. The full information required to make documentation useful, relevant and efficient for each user is not possessed by the designer.

Traditional manuals attempt to capture the designer's mental model of the system, and recent research [1] has focused on making writing manuals synonymous with designing the system. The movement towards 'minimal manuals' has attempted to bring more user experience into the design of documentation through iterative testing and deliberate efforts to design documentation with the user in mind, but the assumption has remained that documentation is generated primarily by the designer..

An alternative approach would capture the knowledge of experienced users within their normal working environment. A good *ad hoc* documentation system should place minimal demand on an author's time, capture examples of real work, and structure knowledge so that novice users can develop an independent understanding of the system.

This paper discusses related work, and presents directions for research along with an outline of a proposed system for producing *ad hoc* documentation.

2 Related Work

Early technical documentation commonly attempted to comprehensively describe each feature of the system separately. Such *systems-oriented*

manuals are suitable for reference, but they are usually opaque to a novice user trying to achieve a particular task. In the 1980s, following studies of users in realistic environments, John M. Carroll noted [2] that users seldom read very far into a manual before formulating their own goals and attempting to use the system to achieve them. He proposed a *minimal manual* to accommodate this behaviour by a variety of methods — the following list of techniques is due to R. John Brockmann [3]

- Cut secondary features of manuals and on-line documents (overviews, introductions, summaries, etc.)
- Focus on what readers need to know in order to immediately apply it to productive work.
- Test repeatedly during design
- Make it easy for the reader of a page to coordinate the documentation with the screen information via pictures of screens or other graphics.
- Use what the readers already know by continuously linking new information to it
- Encourage active exploration of a system via intentionally incomplete information.

Stripping secondary features often reduced the size of documentation by 75%, while focusing it more clearly on realistic tasks and how to complete them. Users were encouraged to develop their own conceptual understanding independent of the designer's model of the system, and Carroll found dramatic improvements in learning time and error handling.

The content of instructional documentation may be divided into concrete procedural ("how-to") information, describing the sequence of actions a user must perform to achieve a goal, and abstract conceptual ("why") information, providing deeper understanding. Research by Gong [4] focused on procedural learning as a defining element of minimal manuals. He found significant improvement in learning performance with manuals constructed using a GOMS model to decompose user goals into subgoals, eventually reaching a keystroke-level procedures for any given task.

More recently, Bergman, Castelli, Lau and Oblinger [5] developed DocWizards, a system that records an expert user's interactions with the Eclipse development environment, capturing the sequence of widgets used to complete a task. DocWizards can compare existing recordings with the actions of users, noting alternative paths by which users might achieve the same goals, and allows input to be parameterised.

Modern documentation usually contains graphical elements such as screen shots in a subsidiary role to text. The transition from text-based

to graphical user interfaces as the primary form of interaction with computers has not been duplicated by documentation. Structured visual languages such as UML have proved suitable for some abstract conceptual explanations, but they generally do not stand alone as full explanations of a system, nor can they be easily understood without training. Graphical user interfaces represent concepts with concrete images on the screen, and a procedure for completing a task in a graphical user interface can readily be decomposed into a sequence of atomic actions in the user interface, each of which has a clear visual expression as an object in a GUI.

Huang, Lu and Twidale have demonstrated a visual approach to procedural documentation called Graphstract [6], which eschews text entirely in favour of small clips of GUI widgets arranged in a vertical sequence to indicate the order in which they should be performed. Experiments suggested modest improvements in performance of users following instructions presented in Graphstract sequences over conventional text-based help. The Graphstract approach suffers from a lack of clear context for actions: locating the widgets in their respective windows on screen requires learning a system of indentation.

3 Proposed Research

I am in the early stages of research: literature review, informal surveying and field study to identify requirements for *ad hoc* documentation, and tools that support its production. Controlled studies will then compare learning performance of users supported by different forms of *ad hoc* and traditional documentation, measured by time taken to complete new tasks, retention of learned knowledge, and subjective responses. This will lead to a set of guidelines for the development and use of *ad hoc* documentation tools. Ultimately, I intend to demonstrate an application for capturing and presenting *ad hoc* documentation, built around a program which queries the OS windowing system as the user works, recording relevant data for each widget the user interacts with. An author will be able to start and stop recording, and add a brief description to each recorded sequence.

Figures 1a and 1b show example visual representations of task sequences. *Action Panels* each contain a closely-cropped screenshot of a single widget being activated by the mouse pointer, and are overlaid on *Context Panels* showing the screen, menu or window in which the actions are to be performed.

The application would also apply a variety of heuristics to compare sequences of user actions to determine suitable higher-level documentation structure. Figure 1c shows an example sequence in which actions common to both of the previous trails have been collapsed into descriptions, implementing a suggestion made by Carroll, that learning is promoted when users are encouraged to complete hidden parts of a task. I intend to explore further automated techniques leading users from procedural to conceptual understanding of applications they use in real working environments.

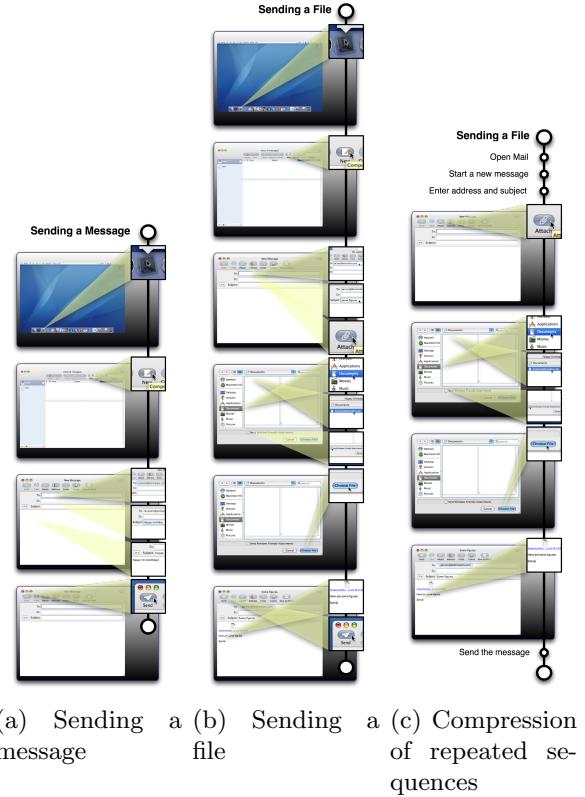


Fig. 1: Example trails

References

1. Thimbleby, H.: Combining systems and manuals. In: Human-Computer Interaction'93. (1993)
2. Carroll, J.M.: The Nurnberg Funnel: designing minimalist instruction for practical computer skill. MIT Press (1990)
3. Brockmann, R.J.: The why, where and how of minimalism. In: Proceedings of the 8th Annual international Conference on Systems Documentation, New York, NY, ACM Press (1990) 111–119
4. Gong, R., Elkerton, J.: Designing minimal documentation using a goms model: A usability evaluation of an engineering approach. In: CHI'90. (1990)
5. Bergman, L., Castelli, V., Lau, T., Oblinger, D.: Docwizards: A system for authoring follow-me documentation wizards. In: UIST'05. (2005)
6. Huang, J., Lu, B., Twidale, M.B.: Graphical abstract help. In: Proceedings of the 6th ACM SIGCHI New Zealand Chapter's international Conference on Computer-Human interaction: Making CHI Natural. Volume 94., New York, NY, CHINZ'05, ACM Pres (2005) 83–89

Extracting Use Case Specifications for Sequence and Class Diagrams Generation

Reynaldo T. Giganto

Department of Computer Science, School of Computing and Mathematical Sciences
University of Waikato, Hamilton, New Zealand
rtg4@cs.waikato.ac.nz

Abstract. Natural language processing (NLP) systems are able to assist developers in their analysis tasks during software development. These systems have empowered Computer Aided Software Engineering (CASE) tools in detecting classes in the requirements document (RD). However, unnecessary classes are included in the final model due to inherent language ambiguity. This paper outlines shortcomings of existing NLP-based CASE systems, the potential use of controlled language for writing RD, the generation of use cases (UCs) from RD to permit automatic generation of appropriate object models such as sequence and class diagrams, and the current research progress. The idea presented in this paper is a result of the early stage of my PhD research.

Keywords: software engineering, natural language processing, CASE.

1 Background of the Problem

Requirements elicitation, analysis, design, implementation and testing are the stages of the object-oriented software engineering development [1]. Software developers write the RD of the system under development during requirements elicitation. The analysis object models, in a form of class and sequence diagrams, are produced by developers from requirements document during the analysis stage. These diagrams are expressed in UML notations and are drawn in a software environment called CASE tool. The crucial activity in the analysis is the identification of classes, their attributes and the relationship between classes. An attempt to facilitate the analysis is done by NLP-based CASE systems (as shown in Table 1). These systems take RD as input and generate class diagram, however a common problem persists which is the generation of unnecessary and/or synonymous classes. This problem is caused by the inherent language ambiguity in the RD. The latest systems, Procasor and RAVEN, are unable to generate the class diagram.

Unnecessary and synonymous classes in the analysis object models affect the design stage of the OO software engineering development. It is therefore worthy to investigate the problem of ambiguity in the RD and come up with solutions to generate appropriate classes.

Table 1. List of Existing NLP-based CASE Systems

System	Input	Strength	Weakness
Saeke, et al system [7]	RD	identifies attributes and objects	high user intervention to select best classes
NL-OOPS [6]	RD	identifies attributes and objects/classes	unwanted classes, high user intervention
CM-Builder [3]	RD	identifies attributes and classes	synonymous classes
RECORD [2]	RD	link between objects is traceable	high user intervention to identify proper classes
GOOAL [8]	RD	identifies objects, classes and attributes, can generate sequence diagram	needs user intervention to resolve ambiguity
Procasor [5]	UCs	identifies objects, operations	UCs assumed to be not ambiguous
RAVEN [9]	RD	validates RD	extracted use case specifications too complex

2 Using Controlled Language for RD and Generating Use Case Specifications

Controlled language minimizes ambiguity. In this research, controlled language used to write the RD will be investigated. If statements in the RD are written using a restricted grammar and the words come from a limited lexicon, the syntactical structures of statements facilitate object analysis and classes with more than one meaning are avoided. However, it is not enough to control the RD because there are non participating objects on it. These objects appear as the unnecessary classes in the class diagram. Use cases are specific functionality of the system under development. A use case is a sequence of actions that the system can perform with interaction to actors of the system [4]. Moreover, a UC contains the exact classes participating in the tasks. It is therefore a major goal of this research to design an algorithm that generates UCs from the controlled RD. In contrast with existing NLP-based CASE systems, the new approach is to generate the class diagram from UCs not from the RD. Since UCs will be crucial, the UC statements will be in a restricted sentence structure and words comes from limited lexicon to avoid ambiguity. The controlled language should not be overly strict so as not to prohibit the expressiveness of the analyst in writing the RD and UCs are both comprehensible by the machine and the human analyst.

The algorithm to generate the UCs is to be implemented in the proposed NLP-based CASE system called ReoCASE (as shown in Figure 1). The system has two major modules: the NLP module and the software engineering module. The NLP module is composed of two sub-modules. NLP module 1 produces a parse tree of the statement in RD. Module 2 uses the parse tree and the UC restricted grammar and

lexicon to generate the initial UC. Module 1 and 2 operates for all statements in the RD. After all initial UCs are generated, module 2 consults a database of UCs of previously designed systems. These stored UCs will be matched and merged with the initial UCs to produce a final set of UCs and updated to the database for further reuse. The final UC statements will be analyzed and object semantics will be updated to an object semantic model (OSM). The SE module (composed of the OO analyzer and OO designer) interprets the OSM and display the UML sequence and class diagrams.

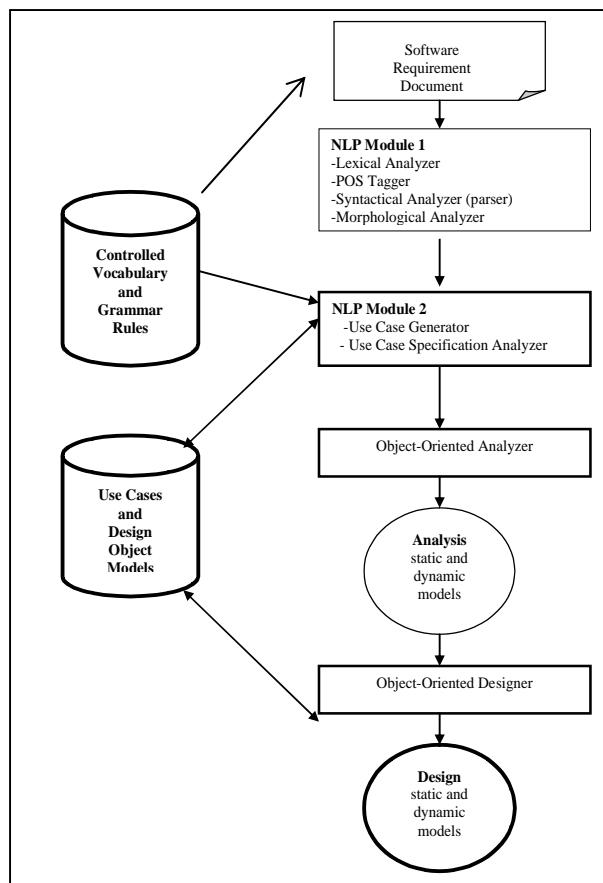


Fig 1. Proposed NLP-based CASE System

3 Research Progress

At present, the NLP tools such as the tagger, the parser and semantic labeler are installed and tested. Uncontrolled RDs are used as input to these tools. Initial experiments show that the parser builds more than one parse tree for ambiguous

sentences in the RD. The semantic labeler tool also produces more than two sets of labels for sentences in the RD. These results will be studied and used as a guide to design the controlled language.

The algorithm to generate the UCs is being finalized and the grammatical structures of the UCs are currently designed. The syntactical structures of these UCs are determined to be based on the following scenarios:

1. An actor requests something to be done by the system
2. A class of the system requests a task to be done by another class.
3. The system sends a message to an actor.

References

1. Bruegge, B., Dutoit, A.H. *Object-Oriented Software Engineering Using UML Patterns and Java*. Pearson, USA.2005. pp 16.
2. Borstler, J. *RECORD-Requirements Collection, Reuse and Documentation*, www.cs.umu.se/~jubo/RECORD.html <http://www.cs.umu.se/~record/T-red/master.html>
3. Harmain, H.M. and Gaizauskas R. “CM –Builder: An Automated NLP-based CASE Tool”, *The Fifteenth IEEE International Conference on Automated Software Engineering*, 2000, <http://dlib.computer.org/conferen/ase/0710/pdf/07100045.pdf>
4. Jacobson, I., Booch G., Rumbaugh, J. *The Unified Software Development Process*. Addison-Wesley, USA.1999. pp 135
5. Mencl, V. *Converting Textual Use Cases into Behavior Specifications [Technical Report]*.2004. Charles University, Prague, Czech Republic. <http://nanya.ms.mff.cuni.cz/publications/tr2004-5-Mencl-textual.pdf>
6. Mich, L. and Garigliano R. “NL-OOPS: A Requirements Analysis tool based on Natural Language Processing”. In the *Proceedings of Conference on Data Mining 2002*, Vol. 3, pp. 321-330, Southampton, UK:WIT Press.
7. Saeki, M., Horai, H., Toyama K., Uematsu, N., and Enomoto H. “Specification framework based on natural language”. In *Proc. of the 4th Int'l Workshop on Software Specification and Design*, 1987, pp. 87-94.
8. Perez-Gonzales, H. *Automated Techniques for Object Oriented Analysis and Design from Natural Language*, Doctoral Thesis, University of South Colorado Springs, Colorado, USA, 2003.
9. Ravenflow. *Bridge The Communication Gap Between Business and Application Delivery*, 2006, <http://www.ravenflow.com/>

Brain Gene Ontology (BGO): Tool to facilitate Education and Research in Neuroinformatics area

Vishal Jain, Nikola Kasabov, Paulo C. M. Gottgtroy, Lubica Benuskova and Frances Joseph

Auckland University of Technology, Auckland, New Zealand
{vjain, nkasabov, pgottgtroy, lbenusko, f.joseph}@aut.ac.nz

Abstract. This article summarizes our research on world-first brain-gene ontology (BGO) system that we use as a tool for educational purpose and research. It covers the concepts in the form of text, graphs, images, voice enabled animations from brain organization (structure, neuronal bundles, and synapses) through central dogma (gene regulation, ion channel proteins and mutations) to modelling and simulation of brain diseases. System has been shown for gaining knowledge on ontology and making discoveries.

1 Introduction

Ontology is a specification of a conceptualization of a knowledge domain [1]. For experimental purposes the medical ontologies [2], biomedical ontology (<http://www.bioontology.org/>) and the gene ontology (GO) have been created (<http://www.geneontology.org/>) [3]. The GO project provides a controlled vocabulary to describe gene and gene product attributes in any organism.

Brain-Gene Ontology (BGO) project lead to the development of a system that was first released and published with IEEE in December 2006, and, is focused on integrating information from different disciplinary domains such as neuroscience, bioinformatics, genetics, computer and information sciences. The system is designed to be used for research, simulation and teaching at different levels of tertiary education. This paper is a shorter version from our earlier publications on BGO and is specifically written to aim for NZCSRSC. BGO version 1 (BGOv1) is released on a CD and it is freely available to academic users and is for non-commercial use only.

2 Information structure of the BGO

BGO is comprised of three main parts: 1. brain organization and function, 2. gene regulatory network, and 3. a simulation model. Fig. 1 illustrates the detailed kind of information available in the BGO about information on genes, proteins, species and diseases. Part 1 contains information about neurons, about their structure and process of spike generation. Part 2 is divided into sections on neurogenetic processing, gene expression regulation, protein synthesis and abstract GRN. Part 3 has sections on computational neurogenetic modeling (CNGM) [4], evolutionary computation, evolving connectionist systems (ECOS) [5], spiking neural network [6], simulation tool [7] and CNGM results.

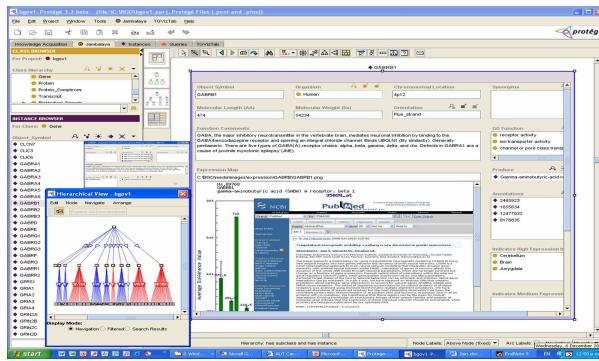


Fig. 1. Snapshot of the BGO

3 Evolving Implementation of BGO in Protégé

Protégé is an open source ontology building environment developed by the Medical Informatics Department of the Stanford University (<http://protege.stanford.edu/index.html>). We have developed a set of plug-ins to enable to visualize, extract and import knowledge from/into different data sources and destinations. BGO is based on Gene Ontology (GO), and Unified Medical Language System (UMLS), Pubmed and Gene Expression Atlas database. BGO utilizes a novel evolving conceptual metadata structure which allows incorporation of new discoveries and adapt its structure [8].

4 New Discoveries and teaching with BGO

BGO system provides conceptual links between data on brain functions and diseases, their genetic basis, and the relationships between the concepts (see Fig. 2). Our system also allows users to select and export the specific data of their interest to analyze in a software machine learning environment, such as WEKA (<http://www.cs.waikato.ac.nz/ml/weka/>) and NeuCom (www.theneucom.com) to train prediction or classification models and to visualise relationship information.

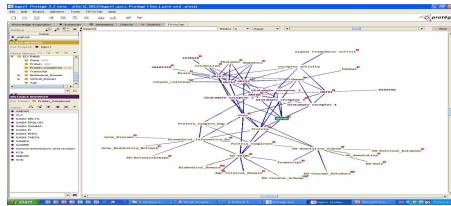


Fig. 2. Visualizing relationships between different molecules in BGO

BGO teaching interface is built using textual, graphical, audio and visual media. 3D animations, gives both a dynamic narrative introduction and overview to the BGO (see Fig. 3). It exemplifies the importance of use of ontologies in knowledge management and interpreting relationships between molecules and brain functions.



Fig. 3. Snapshots from novel voice enabled animations in BGO

5 Conclusions

The BGO is designed to facilitate active learning and research. It defines and represents the concepts existing in the domain of brain and genes, their attributes and the relationships between them. It is represented as a knowledge base and allows users to navigate through the rich information space from di-

4 V. Jain, N. Kasabov, P. C. M. Gottgtroy, L. Benuskova, F. Joseph

verse fields; to download data that can be analysed in a software machine learning environment; to visualize relationship information; and to add new information. In future, we plan to merge BGO [9] with GO [3] to enable more sophisticated discoveries.

Acknowledgments

This work is supported by NERF AUT0201, TAD, RELT and KEDRI. We thank Scott Heappey for creating all animations. For more information on brain-gene ontology system, please visit: www.kedri.info

References

1. Gruber TR (1993) A translation approach to portable ontologies. *Knowledge Acquisition* 5: 199-220
2. Pisanelli DM (ed) (2004) *Ontologies in Medicine*. IOS Press, Amsterdam
3. Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM (2000) Gene ontology: tool for the unification of biology. *The Gene Ontology Consortium*. *Nature Genetics* 25: 25-29
4. Benuskova L, Kasabov N (in press) Towards Computational Neurogenetic Modeling. Springer, New York
5. Kasabov, N. (1998) The ECOS framework and the 'eco' training method for evolving connectionist systems, *Journal of Advanced Computational Intelligence*, vol.2 (6):1-8
6. Kasabov N, Benuskova L (2004) Computational neurogenetics. *Journal of Computational and Theoretical Nanoscience* 1: 47-61
7. Benuskova L, Jain V, Wysoski SG, Kasabov N (2006) Computational neurogenetic modelling: a pathway to new discoveries in genetic neuroscience. *Intl. J. Neural Systems* 16: 215-227
8. Gottgtroy P, Kasabov N, MacDonell S (2006) Evolving ontologies for intelligent decision support. In: Sanchez E (ed) *Fuzzy Logic and the Semantic Web (Capturing Intelligence)*. Elsevier, Amsterdam
9. Kasabov, N., Jain, V., Gottgtroy, P., Benuskova, L. and Joseph, F. (2006). Bain-Gene Ontology: Integrating Bioinformatics and Neuroinformatics Data, information and Knowledge to Enable Discoveries. IEEE, ISBN: 0-7695-2662-4, pp. 13

Integrative Architecture for Concurrent Artificial Intelligence in Robotic Systems

Guy K. Kloss, Napoleon H. Reyes, and Ken A. Hawick

Institute of Information and Mathematical Sciences, Computer Science
Massey University, Albany/Auckland, New Zealand
`{G.Kloss | N.H.Reyes | K.A.Hawick}@massey.ac.nz`

Abstract. Artificial Intelligence is implemented in various sub-systems of larger applications. Especially in the field of robotics, many independent components need to interact cooperatively and in parallel in order to yield desired results. A high degree of coupling between the systems exists, and some hybrid algorithms require intensive cooperative interactions internally. In robotics it is seldom, that just one specific problem is targeted by an implementation. The proposed architecture is currently being realized for Robot Soccer (MiroSot and Humanoid) and for Soccer Simulation Leagues. The ability to re-use and re-combine the components on demand, therefore is essential, while maintaining the necessary degree of interaction. For this reason various software components are designed on top of a Service Oriented Architecture (SOA) using Web Services. An additional benefit is a platform independence regarding both the operating system and the implementation language.

1 Introduction

Robotic Systems are often used in place of humans or biological organisms. So their design specification requires taking many interacting sub-systems into account. That usually includes: vision and other sensory systems; motor control and other actuators; action and strategy planning; adaptability; and possibly others. Each of these systems may be by itself composed of several sub-systems employing a network of algorithms.

A typical setup for such a scenario is a robot soccer league's system [1] with a central vision system (Fig. 1). It is advantageous that a single computer may be used for all purposes. This machine then provides all control functionality of a single team and may introduce a bottle neck in computational resource to the system. For example a robot wants to move from location *A* to *B*:

1. The **vision system** detects the surrounding environment, by identifying object types and their location through the position in the image.
2. The **path planning** algorithm requires the robot's own position, the positions of the target location *B* and the positions of obstacles.
3. The **motion control** system requires the robot's own and the target's position and orientation for the *next* way point.

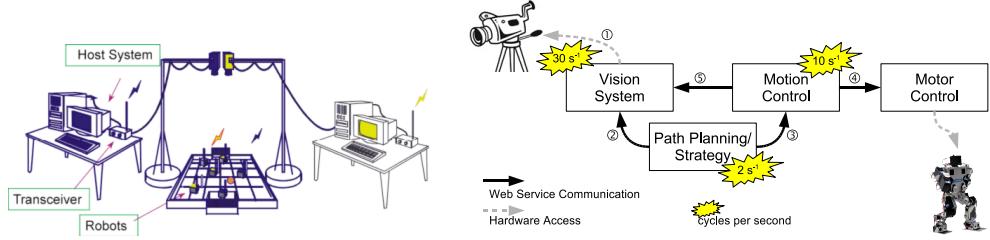


Fig. 1. Physical setup in a MiroSot competition.

Fig. 2. Distribution of workflow components on the network.

4. The **motor control** then requires the motion control's designated actuator states to map them to the devices using the hardware control interfaces.

In this example the various sub-systems ideally operate all at their individual rate (e.g. vision system with the frame rate of the camera) and communicate their information through interfaces on demand to other sub-systems (see Fig. 2).

This paper outlines the design decisions currently in the implementation phase for a new system, based on a Service Oriented Architecture (SOA) using Web Services that aims to achieve this.

2 System Architecture

If dealing with one specific problem domain only (e.g. Robot Soccer in the MiroSot small league), the software can be built on top of the (robot) vendor supplied software tool kit only. However, ongoing alterations by multiple independent developers in this code base significantly decrease the comprehensibility. Finally an implementation for multiple domains (e.g. for MiroSot and humanoid league) will completely fail on the provided vendor's development kit.

A goal was to cope with complexities introduced by the design for implementing specializations. It requires addressing multiple domains and distribution/parallelization of tasks, while still gaining independence of programming languages and the executing operating system.

By compartmentalizing the systems, the ties between functional sections were loosened. Communication between the components is *only* possible through the defined interfaces to them. The interfaces are exposed in a Service Oriented Architecture (SOA) as Web Services to the rest of the system, so that individual components can be placed on remote systems as well as locally. The next benefit of using Web Services for communication is, that all components can be implemented using an implementation and platform fittest for the task. This yields a building block approach in which a larger application can be constructed from available components in the “tool box”.

This approach contrasts the tightly coupled, monolithic architecture provided by the hardware vendors. In the vendor's system all steps of the process chain are forced to run at the same beat. Thus, in a decoupled system the number of executions for some sub-systems can be cut by a factor of 15.

3 Implementation

The implementation needs to provide mainly two things: Utilizing the *domain independence* of created components (robot soccer, soccer simulation, robot simulation, robot rescue and others) and focusing the development on *one specific component*, disregarding all other neighbouring components (or mocking them).

To highlight this, Sect. 1 provides the necessary steps for an analysis. The vision system (step 1) requires computationally expensive operations at a high rate (frame rate of camera). So it will be implemented most likely in a native/compiled language (e.g. C++) on a system that is capable of interfacing the camera. Further path planning and motion/motor control can be performed on another system. These could be implemented in languages that are by far more efficient in terms of development effort (e.g. Python, Java, Visual Basic).

All sub-systems are interfacing a communication layer, managing requests transparently through Web Services [2]. In case of changing demands this single layer *only* needs to be modified for an alternative coupling. Only the vision system and the motor control feature a direct access to hardware. Properly designed interfaces provided, these can easily be replaced by the virtual hardware of simulation systems, or replaced by alternative specific hardware drivers without introducing further dependencies. The path planning algorithm would request a list of vectors (positional and velocity) for all objects from the vision system. Even though the strategy and path planning may only need to be updated twice every second, it can still take advantage of the precise computations of the vision system – performing synchronously with the 30 fps of video capturing.

New/updated paths result in an update of way points for the robots. These vectors are requested on demand by the motion control, which, in turn, relays actuator settings to the motor control [3]. The motor control (or locomotion layer [4]) represents a robot’s embodiment. It converts control signals from the motion control (steering layer) into motion of the robot’s “body.” This motion is subject to constraints imposed by the body’s physically-based model. As this layer is dealing with the control of physical equipment it may take advantage of an increased update rate (10 Hz) due to the decoupled system.

4 Performance

Performance may be crucial to the success of a robotics application. The proposed architecture introduces communication overheads and uses interpreted languages rather than natively compiled ones. The main advantage, however, is that it frees one from worrying about other tasks introduced by the framework’s code used for the implementation. Thus, a loosely coupled application design will lead to more performance in terms of (the quality of) research output. Additionally – as the decoupled implementation is based on an SOA – services can be requested on demand, rather than at the time of availability.

In many publications on robotics a time frame of 33 ms for the maximum iteration duration of the control system are given. This value has been derived

from the frame rate ($30 \text{ fps} \rightarrow 33 \text{ ms}$) of cameras forcing the computationally dominant vision system and the rest of the control chain. As a result of decoupling this force is removed from most sub-systems, giving the freedom to spend clock cycles on the architecture and more suitable implementation languages.

To evaluate the communication overhead a “worst case” test has been undertaken. A purely interpreted implementation for Web Services in Python has been benchmarked. Times were taken on a single host and across two bridged 10 MBit/s network segments to determine the communication overhead. The slimmest communication (within one host) using no WSDL (Web Services Definition Language) interface description ($< 3.5 \text{ ms}$) has been tested against the full-fledged networked WSDL request ($< 35 \text{ ms}$). Using the highest determined request frequency (10/s) from Fig. 2 it can be seen that the communication overhead is low enough to perform sufficiently.

5 Conclusions

Investment in re-architecturing the system yields an easier and more focused implementation of sub-tasks. Freedom of choice for the implementation language away from C++ *only*, and freedom not to have to worry about “neighbouring” functional components to a problem and dramatically increase the speed of development. Due to a better focus on the current core problem, more innovative and robust solutions were gained.

Someone interested in improving the system’s AI, therefore, would be relieved of the impact on the rest of the infrastructure. Development can be performed in any programming language that is suitable (or familiar). Additionally, a widely used and standardized protocol for an SOA has been used to further reduce system induced barriers for communication. And finally, the host and operating system used for a specific sub-task is independent from the rest of the system.

Computational efficiency of the system could be gained. This was achieved by decoupling components for distribution on hosts most suitable for them. All components work in parallel at the most appropriate rate without being forced into the “dominant” beat (e.g. the camera’s frame rate), to save valuable CPU cycles and utilize parallel environments (multiple CPUs/cores or machines) most efficiently without sacrificing accuracy.

References

1. MiroSot league on FIRA Web Site. [Online] <http://www.fira.net/>
2. J. M. Vidal, P. Buhler, and H. Goradia, “The Past and Future of Multiagent Systems,” in *Proceedings of AAMAS Workshop on Teaching Multi-Agent Systems*, 2004.
3. C. L. Hwang, N. W. Chang, and S. Y. Han, “A Fuzzy Decentralized Sliding-Mode Control for Car-Like Mobile Robots in a Distributed Sensor-Network Space,” in *Presentation at the Third International Conference on Computational Intelligence, Robotics and Autonomous Systems*, 2005.
4. C. W. Reynolds, “Steering Behaviors for Autonomous Characters,” in *Proceedings of Game Developers Conference*, 1999.

A Discriminative Approach to Structured Biological Data

Stefan Mutter^{*} and Bernhard Pfahringer

Department of Computer Science
The University of Waikato
Hamilton, New Zealand
`{mutter,bernhard}@cs.waikato.ac.nz`

Abstract. This paper introduces the first author's PhD project which has just got out of its initial stage. Biological sequence data is, on the one hand, highly structured. On the other hand there are large amounts of unlabelled data. Thus we combine probabilistic graphical models and semi-supervised learning. The former to handle structured data and the latter to deal with unlabelled data. We apply our models to genotype-phenotype modelling problems. In particular we predict the set of Single Nucleotide Polymorphisms which underlie a specific phenotypical trait.

Keywords: Bioinformatics, Probabilistic Graphical Models, Semi-Supervised Learning, Single Nucleotide Polymorphism

1 Introduction

Biological data is highly structured in many different aspects. First of all there is an obvious structure in the sequence of DNA bases and amino acids. Different parts of a sequence, even far away from each other, can interact. These interactions may depend upon time and place. Another example is the regulatory network of gene expression which is a complex system.

In recent years there has been a push for methods that are able to deal with this kind of data, because, traditionally, Machine Learning has focused on independent and identically-distributed (iid) data [1]. This is why it is important to extend recent advances in machine learning theory and practice to structured, interdependent data.

2 Structured Data

As complex structured data becomes the focus of research, probabilistic graphical models become more and more important. They are well-founded in probabilistic and graph theory. Consequently a graphical model is a family of probability distributions that factorise according to an underlying graph [2]. A common

^{*} Author to whom correspondence should be addressed.

distinction between probabilistic graphical models is to differentiate between generative and discriminative models.

A generative model models the full joint probability distribution $p(y, x)$ where the variables y stand for the attributes of the entities we wish to predict and x stand for our observed knowledge [2].

In contrast a discriminative model is directly based on the conditional probability $p(y|x)$ [2–4]. Sutton and McCallum [2] point out that the crucial difference between a generative and a discriminative model is that the latter does not include a model for $p(x)$. First of all, for a classification task this model isn't needed anyway, secondly it often contains highly dependent features. Thus it is hard to model. However, if we want to integrate interdependent features in a generative model, Sutton and McCallum [2] offer two possibilities. On the one hand, potentially unwarranted independence assumptions can help, on the other the introduction of additional parameters can solve the problem. The second approach can only be used in a limited way because the model can easily become intractable. In contrast there exist well-known examples for the first approach e.g. Naive Bayes works well in document classification, but the independence assumption can also hurt performance on average across a range of applications where its discriminative counterpart logistic regression outperforms Naive Bayes [5].

We are looking at probabilistic graphical models for sequences. A Hidden Markov Model (HMM) is a well-known generative model for sequences. Its discriminative counterpart is a Conditional Random Field (CRF). This research project will focus on this discriminative technique. A CRF can be seen as an extension of logistic regression to arbitrary graphical structures [2]. Thus, in addition, CRFs relax the independent and identically-distributed assumptions in the sequence itself and between sequences. In Bioinformatics they have been successfully applied to gene prediction, RNA structural alignment, protein structure prediction [2] and finding gene and protein mentions in the literature [6].

3 Unlabelled Data

In many areas including Biology there exists a large amount of unlabelled data, because labelling is often difficult, time-consuming and expensive.

In this context where labelled and unlabelled data exists semi-supervised learning is a new approach in Machine Learning. It uses a potentially large amount of unlabelled data together with a usually small amount of labelled data to build a classifier. Generative models are the oldest semi-supervised learning technique [7].

Usually we can get $p(x)$ from unlabelled data [7]. For discriminative learning it is believed that semi-supervised learning cannot help if $p(x)$ and $p(y|x)$ do not share parameters [7, 8]. But, as a lot of approaches show, semi-supervised learning can outperform supervised learning when it is applied carefully and the underlying assumptions are correct [7]. Current research tries to adapt discriminative techniques to semi-supervised learning [9, 1].

4 Genotype-Phenotype Modelling

We are investigating biological sequences in particular sequences of single nucleotide polymorphisms (SNPs) where each one is a sequence alternation of a single nucleotide in a DNA sequence which occurs in at least one percent of the population.

A fundamental problem in contemporary genetics is the relation between genotype and phenotype known as genotype-phenotype modelling. Examples include identifying superior dairy cows that is identifying genes that are responsible for phenotypical traits which increase economic merit [10].

A lot of SNPs have been identified by high-throughput methods and need now to be analysed. SNPs can be used as genetics markers but they are also a reason for phenotype differences even though most SNPs have no effect on the phenotype. This is why it is important to find the SNPs that are related to a particular trait. This problem is called tagSNP selection [11]. Lee [11] emphasises the need for new, probabilistic methods.

As the number of discovered genes that contribute to a specific phenotype grows, so does the complexity of models describing genotype-phenotype relations [12]. Rodin [12] suggests the use of probabilistic graphical models to represent this kind of structured data.

5 Research Synopsis and Project Status

This PhD project aims for advances in genotype-phenotype modelling by the use of probabilistic graphical models, especially Conditional Random Fields. Due to the fact that in biological domains there is a vast amount of unlabelled data, the incorporation of semi-supervised learning methods is an important aspect. The primary source of data are biological sequences.

From a Machine Learning point of view the underlying hypothesis is that discriminative techniques should outperform generative ones on a classification task [9]. This statement is supported by Vapnik [13], who argues that it is better to solve the classification problem directly than looking at the more general problem of modelling the joint probability distribution as an intermediate step. However the work of Ng and Jordan [14] shows empirical results suggesting that discriminative learners have a lower asymptotic error but generative models approach their (higher) asymptotic error faster. Research in this area will also lead to get some more insights in the differences between generative and discriminative modelling.

We expect that the adaptation of models to perform semi-supervised learning should enhance them. However first results also highlight that there is a lack of appropriate graphical structures for biological problems. This is crucial, because the graph determines how a family of distributions get factorised. Because the problems are highly structured, a good representation of the structure is essential.

This allows two possible ways of optimisation: Using semi-supervised learning

and enhancing the graphical structure. Currently a combination of both seems to lead to promising results. The next step is to define an exact optimisation criterion.

We are going to build models using Conditional Random Fields and apply them to SNP data first. Currently we are setting up an experimental environment and pre-processing the data so that it can be used to solve the tagSNP problem.

References

1. Lafferty, J.D., Zhu, X., Liu, Y.: Kernel conditional random fields: representation and clique selection. In Brodley, C.E., ed.: Machine Learning, Proceedings of the Twenty-first International Conference (ICML 2004). (July 2004)
2. Sutton, C., McCallum, A.: An introduction to conditional random fields for relational learning. In Getoor, L., Taskar, B., eds.: Introduction to Statistical Relational Learning. MIT Press (2006) To appear.
3. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proc. 18th International Conf. on Machine Learning, Morgan Kaufmann, San Francisco, CA (2001) 282–289
4. Wallach, H.: Efficient training of conditional random fields. Master's thesis, University of Edinburgh (2002)
5. Caruana, R., Niculescu-Mizil, A.: An empirical comparison of supervised learning algorithms using different performance metrics. Technical Report TR2005-1973, Cornell University, Ithaca, USA (2005)
6. McDonald, R., Pereira, F.: Identifying gene and protein mentions in text using conditional random fields. BMC Bioinformatics **6** (2005) S6
7. Zhu, X.: Semi-supervised learning literature survey. Technical Report 1530, Computer Science, University of Wisconsin-Madison (2005)
8. Seeger, M.: Learning with labeled and unlabeled data. Technical report, University of Edinburgh (2001)
9. Brefeld, U., Büscher, C., Scheffer, T.: Multi-view discriminative sequential learning. In Gama, J., Camacho, R., Brazdil, P., Jorge, A., Torgo, L., eds.: ECML. Volume 3720 of Lecture Notes in Computer Science., Springer (2005) 60–71
10. Garrick, D., Snell, R.: Emerging technologies for identifying superior dairy cows in new zealand. New Zealand veterinary journal **53(6)** (2005) 390–399
11. Lee, P.H.: Computational haplotype analysis: An overview of computational methods in genetic variation study. Technical Report 2006-512, Queen's University, School of Computing, Kingston, Canada (April 2006)
12. Rodin, A., Boerwinkle, E.: Mining genetic epidemiology data with bayesian networks i: Bayesian networks and example application (plasma apoe levels). Bioinformatics **21(15)** (2005) 3273–3278
13. Vapnik, V.N.: Statistical Learning Theory. John Wiley & Sons (1998)
14. Ng, A.Y., Jordan, M.I.: On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In Dietterich, T.G., Becker, S., Ghahramani, Z., eds.: Advances in Neural Information Processing Systems 14, Cambridge, MA, MIT Press (2002)

Verifying Privacy Preserving Auctions

Ben Palmer

Supervised by: Kris Bubendorfer and Ian Welch

Victoria University

Wellington

Ben@mcs.vuw.ac.nz

Abstract. Auctions have many uses in computing. From allocating resources in a grid environment to selling an old pair of shoes on eBay, auctions have properties that benefit both buyers and sellers. A privacy preserving auction provides protection for bidders in a competitive environment where any information revealed can be used by other competing bidders to their advantage. In an environment with no pre-existing trust, like the Internet, an auction verification scheme can provide assurance that parties are correctly executing the auction protocol. For example, a bidder may want to verify that the auctioneer conducting the auction counted their bid, and that the auction outcome has been correctly computed. This project investigates the use of verification in privacy preserving auctions. We plan to add verification to an existing privacy preserving auction scheme, as well as implementing a verifiable privacy preserving auction scheme.

Key words: Network Research, Internet Security, Distributed Systems

1 Introduction

Suppose you were taking part in a sealed bid online auction for some goods. How would you know the auctioneer has named the correct bidder as the winner, or even counted your bid? If it is a second price auction (the winner pays the second highest bid price), how would you know the auctioneer has not inflated the second price for their own profit? One option is to always use a trusted party, but in the world wide marketplace of the Internet, this is not always possible. This lack of transparency, and pre-existing trust on the auctioneer, are common problems identified with sealed bid auctions. This is especially true of Vickrey or second price auctions that have the advantage that the dominant strategy is for bidders to submit bids of their actual evaluations. It has been speculated that the main reasons for not using a Vickrey auction in real world applications is the ability of the seller to change the auction result, or reveal bidder's private information [1].

If bidders can verify that the auction was executed correctly, we can reduce the need to trust the auctioneer. The key properties we want to verify are that the auction protocol has executed correctly, and that the auctioneer has not

tampered with bids or results. This prevents the auctioneer changing the auction result, removing one of the problems with Vickrey auctions. The auctioneer can also verify that the bidders have submitted valid bids. Verification results can also be provided to a reputation service for future reference. Additionally, privacy preserving auctions are used to prevent the revealing of bidder's private information. We plan to use these auctions to perform resource allocation in a Grid architecture.

2 Background

A large amount of research has been done recently on privacy preserving auctions, and the related field of electronic voting. Auction protocols can execute various types of auction: first price (one good, winning bidder pays highest price); Vickrey (one good, winning bidder pays second highest price); combinatorial (many goods, winning bidder pays highest price); and, the generalised Vickrey auction (GVA) (many goods, winning bidder pays highest price less a discount). What we want is a privacy preserving verifiable GVA that hides the values of bids, as these are true valuations, and allows bidders to check that the auction protocol has been executed correctly. However, the existing schemes reviewed below only go part way towards this ideal.

Brandt has presented a bidder resolved auction protocol [2] that is verifiable, but does not support a GVA or combinatorial auction.

A verifiable privacy preserving auction scheme that supports Vickrey auctions, but not a GVA, has been developed by Kikuchi [3]. It makes use of verifiable secret sharing to allow bidders to verify auctioneer's calculations, and auctioneers to verify the bidder's encrypted bid values [3].

Naor, Pinkas, and Sumner have developed a privacy preserving auction scheme that supports combinatorial auctions [4] using garbled circuits. An auction issuer creates a garbled circuit that an auctioneer uses to conduct a privacy preserving auction. Privacy is preserved as long as the two parties do not collude. This auction scheme provides verification for bidders to verify that their bids were counted and to verify the auction result, but no way to verify that the auction issuer and auctioneer have not colluded. The garbled circuits that have to be sent between the auction issuer and the auctioneer can be quite large; the authors suggest possibly sending them on a CD or DVD.

Lipmaa, Asokan, and Niemi have developed a first price auction scheme that partitions information between two parties so that one party working alone cannot subvert the auction [1]. A series of range proofs verify the auction.

Yokoo and Suzuki have developed a privacy preserving combinatorial auction scheme [5] using homomorphic encryption and dynamic programming. Homomorphic encryption is used to carry out calculations on encrypted values thereby preserving privacy, and dynamic programming is used to find an optimal solution and to allow distributed calculation of a result. This auction scheme does not provide any verification.

3 Goals

The privacy preserving auction scheme designed by Yokoo and Suzuki [5] has many useful properties. It supports combinatorial auctions and has been extended to support a GVA. It keeps losing bid values secret, and is non-repudiable, that is bidders cannot deny their bids. The primary goal of this project is to add verification to this auction scheme while not breaking these security properties.

The main techniques we will be using for the verification are a zero knowledge proof of knowledge of a discrete logarithm, plaintext proof of equality, and a verifiable shuffle of encrypted values.

3.1 Zero Knowledge Proof of Knowledge of a Discrete Logarithm

Zero knowledge proofs are used to prove some statement, without revealing any other information other than what is known before the proof was executed. Using a Σ protocol of commit, challenge, response, we can prove knowledge of a discrete logarithm in zero knowledge.

For example, say we have $y = g^x$ such that y and g are publically known, but x is known only to a prover. The prover can prove to a verifier that they know x without revealing it by conducting the following protocol:

- Prover picks a random number z and calculates $a = g^z$ and sends a to the verifier.
- Verifier picks a random number c and sends c to the prover.
- Prover calculates $r = z + cx \bmod q$ and sends r to the verifier.
- Verifier checks that $g^r = ay^c$ as $g^r = g^{z+cx} = g^z g^{cx} = ay^c$.

This technique is the basis for proving that encrypted bids are well formed, and that the auctioneer has correctly compared a set of values without revealing individual values.

3.2 Plaintext Proof of Equality

Given two homomorphic encrypted values E_1 and E_2 , we can prove they decrypt to the same item due to the homomorphic property of the encryption. The prover will divide one item by the other using modulo division and then decrypt and publish the result. If the result is 1, then we know the plaintexts of the encrypted values must be equal. This is used to prove the bids are well formed, and to prove that randomisation carried out by an auctioneer is correct.

3.3 Verification of a Secret Shuffle of Encrypted Values

Furukawa and Sako have presented a method to verify a secret shuffle of encrypted values [6]. Given a set of encrypted values, we can apply a permutation to these values, randomise them, and publish the result. We can then prove that we have applied a permutation and not tampered with any of the underlying plaintext values, without revealing the permutation used or any of the plaintext values. This technique is used to prove that one of a set of bids is the maximum bid, without revealing the bid value.

4 Current Status and Future Work

Currently, we have designed and are implementing the verification process for the homomorphic privacy preserving GVA scheme by Yokoo and Suzuki [5]. A second phase of the project will involve implementing the verifiable privacy preserving auction scheme by Naor, Pinkas, and Sumner [4].

After these steps are completed, we will have two verifiable privacy preserving GVA schemes to compare. These two schemes are ideal choices as one uses a group of servers and threshold encryption [7] to provide security, while the second protocol has two servers and relies on these two servers not colluding to provide security.

We then plan to examine the security and computational and communication complexity of the two solutions from an abstract perspective, as well as a practical perspective based upon the implementation of the schemes. We can compare the time taken for the two auction schemes to complete based on parameters such as number of bidders, number of goods, or the number of available prices. We can also compare how much data is transferred while conducting the auctions. We also intend to perform a security analysis.

The two auction schemes are intended to be used to perform resource allocation in a grid system, and so will also be added to the Globus grid architecture.

References

1. H. Lipmaa, N. Asokan, V. Niemi: Secure Vickrey auctions without threshold trust. In Proceedings of the 6th Annual Conference on Financial Cryptography (2002)
2. Felix Brandt: How to Obtain Full Privacy in Auctions. International Journal of Information Security (2006)
3. Hiroaki Kikuchi: (M+1)st-Price Auction Protocol. FC '01: Proceedings of the 5th International Conference on Financial Cryptography (2002) 351-363
4. Moni Naor, Benny Pinkas, Reuben Sumner: Privacy Preserving Auctions and Mechanism Design. EC '99: Proceedings of the 1st ACM conference on Electronic commerce (1999) 129-139
5. Makoto Yokoo and Koutarou Suzuki: Secure multi-agent dynamic programming based on homomorphic encryption and its application to combinatorial auctions. AAMAS '02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems (2002) 112-119
6. Jun Furukawa and Kazue Sako: An Efficient Scheme for Proving a Shuffle. CRYPTO '01: Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology (2001) 368-387
7. Adi Shamir: How to share a secret. Communications of the ACM, Vol. 22 (1979) 612-613

Simulation for LEGO MINDSTORMS™ Robotics

Yuan Tian, Keith Unsworth, and Alan McKinnon

Applied Computing Group
PO Box 84, Lincoln University, Canterbury, New Zealand
E-mail: {tiany5, unsworth, mckinnon}@lincoln.ac.nz

Abstract. In this Masters project, we develop a simulator for LEGO MINDSTORMS™ to simulate the actions of LEGO robots in a virtual 3D environment using Open Dynamics Engine (ODE) and OpenGL combined with ROBOLAB™. Users can test their ROBOLAB program through the simulator before downloading it into the LEGO MINDSTORMS RCX. Moreover, the simulator can track and display ROBOLAB program execution as the simulation runs so that users can view that a certain code execution refers to a particular action of the virtual robot.

Keywords: Graphics, Simulation, LEGO MINDSTORMS, ROBOLAB, OpenGL, ODE

1. Introduction

Simulation plays an important role in education and training, as well as in robotics development, such as Webot™ [2]. This is because simulation not only provides a faster and cheaper way to design and develop robots, but also enables users to explore environments and experiment with robotic events that may be unavailable because of distance, time, or safety factors.



Fig. 1. Robotic Command Explorer (RCX)

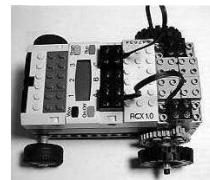


Fig. 2. A sample LEGO MINDSTORMS robot with two motors connected to the RCX

LEGO MINDSTORMS and ROBOLAB are used to help students learn science, technology, engineering, and mathematics concepts with hands-on, naturally motivating building sets, programming software, and curriculum-relevant activity materials [1]. LEGO MINDSTORMS toolkits contain a programmable LEGO® brick

called a Robotic Command Explorer (RCX) (see Fig. 1), LEGO pieces, sensors and motors.

To enable a LEGO robot to execute certain actions, users can use ROBOLAB to program the behaviour of the robot and download the program into the RCX. In the ROBOLAB environment users can program the functions of a LEGO robot using LabView visual programming terminology [3], which uses icons that are wired together, rather than typing in lines of code (see Fig. 3). Hence, with ROBOLAB, users including young children can design and program their LEGO robot by selecting recognizable images.

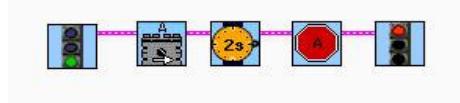


Fig. 3. An example of a ROBOLAB program, which instructs a motor to go forward for 2 seconds and stop

However, users cannot test and debug their ROBOLAB control program before downloading it into the RCX. As a result, the LEGO robot might not have the expected behaviour because of errors within the program. Moreover, there might be some people who may not have access to the LEGO MINDSTORMS toolkits because of lack of resources. Hence, in this project, we develop a simulator for LEGO MINDSTORMS using OpenGL graphics software library and Open Dynamics Engine (ODE). ODE is a software library for simulating articulated rigid body dynamics. [4]

2. System Overview

The controller program, ROBOLAB, generates instructions in the LEGO Assembly Language (LASM). See for example Table 1. The simulator reads the LASM instructions directly from ROBOLAB, and generates the corresponding actions of a virtual LEGO robot using OpenGL together with ODE. OpenGL is used to display both the components of the robot and the 3D environment in which it moves. ODE constructs a virtual robot using these components, incorporating realistic physics effects, such as wheel turning friction and collision detection. Note that before running a simulation OpenGL and ODE are used to construct pre-defined environments and virtual robots so that users can select a robot and its environment immediately before the simulation runs.

Table 1. Examples of LASM codes

Example Codes	Comments
dir 2,1	change the direction of the listed outputs (e.g., a motor)
Label1003	address that can be used to jump to different program instructions

3. System Architecture

The system has two main parts including the robot programming environment (ROBOLAB) and the robotic simulator (see Fig. 4). In the simulator, the actions of a pre-defined virtual robot within its environment is programmed with OpenGL and ODE in the C++ language. As the Robot Controller interprets the LASM commands the user can change the 3D view using the GUI.

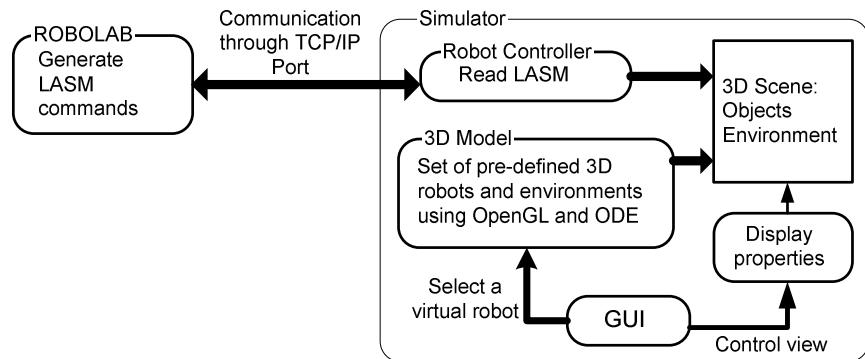


Fig. 4. The main components of the system

Robot Controller. The robot controller reads the LASM commands from ROBOLAB and calls an executable C++ function corresponding to each command, in order to generate the actions of the robot (see Fig.5).

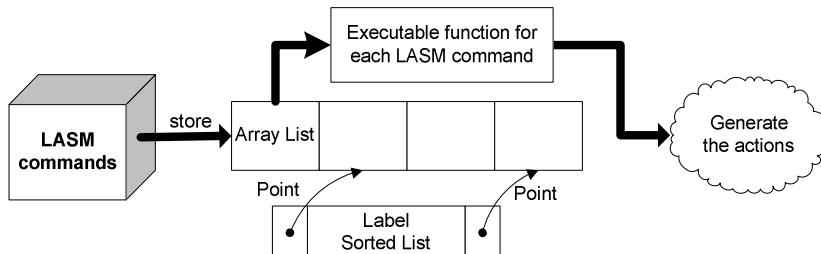


Fig. 5. The structure of robot control program

An array list is used to store all the LASM commands that are generated from ROBOLAB in sequence. A label dictionary sorted list is also created based on the LASM array list. This stores the index number of an address, Label1002 for instance, as a reference to the array list so that we can locate the label in the array list. We have implemented a simple virtual machine which executes each LASM command within the array list in sequence. The sorted list provides quick access to the appropriate location in the array list when branching occurs. As each LASM command is

executed a case statement is used to select the appropriate function to be executed. The function may involve execution of OpenGL or ODE commands or it may simply involve the transfer of control to elsewhere in the array list via the sorted list of labels.

3D Scene. The 3D robots are pre-defined using ODE and OpenGL. OpenGL displays the environment with the use of features such as lighting and texture mapping, and generates components such as a box, sphere and cylinder which are used to construct a virtual robot. ODE constructs a virtual robot by connecting these components using different joint types and including integrated collision detection with friction to simulate realistic physics.

GUI. Users visualize the behaviours of the virtual robot, and are able to interact with it within the 3D environment using a GUI. For instance, users need to choose a pre-defined virtual robot from a list before the simulation runs. Users also can change the viewpoint and lighting using the keyboard or mouse. In addition, program execution is tracked and presented in the display area.

TCP / IP Communication. The simulator communicates with ROBOLAB through a TCP/IP port. The simulator acts as a server application and receives and sends data using a TCP/IP socket. On the client side, ROBOLAB sends program data to the simulator through the same TCP/IP port.

4. Results and Future work

Currently, the project is still in the development stage. The simulation is limited to the simple motion of a vehicle with two motors. The virtual vehicle can go forward or backward for a certain time period that is controlled by ROBOLAB. For future enhancement, sensors including a touch sensor and a light sensor will be implemented to incorporate physics provided by ODE so that the virtual vehicle can perform more complex actions, such as reacting to collisions or following a pre-defined track.

References

1. B.Erwin, M.Cyr, and C.B.Rogers. (2000) Lego engineer and robolab:Teaching engineering with labview from kindergarten to graduate school. *International Journal of Engineering Education*, 16(3)
2. Michel, O. / Cyberbotics Ltd (2004) Webot™: Professional Mobile Robot Simulation, pp. 39-42, *International Journal of Advanced Robotic Systems, Volume 1 Number 1, ISSN 1729-8806*
3. Jeffery, T. (2002) LabVIEW for Everyone (2nd edition) *Prentice Hall*
4. Russell, S. (2004) Open Dynamic Engine User Guide, v0.5, retrieved from <http://www.ode.org/>

Was it Pod Worthy? A Preparatory Plan for Evaluating Podcasting in Higher Education

Anna Wingkvist¹ and Jason Alexander²

¹School of Mathematics &
System Engineering
Växjö University,
Växjö, Sweden
{anna.wingkvist@vxu.se}

²Department of Computer Science &
Software Engineering,
University of Canterbury,
Christchurch, New Zealand
{jason@cosc.canterbury.ac.nz}

Abstract: Students and teaching staff in higher education are constantly looking for new tools to help them study and teach more efficiently. The University of Canterbury began ProjectPodcast to introduce podcasting to a number of subjects as an add-on to the current course curriculum. Podcasting is being used to enhance mobile learning and enthuse both students and lecturers. Previous podcasting evaluations show that there exists a need for both audio content from lectures, or so called “LectureCasts” as well as supplementary material or “Sup!Casts”. In this study, we will be evaluating ProjectPodcast. The evaluation is aimed at both the student population as well as the lecturing staff in order to gain knowledge about their impressions of podcasting. Prior podcasting surveys have received low response rates, due to the choice of time, location and medium. Hence, our evaluation plan has been created with the goal of encouraging feedback from students and lecturers. In this paper we present our preparatory plan for evaluating ProjectPodcast.

Keywords: Podcasting, higher education, evaluation, educational technology, mobility

1 Introduction

The current generation of students in higher education are rapid adopters of new technology, with lecturers always looking for new and exciting methods to encourage students to continue their studies outside the classroom. Podcasting is seen as an innovative way to engage students in their course work. The simple act of utilising technology as part of their learning is often seen as a more attractive option to students than the thought of sitting down to read course material. Several universities have picked up on this and are producing their own podcasts to aid students in their studies.

Podcasts are multimedia files, usually in the mp3 format, distributed by subscription to an RSS-feed that allows downloads to be ‘pushed’ onto digital playback devices. Some universities are providing audio content of lectures, while others are producing supplementary material, with the hope of stimulating interest in the subject area. By nature, podcasts are informal recordings often made in one continuous session with little or no editing done before release. Despite often large

audiences, inducing feedback from listeners is one of the most difficult tasks for podcast creators. This is especially important in the educational situation, where teaching staff need to know whether their efforts are being wasted on producing podcasts or whether the material covered could be adjusted to further benefit student's learning. There has also become an ever increasing need to convince management that time and money is being spent appropriately. The best way to provide evidence of the worthiness of a concept is through evaluation.

This paper describes the preparatory planning for evaluating ProjectPodcast—an initiative at the University of Canterbury to encourage lecturers from a variety of departments (including Computer Science, Economics, Japanese, Music and Education) to provide podcasts for their courses. We firstly give examples of how podcasts are currently used in higher education and then consider evaluations of podcasting that have already been conducted and the shortcomings of these. We then describe our preparatory planning for the creation of the ProjectPodcast evaluation. The evaluation includes both the student population and the teaching staff who are producing the podcasts. We also discuss our thoughts on how to encourage participation, especially from the students involved in the project.

2 Podcasting in Higher Education

Universities have taken two different paths in providing podcasts for their students. The first has been to provide recordings of the lectures, allowing those who miss all or part of them to catch-up. We call these "LectureCasts". The second is to produce podcasts containing supplementary material, news and information, which may or may not form part of the examinable course content. These we call these "Sup!Casts", the name deriving from: "supplementary" and the colloquial abbreviation of "What's Up?"—"Sup?". The question mark is replaced with an exclamation mark in recognition of podcasting being a one way medium.

Podcasting in higher education has caught on but evaluation data is scarce and only a few universities have published their results. An online survey for the podcast pilot in 2005 at University of Washington reported by Lane (2006) found that 70% of students said that the LectureCasts supported their learning and were helpful when preparing for homework and exams. The response rate was low, 41 out of 148 enrolled students completed the voluntary survey, but this may indicate the perceived value. Interestingly, 81% of the students used a desktop computer rather than a portable player to listen to the podcasts. The University of Southern California had two spring courses in 2006 with their LectureCasts being evaluated and the outcome was regarded as positive in both cases (Wolff, 2006). However, the reasons for their success differed as one course had a large number of students for whom English is their second language who listened to the whole lectures again, while the participants of the other course valued having the recording to replay specific explanations to understand difficult material.

At the University of Canterbury an initial survey asked students to report their level of interest (5-point scale from 1 for not interested to 5 for very interested) for three types of material in the podcasts. The number of students showing an interest

level of 4 or 5 (i.e. more than neutral) was 50% for the recordings of lectures, 72% for summaries and extra information, and 65% for related topical issues. Overall the students indicated a preference for the supplements, although the demand for LectureCasts is present (Bell et al., 2006). A deeper analysis of the collected surveys revealed that students are very reluctant to respond to open ended questions. Likert scale or tick-box style questions were far more likely to be answered. This is important for future questionnaires as some respondents may have been ‘turned off’ by the large amount of writing required.

The results from the previous evaluations have lead us to believe there is a need for both LectureCasts as well as Sup!Casts. LectureCasts are especially useful for students for whom their native language differs from that of the course. Also, the opportunity to be able to replay all or certain parts of a lecture is valuable when the material is complex, if the student lost focus or simply did not attend. It seems that LectureCasts are used in conjunction with other study equipment (notes, textbooks, and websites). This reinforces our view that podcasting lectures does not take full advantage of the potential of the medium to facilitate mobile learning.

3 Evaluation Methodology

The evaluation of ProjectPodcast is to be performed in-house, so to reduce bias and outside critique, we will follow Oliver’s (2000) structured model of evaluation. In this section we describe how the evaluation steps will be followed for ProjectPodcast:

1. *Identification of Stakeholders*: In the case of ProjectPodcast we have three groups of stakeholders: the administrators, the lecturers and the students. The administrators include the funding body and the staff involved in organising and promoting the project. The lecturers are those in various departments who have volunteered to be involved in the project and produce podcasts for their courses. Finally, we have the students who will be listening to the podcasts.

2. *Selection and Refinement of Evaluation Question(s), based on the Stakeholder Analysis*: Before commencing the evaluations, the questionnaires and the core interview questions will be shown to members of the administration and teaching staff to ensure the evaluation will provide them with all of the feedback they require.

3. *Selection of Evaluation Methodology*: In any evaluation there is the choice between qualitative and quantitative methods. The area of learning technology is inherently multidisciplinary and we believe it is better to choose the evaluation method best suited to the situation instead of sticking to one paradigm. In our case we aim to mix the two, triangulating in order to achieve valid results.

4. *Selection of Data Capture Techniques*: For the students this will mainly be through questionnaires and technical data collection but also some semi-structured interviews and focus groups will be held. Electronic questionnaires are preferred, as they have several data collection advantages: they allow easy data collation when the survey is completed and they allow easier dissemination to a large group of users. Student questionnaires will be anonymous and completed online, however interested students will also be able to volunteer for focus group discussions via a tick-box. The survey forms that we have created also contain questions regarding age, gender and

language proficiency to allow us to determine whether the medium is better suited to certain demographic groups. We will also use technical records available from the podcast servers to create statistics on the RSS subscription rates, the number of downloads per podcast and the location of the requests for the podcasts. Student questionnaires will be carried out at both the half-way point (to allow lecturers to adjust their podcasts) and at the completion of the course.

The evaluation of the lecturer's experience will be through questionnaires and semi-structured interviews. The teaching staff are likely to be more willing to provide feedback on the project, as they will have actively volunteered to be involved in it. Questionnaires will be used to gather basic statistical data and then interviews will be used to allow us to gain a more in depth knowledge and understanding of their experience and issues they had.

5. Selection of Data Analysis Techniques: Analysis of tick-box style questions will be performed with standard statistical analysis tools. The written comments from the questionnaires and the recorded interviews/focus group discussions will be combined into a report.

6. Choice of Presentation Format: Both a formative and a summative report will be presented to the stakeholders of ProjectPodcast.

Conclusion

The aim of this work was to generate a preparatory evaluation plan to enable us to effectively and efficiently assess ProjectPodcast. From our experiences and that of others we have found that students are reluctant to give feedback on podcasts. To circumvent this we will use closed questions, with tick boxes and conduct the questionnaires using a web based system. This data will be combined with that gathered in the semi-structured interviews and focus groups. The survey will include the lecturers, as they have been overlooked in the past. This will allow for more rounded conclusions on the perceived value of podcasts to be drawn, with perspectives from both the teaching staff and students involved. Finally, positive results will be used to encourage management and funding bodies to continue their support for technology based projects of this kind.

References

- Bell, T., Cockburn, A., Wingkvist, A. and Green R.: Podcasts as a Supplement in Tertiary Education: an Experiment with Two Computer Science Courses. Proceedings of MoLTA 2007, p. 70-77 (2007).
- Lane, C.: Podcasting at the UW: An Evaluation of Current Use. The Office of Learning Technologies, University of Washington (2006).
- Oliver, M.: An Introduction to the Evaluation of Learning technology. Educational Technology & Society 3(4) ISSN: 1436-4522 (2000).
- Wolff, T.: Podcasting made Simple. SIGUCCS '06, November 5-8, Edmonton, Alberta, Canada (2006).

3-4 Heap Workspace Operations

Tobias Bethlehem

Department of Computer Science, University of Canterbury
Christchurch, New Zealand
`tbe13@student.canterbury.ac.nz`

Abstract. As an alternative to the Fibonacci heap data structure, and a variation of the 2-3 heap, this preliminary research paper presents a summary of the 3-4 heap workspace operations. When implementing graphic algorithms such as Dijkstra Single Source Shortest Path algorithms, this new data structure can be consumed as the priority queue being used.

1 Introduction

Derived from Tadao Takaoka [1] 2-3 heap data structure, this research paper will provide a brief summary of one domain area within the 3-4 heap, workspace operations. Left out of this paper are the top level operations which by itself is an equally sized research domain, a small summary is given later.

At University of Canterbury, I am studying part-time for a postgraduate Master qualification researching into the 3-4 heap data structure. Having recently commenced this research, there unfortunately are no quantitative figures available because the implementation stage is currently a work in progress, estimated completion around middle of year 2007. This naturally means that all of the theoretical analysis governing workspace and top level operations have been completed. My research goal is to discover the efficiency of this data structure in relation to its 2-3 heap counterpart when both are ran under identical operating scenarios.

The 3-4 heap, like 2-3 heap, can be consumed by Dijkstra [2] Single Source Shortest Path (DSSSP) as the data structure implementing a priority queue. Testing scenarios will involve generating a random graph and feeding this into the same DSSSP implementation twice, respectively consuming 2-3 heap and 3-4 heap. This approach will generate comparable quantitative figures.

Glossary of terms used in this paper are provided towards the end of this paper.

2 Heap Workspace

The 3-4 heap has a semi-rigid structure (see Fig. 1) where the minimum number of nodes on a trunk is three and the maximum is four, this permits a trunk to shrink by one, hence the term 3-4 heap. The number of nodes per heap is between nine and sixteen. If there is a requirement to shrink a trunk by more

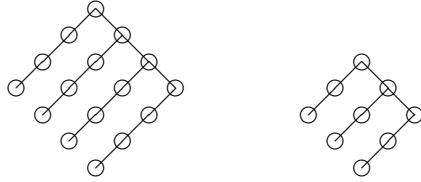


Fig. 1. A 3-4 heap with maximum and minimum number of nodes

than one node, then an adjustment is made by moving a few nodes from nearby trunks within this heap. Should it no longer be possible because the workspace has less than nine nodes, then the adjustment will trigger a propagation event where the heap is merged into neighbouring higher degree heap. This adjustment will require amortised cost analysis.

To aid amortised analysis, the trunk is given a measurement value called ‘potential’. This is calculated by counting the number of comparisons required to place each node into its correct position in ascending order, this position is found by using linear search. We define the potential of a trunk with one node to be 0, two nodes to be 1, three nodes to be 3, and four nodes to be 6.

Amortised cost is a figure derived by taking the net difference in potential of a heap before and after a node removal operation, and adding to this the number of node-to-node comparisons made. If this figure is of value zero, then the amortised cost is said to be free, if positive then a cost has been incurred, and if negative then a profit has been made.

In Fig. 1, the left-most is a complete 16 node heap with potential of 30, and the right-most is a complete 9 node heap with a potential of 12, these form the upper and lower bounds of standard arrangement and a heaps workspace.

In Fig. 1, studying the root node on either heap, the trunk sloping left-down (7pm on a clock face) is in the 1st dimension and the trunk sloping right-down (4pm on a clock face) is in the 2nd dimension. If this structure is part of a larger 3-4 heap where each node itself is another heap, those trunks can be said to be of i -th dimension and $(i+1)$ -th dimension. It should be noted that these additional workspaces are not taken into consideration, only the nodes within the current workspace are.

3 Performing Node Comparisons

Standard linear ascending comparisons are achieved by starting with the smallest node on a trunk and comparing it with the node that requires insertion. If the insertion position is not found, then the next node on the trunk is compared against the node that requires insertion, and so forth.

4 Workspace Operations

A decrease key operation occurs when the number of nodes in the 3-4 heap workspace is reduced by one. The node that gets removed is always located on the i-th dimension. The aim during these operations is to keep the amortised cost as small as possible. The amortised cost is calculated by taking the workspace net potential difference before and after a decrease key operation, and then adding to this the number node-to-node comparisons made.

If there is a requirement to shrink a trunk below three nodes, then an adjustment is made by moving a few nodes from nearby positions to ensure the heaps rigid dimensions are retained. Should this adjustment process no longer be possible because the workspace has less than nine nodes, then it cannot be restructured and must be propagated into a higher dimension. This propagation event will continue indefinitely until the heap has stabilised into standard arrangement form. A critical property required for moving this tree from one dimension into another, is for the amortised cost to equal zero.

It is calculated by myself that there are 14 decrease key operations required for a 3-4 heap to reduce in size from its maximum until a propagation event occurs, however these have been omitted from this paper due to the page limit requirement.

5 Top Level Operations

This section has been left out of this paper to ensure its conciseness. Generally top level operations can be viewed as having a collection of pigeon holes (positions) where each pigeon hole can only hold 3-4 heaps based upon their degree. The operations that can be done are insert, delete minimum and merge. Operation insert controls adding a 3-4 heap into its correct top level position. Delete minimum removes the root node of the 3-4 heap found in a top level position, causing this heap to break up. Merge handles the repositioning and reconstruction of the delete minimum operations fragments into their new appropriate top level position.

6 Glossary of Terms

Identified in Fig. 2 is the terminology used to describe a 3-4 heap

node	This is the circle and is used to represent a value
branch	This is the line connecting two nodes and a branch of trunk length of one
root node	This is the node with no parent node
trunk	A trunk is a straight line connecting several nodes
tree/heap	This is a group of trunks connected together
degree	The number of trunks connected to the root node

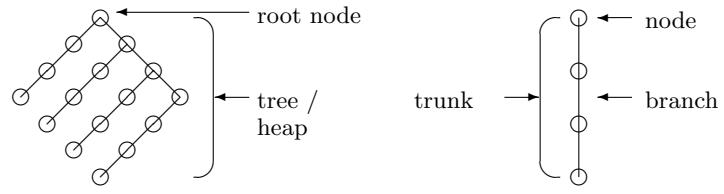


Fig. 2. Terminology

7 Conclusion and Future Work

The purpose of my research is to prove or otherwise prove, that the 3-4 heap is more efficient than the 2-3 heap acting as a data structure consumed by Dijkstra Single Source Shortest Path algorithm. Due to the page limit of this paper, I unfortunately was not able to describe in full details the inner workings of the 3-4 heap or highlight the differences and similarities it shares with the 2-3 heap.

From the completed theoretical analysis, all the key features of the 2-3 heap have been retained but the actual impact on performance from these subtle differences will not be established until implementation is completed around the middle of year 2007.

References

1. Takaoka, T., Theory of 2-3 Heaps, *Discrete Applied Mathematics*, Volume 126, 115–128, 2003
2. Dijkstra, E.W., A note on two problems in connexion with graphs, *Numer. Math.* 1 (1959) 269–271.
3. Prim, R.C., Shortest connection networks and some generalizations, *Bell Sys. Tech. Jour.* 36 (1957) 1389–1401.
4. Fredman, M.L. and R.E. Tarjan, Fibonacci heaps and their uses in improved network optimization algorithms, *Jour. ACM* 34 (1987) 596–615
5. Vuillemin, J., A data structure for manipulating priority queues, *Comm. ACM* 21 (1978) 309–314.

Using Web services to manage data from a variety of client applications

Yanbo Deng, Clare Churcher, Walt Abell and John McCallum

Applied Computing Group
P.O. Box 84, Lincoln University, Canterbury, New Zealand
{dengy6, churcher, abell} @lincoln.ac.nz

Crop and Food Research
Private Bag 4704, Christchurch, New Zealand
McCallumJ@crop.cri.nz

Abstract. There are a number of ways to retrieve and analyse data already in a database however it can be more difficult to enter data which has originally been stored in different sources and formats. My master's thesis focuses on investigating a generic, platform independent way to simplify the population of databases. The proposed solution is a Web service based system which will supply essential data management functionality such as inserting, updating, deleting and retrieval of data. These functions will allow developers to easily customize their own data loading applications depending on local data sources, data formats and users' requirements. Currently, we are designing and implementing a data management Web service prototype in order to evaluate how it can facilitate developers to build client applications to load original data from multiple data sources and applications.

Keywords: Data management; Data loading; Web services

1 Introduction

Databases often need to be accessed from a variety of different applications, so there are many ways to retrieve data from the databases. Some data providers supply data extracting Web services in order to allow users to create new client applications [1]. For example, developers can build a data analysis application which obtains data from a public database via a Web service.

Where we have a complex database which is designed for a large community of users there are often problems with data entry. The original data may be stored by different types of users in a number of different ways. There are number of difficulties with loading such data into a database: First, the target database structure might be different from the original data structure, so the original data can not be imported directly to the target database tables. Second, the original data formats may be different from the target database. Third, the original data may come from a variety of sources, applications, and platforms such as text documents in a pocket PC, Excel

spread sheets on a Windows OS, database tables on a Linux OS or statistical applications developed in different programming languages.

These issues can cause difficulties when a diverse community of users needs to populate a large shared database. To create applications for inserting data, developers have to understand the database schema which may include several tables, triggers, sequences, database functions and referential integrity constraints. Also, the original data will need to be transformed into a format that is able to be inserted into the various tables in the database. Assisting users to transform data for inserting into the target database is a critical and complicated process. For example, it is necessary to consider operations such as reformatting data types, handling duplicate records, and checking for invalid data. It is difficult to provide all this functionality in a way that can satisfy different users' requirements, because the original data sources and formats can be very different in each organization. Data loading software which is suitable for one group of users, may be inappropriate for other clients at different sites.

Our solution proposes separating the data loading functions from the local applications. For instance, the input data checks, data transformation (separating data into the appropriate tables) and error handling are common procedures for all users involved in data loading. If a cross platform software component is provided to encapsulate these common procedures then end user developers can more easily tailor their own applications at the client end. We look at how Web services can address these issues. Designers could provide a data loading Web service which encapsulates the complex data model and provides the common data loading processes (input data checks, data transformation and error handling) through a simple programming interface. A Web service is a cross platform technique which allows original data to be passed from any data source by XML messages via HTTP protocol.

2 An example of a large database

As an example of a large database with a diverse community of users we look at Germinate [2]. Germinate is an international database system for storing plant genetic resources and related information. It is a data integration system which contains plant description information, and plant genotype and phenotype data. The Germinate data model has been published for the community of users by Scottish Crop Research Institutions (SCRI). The New Zealand Institute for Crop & Food Research Limited (CFR) is interested in storing their plant data for crops such as onions and oats in Germinate. Currently the data at CFR is stored in a variety of different sources, such as Excel spreadsheets, Access databases, and other applications, and there are difficulties with loading it into Germinate. Each user has to transform their data to get it into the appropriate format for loading in Germinate. We will investigate a Web service as a way to create a data management application that will facilitate loading data from various sources.

3 Data management system overview

In this work, we attempt to centralize the data loading functions as much as possible, so that the individual applications for each client can be smaller and quicker to develop. We propose using a data management Web service to help developers communicate with Germinate. The Web service will be accessible from any programming language via the network. For example, client implementations for various tools (like Access, Excel or Web clients) will be able to easily use the Web service to interact with the database. In order to minimize the efforts of client applications developers, we will also develop client side toolkits which will provide simple functions to help developers invoke the Web service. Currently, we are developing VBA client side toolkits for Microsoft Excel and Access solutions, but the toolkits can be provided in many languages such as Java, C# or PHP depending on the particular client application.

The proposed system (Fig. 1.) includes three modules: (1) User interface (2) Client side toolkit (3) Data access Web service. This structure means that developers need only add a few lines of code to pass the relevant parameters from the user interface to the toolkit in order to call a function of the data access Web service.

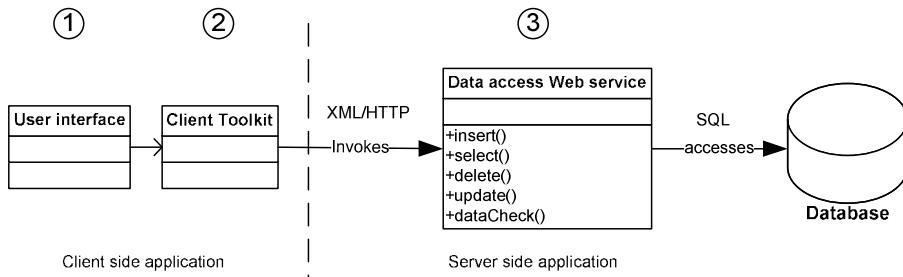


Fig. 1. Overview of data management system architecture(circled figures indicate each module)

The Web service acts as a data access component, which can hide the complex data model and SQL query operations [3]. The service provides essential functionality to interact with the database such as inserting, selecting, updating and deleting data operations.

Inserting data is the most complex operation. For example a typical insertion operation at CFR will include about 20 pieces of information which need to be inserted into up to 18 different tables in Germinate. Our Web service insert function will receive the information from client applications and direct the data into the correct Germinate tables using SQL commands. The insert data function will allow users to supply all relevant data for entry as a unit, so they do not need to worry about which table each field of data should go into. In addition, before loading the data, the Web service will check the user's input data to ensure integrity. For example, the function will check for duplicate records, incorrect data formats (e.g. date formats) and invalid inputs (e.g. wrong institution codes). If invalid data is being loaded, the function will return error messages to client applications.

The client toolkits (Fig. 2) can be incorporated into a variety of different user interfaces to communicate with the Web service. For example, a PHP toolkit could be

developed for use with Web data entry interfaces, and a VBA toolkit could be built to enable batch processes such as submitting several records from an Excel worksheet. The developer can provide a GUI to allow users to enter data and then pass this data to the client toolkits. After that, the toolkit will transfer the data as parameters of an XML request, which calls the appropriate data access functions of the Web service. Whether the transaction succeeds or fails, the web service function sends understandable response messages to the client applications.

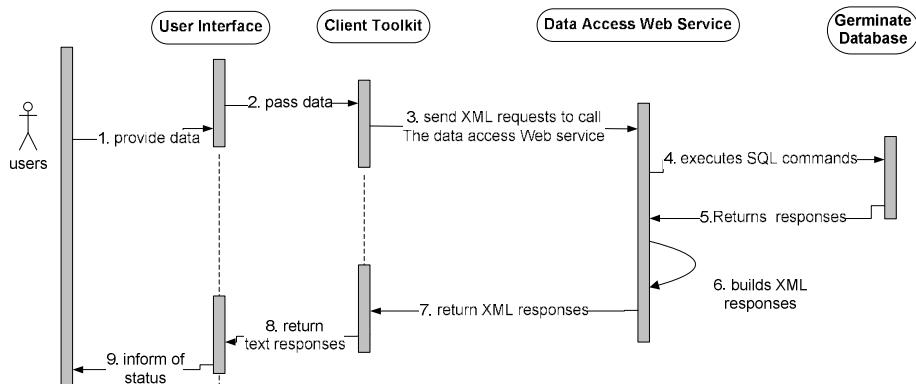


Fig. 2. Invoking the Web service from a user interface via a client toolkit

4 Conclusion and future work

This paper presents a Web service based data management system, which allows client applications to be easily customized for local data sources, formats, data arrangements and user requirements. Client toolkits can be supplied for a variety of platforms to enable users to access the Web service in a convenient and transparent manner. For example, they will be able to simply load data from spreadsheets, databases and other applications directly into a database via the Web service.

In the future, we may look at other Web services issues such as performance and security. For example, we may attempt to optimize the Web service to reduce XML parsing time, and improve the Web service security to provide a robust system via the Internet.

References

1. Steiner, J Jeffery; Minoura, Toshimi & Xiong, Wen WeatherInfo: a Web-based weather data capture system American society of agronomy j. 97:633-639 (2005)
2. Lee, Jenifer (2005) GERMINATE 1.8.4 Document Retrieved March, 2006 from http://germinate.scri.sari.ac.uk/graphics/germinate/distribution1.8/GERMINATE%201_8_4%20description.doc
3. Nock, Clifton (2003). Decoupling Patterns. Data Access Patterns: Database Interactions in Object-Oriented Applications (pp180-193) Boston, MA , USA: Pearson Education

Knowledge Management Literature Review

Yulong Gu¹

¹ Department of Computer Science, University of Auckland, New Zealand
yqu029@cs.auckland.ac.nz

Abstract. Organizational research on Knowledge Management (KM) has studied the nature of knowledge, the scope of KM, the factors and mechanisms that affect KM outcomes, as well as the theoretical KM frameworks. This paper discusses the implications of past KM studies and identifies nine significant contextual, cultural, structural, managerial, cognitive and technological factors that may impact the overall KM outcomes from a KM initiative or project.

1 Introduction

Knowledge is a mix of framed experience, values, contextual information and expert insight [1]. Polanyi classified human knowledge into “explicit” and “tacit” knowledge [2, 3]. The different views of knowledge lead to different perceptions of **Knowledge Management (KM)** [4]. For example, if knowledge is viewed as an object, or is equated with information access, then KM should focus on building and managing knowledge stocks. If knowledge is a process, then KM focus is on knowledge flow and the processes of knowledge creation, sharing, and distribution. A third view of knowledge is an organizational capability, then KM centers on building core competencies, understanding the strategic advantage of know-how, and creating intellectual capital [5]. This paper establishes part of the theory base for an ongoing PhD project in the Department of Computer Science at the University of Auckland. The title of this project is *Human Genetic Variation Knowledge Management*. This paper reviews general KM issues that have been addressed by past organizational literature.

2 KM approaches in practice

Past KM practices have taken mainly two approaches or strategies: product-centric KM and process-centric KM, reflecting first and second KM focuses. But there is no successful experience reported as capability-centric approach (type three KM) – to build core competencies and to create intellectual capital [5].

Product-centric KM views knowledge as an objective asset to be codified, stored and managed like other organizational assets [6, 7]. It relies on the transformation of implicit or explicit knowledge from employees’ heads into written information in documents and the subsequent management of these documents [8, 9]. On the other hand, **Process-centric KM** views knowledge as residing with a person and/or a busi-

ness process. There is no attempt in this strategy to formally capture and store knowledge; instead, it provides pointers to individuals who are likely to have the relevant expertise [6]. Process-centric KM applications include database of experts, decision aids and expert system, workflow management system, groupware, systems supporting Community of Practice, and ‘hardwiring’ of social networks [5, 10].

3 Theoretical KM frameworks

KM models are proposed to guide KM studies and to provide best practice, e.g. the theoretical framework for organizing research on KM [11], KM effectiveness model [12], Knowledge Management Systems (KMS) Success Model [13], etc.

The contextual properties of units (e.g., individual, group, and organization), relationships between units, and the knowledge itself may all affect KM outcomes (creation, retention, and transfer) [11]. Furthermore, three key causal mechanisms (*ability*, *motivation* - rewards and incentives, *opportunity*) may help explain how and why certain contextual properties affect KM outcomes [11]. On the other hand, two organizational capabilities are referred as the “preconditions” for effective KM: (i) the knowledge infrastructure capability (social capital or network of relationship) and (ii) the knowledge process capability (knowledge integration), with the latter being influenced by contingent knowledge tasks, see Fig. 1.

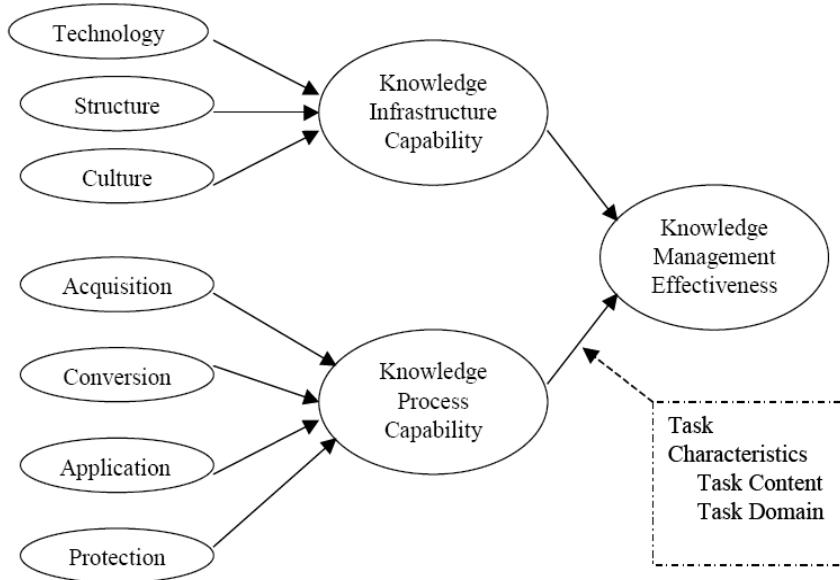


Fig. 1. KM Effectiveness Model [12]

KMS are a class of Information Systems (IS) to manage organizational knowledge and to support knowledge processes [5]. KMS success model was developed from IS

Success Model [14] and regarded the organizational effectiveness as KM outcome [13]. Fig. 2 depicts how the individual's and organization's performance at workplace are improved by using quality KMS.

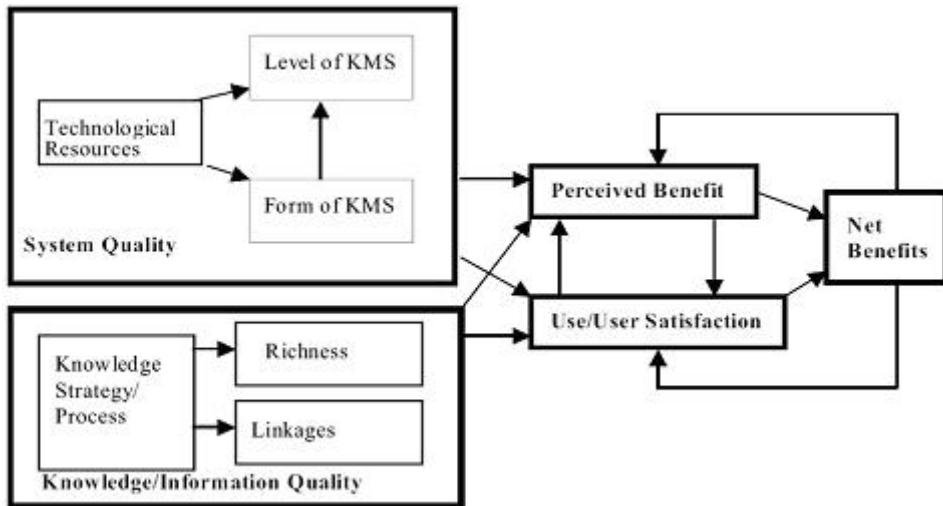


Fig. 2. KMS Success Model [13]

According to above figure, the quality of a KMS contributes to KM outcomes significantly; and it has at least three dimensions: (i) the technological resources – the ability to develop, operate, and maintain a KMS, (ii) KMS form – the extent to which organizational memory and KM processes are computerized and integrated, and (iii) KMS level – the ability to bring past information to bear upon current activities [13].

4 Discussion and Conclusion

From past KM approaches and frameworks, we identify nine categories of contextual, cultural, structural, managerial, cognitive and technological issues that are critical for the KM efforts in an organization as: I. KM context [11], II. KM process [11], III. knowledge process capability [12], IV. contingent task characteristics [12], V. technology/system quality [13], VI. knowledge/information quality [13], VII. perceived benefits and use/user satisfaction [13], VIII. knowledge infrastructure capability [12] and IX. KM outcome [12, 13]. Further exploratory and empirical studies will offer more insights on the significance of the nine constructs in an organizational KM project and the relationships among these issues during the project implementation. In the case of our project, by applying these nine constructs, we are trying to identify the context of human genetic variation knowledge management studies, the predispositions and factors that may impact KM outcomes, and important KM processes in the genetics domain. We are also trying to understand the significance of the relationships among these issues. By taking an IS approach, our project will eventually point

a way for improved capture and dissemination of human genetic variation knowledge from routine genetic research activities to contribute to the global genetics knowledgebase.

Acknowledgement

I'm in debt to many people for their contribution and inspiration to this ongoing project, especially to my two mighty supervisors: Prof. James Warren and Dr. Alexei Drummond. Thanks very much for your coolest SUPER VISIONS!

References

1. Davenport, T.H., Prusak, L.: *Working Knowledge: How Organizations Manage What They Know*, Cambridge, MA: Harvard Business School Press, 1997.
2. Nonaka, I.: A dynamic theory of organizational knowledge creation. *Organization Science*, 5, 1, (1994) 14-37.
3. Polanyi, M.: *The Tacit Dimension*, London: Routledge & Kegan Paul, 1966.
4. Carlsson, S.A., El Sawy, O.A., Eriksson, I., Raven, A.: Gaining competitive advantage through shared knowledge creation: in search of a new design theory for strategic information systems. In *Proceedings of the Fourth European Conference on Information Systems*, Lisbon, (1996), pp. 1067-1075.
5. Alavi, M., Leidner, D.E.: Review: knowledge management and knowledge management systems: conceptual foundations and research issues. *MIS Quarterly*, 25, 1 (2001), 107-136.
6. Dennis, A.R., Vessey, I.: Three knowledge management strategies: knowledge hierarchies, knowledge markets and knowledge communities. *MIS Quarterly Executive*, 4, 4, (2005).
7. Schultze, U., Leidner, D.E.: Studying knowledge management in information systems research: discourses and theoretical assumptions. *MIS Quarterly*, 26, 3, (2002), 213-242.
8. Bossen, C., Dalsgaard, P.: Conceptualization and appropriation: the evolving use of collaborative knowledge management. In *Proceedings of the 4th decennial conference on Critical computing: between sense and sensibility CC '05*, (August 2005), pp. 99-108.
9. Sambamurthy, V., Bharadwaj, A., Grover, V.: Shaping agility through digital options: reconceptualizing the role of information technology in firms. *MIS Quarterly*, 27, 2, (2003), 237-263.
10. Brown, J.S., Duguid, P.: Organizing knowledge. *California Management Review*. 40, 3, (1998), 90-111.
11. Argote, L., McEvily, B., Reagans, R.: Managing knowledge in organizations: An integrative framework and review of emerging themes. *Management Sci.* 49(4 Special Issue), (2003), 571-582.
12. Lindsey, K.: "Measuring Knowledge Management Effectiveness: A Task-Contingent Organizational Capabilities Perspective," Eighth Americas Conference on Information Systems, pp. 2085-2090, 2002.
13. Jennex, M.E., Olfman, L.: A knowledge management success model: an extension of DeLone and McLean's IS success model. In *Proceedings of Ninth Americas Conference on Information Systems*, (August 2003).
14. DeLone, W.H., McLean, E.R.: Information systems success: the quest for the dependent variable. *Information Systems Research*, 3, 1, (1992), 60-95.

Population-based Monte Carlo

Sergio Hernandez

Victoria University of Wellington
School of Mathematics, Statistics and Computer Science
Wellington, New Zealand
sergio@mcs.vuw.ac.nz

Abstract. Population-based Monte Carlo simulations can be used to extend Monte Carlo (MC) methods for solving complex multimodal posterior distributions that can arise in the context of inference over a multimodal distribution using an interacting particles system.

This paper presents preliminary results that shows a population-based Monte Carlo strategy can help the large number of samples required for performing inference with the standard implementations of MC methods, while introducing better adaptivity and exploring capabilities.

1 Population-based Monte Carlo

Using parallel Monte Carlo simulations has the drawback that many samples are discarded for being far from the likelihood modes, and the mixing times can be . One approach that improves this situation is adaptively modify the direction of the new samples upon its performance. One of these approaches is the Adaptive Direction Sampling algorithm, where each sample changes its direction using global statistics of the population upon the so called snooker moves [1].

Population Markov Chain Monte Carlo (PopMCMC) methods has been proposed in [2] as a way for combining evolutionary algorithms with MCMC. Although these developments are closely related to previous works that makes use of several Markov Chains like Parallel Tempering [3], the PopMCMC framework provides more freedom in terms of the kind of moves and interactions for the Markov chains with an immediate gain in diversity and mixing. The Evolutionary Monte Carlo (EMC) algorithm combines tempering with crossover operations, that enhances information sharing between the Markov chains [4]. Further developments in evolutionary optimization for parallel Markov chains has been done in [5].

The method proposed in this paper shares some similarities with adaptive Monte Carlo methods like Population Monte Carlo [6] and Sequential Monte Carlo Samplers [7] in the design of a population-based move for parallel simulation. Particularly, we introduce a population-based stochastic optimization method for perturbing the Markov chain in an interacting simulations setting.

1.1 Random Walk Metropolis-Hastings

The Metropolis-Hastings algorithm is an algorithm that uses Markov chains for sampling from a probability distribution $\pi(\cdot)$ that at least is known up to a constant Z .

$$\pi(x_{1:k}) = Z^{-1} \exp(-h(x)) \quad (1)$$

The Metropolis algorithm starts with an initial setup x_0 and proposes a new state using a transition kernel $T(x_{1:k}^i, x_{1:k}')$. The new states converges to the target distribution because it's used as the limiting distribution for the Markov chain [8].

The new state is accepted with probability

$$\alpha = \min\left(1, \frac{\pi(x_{1:k}' T(x_{1:k}^i, x_{1:k}'))}{\pi(x_{1:k}^i T(x_{1:k}', x_{1:k}^i))}\right) \quad (2)$$

The transition kernel can have any form and the stationary distribution will remain fixed, but if one considers a symmetrical proposal that keeps $T(x_{1:k}', x_k^i) = T(x_k^i, x_{1:k}')$, the acceptance probability reduces to:

$$\alpha = \min\left(1, \frac{\pi(x_{1:k}')}{\pi(x_{1:k}^i)}\right) \quad (3)$$

In practice it's difficult to find a good transition kernel, so the random-walk Metropolis is a simple and straightforward implementation. The transition kernel is chosen as a perturbation of the previous state by means of additive zero-mean Gaussian noise η .

$$x_{1:k}' = x_{1:k}^i + \eta \quad (4)$$

1.2 Population-based Metropolis-Hastings

Running multiple independent chains in parallel with over dispersed initial states can be worthless because of the burn-in time and the autocorrelations of the chains, so a population-based proposal take advantage of the interactions of the chains in order to improve the mixing and the autocorrelations of the chains.

A particular case of an EMC is the use of snooker moves [1], where the population members are moved towards each other using the global fitness information of the population. The Particle Swarm Optimization [9] (PSO) algorithm can be used for introducing cooperative dynamics in the interacting Markov chains. Swarm intelligence and specifically PSO has been used successfully in several multi-objective optimization problems. The algorithm is inspired by the social behavior of bird flocks and makes use of an analogy whereby each particle flying through the search space represents a possible solution to the multi-objective optimization problem. The position of each one of the particles is influenced by both the best position achieved by itself (own experience) and the position of the

globally best weighted particle (social component). The basic algorithm uses the position of the best particle, but smaller neighbourhoods can be used in more complex scenarios.

The PSO adaption is done by iterating a velocity measure $v_k^{i,j}$ for each particle, using $x_k^{i,j}$ as the best position of particle i and $x^{j,j}$ as the global best position in the j -th iteration.

The velocity equation of particle i at time k can be written:

$$v_k^{i,j+1} = v_0 + c_1 r_1(x_k^{i,j} - x_k^{i,j}) + c_2 r_2(x^{j,j} - x_k^{i,j}) \quad (5)$$

c_1 and c_2 are acceleration constants, r_1 and r_2 are momentum constants and v_0 is an inertia constant. All these constants are dependent on the problem domain and can be related to the model noise. Each iteration of the PSO uses the weights of the previous iteration, via the equation:

$$\begin{aligned} x_k^{i,j} &= \text{argmax}_{x_i}(w_{k-1}^i p(y_k | x_k^{i,j})) \\ x^{j,j} &= \text{argmax}_{x_j}(w_{k-1}^j p(y_k | x_k^{i,j})) \end{aligned}$$

with the position of the particle being updated using:

$$x_k^{i,j+1} = x_k^{i,j} + v_k^{i,j+1} \quad (6)$$

The above equations defines an adaptive random walk Metropolis-Hastings that improves the choice of the update. The new acceptance ratio for the updated particles positions:

$$\alpha = \min\left(1, \frac{q(x'_k | x_{0:k-1}^i, y_{1:k})}{q(x_k^i | x_{0:k-1}^i, y_{1:k})}\right) \quad (7)$$

The following example shows the result of using population-based Metropolis-Hastings for a simulating a mixture of 2 Gaussian distributions. The simulation was done with 200 chains, starting from over dispersed points.

2 Conclusions

This work shows the basic implementation of a population-based optimization algorithm for a self-adapting iterated importance sampling strategy. The algorithm requires basic tuning of the parameters and achieves fast convergence and is also able to explore the state-space when the starting points are far from the modes. While standard non-interacting MCMC methods can be used for improving SMC, additional work must be done in order to propose model jumps when the number of modes is unknown.

Further work must be done in proving the convergence of the method proposed, the preliminary results shows that the algorithm can effectively be used in dynamic environments like tracking an unknown number of objects, where standard Bayesian methods fails to achieve a fast estimate in real time under severe noise conditions.

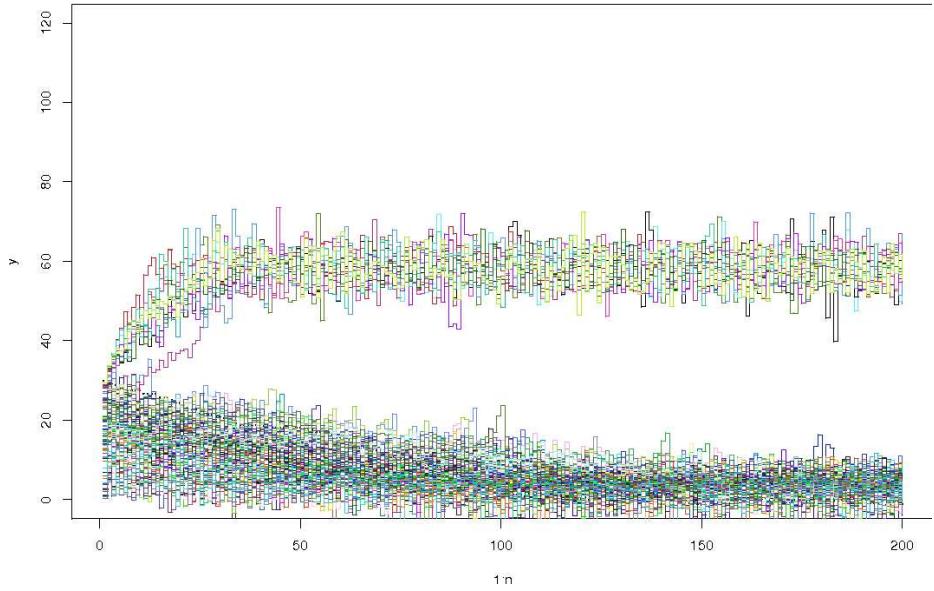


Fig. 1. Population-based Metropolis Hastings Sampling.

References

1. Gilks, W.R., Roberts, G.O.: Adaptive direction sampling. *The Statistician* **43**(1) (1994) 179–189
2. Laskey, K., Myers, J.: Population Markov Chain Monte Carlo. *Machine Learning* **50**(2) (January 2003) 175–196
3. Geyer, C., Thompson, E.A.: Annealing Markov Chain Monte Carlo with applications to ancestral inference. *Journal of the American Statistical Association* **90**(431) (September 1995) 909–920
4. Liang, F., Wong, W.: Real-parameter evolutionary sampling with applications in Bayesian mixture models. *Journal of the American Statistical Association* **96**(454) (2001) 653–666
5. Cajo Ter Braak, J.F.: A Markov Chain Monte Carlo version of the genetic algorithm differential evolution: easy bayesian computing for real parameter spaces. *Statistics and Computing* **16**(3) (September 2006) 239–249
6. Cappe, O., Guillin, A., Marin, J., Robert, C.: Population Monte Carlo. *Journal of Computational and Graphics Statistics* **13**(4) (January 2004) 907–25
7. Del Moral, P., Doucet, A., Jasra, A.: Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **68**, pages = (26) (June 2006)
8. Gilks, W.R., Richardson, S.: *Markov Chain Monte Carlo in Practice*. Chapman Hall, New york (1996)
9. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proc. IEEE International Conference on Neural Networks. (1995)

Accelerate process of tracing back DoS/DDoS attack¹

Alireza Izaddoost

Department Of Communication Technology and Network
aizaddoost@yahoo.com

Under Supervision Dr. Mohamed Othman
Putra University of Malaysia (upm)

Abstract. One of the significant groups of security endangements in the internet to day is DoS/DDos attacks. Some traceback approach have been proposed to trace the spoofed source of attack . One of these methods is the Intention-driven iTrace and this method is based on ICMP traceback (iTrcae). With this method it will be possible to generate effective iTrcae messages . Effective iTrace messages can provide enough information to the victim to reconstruct the path of the source of attack faster .In our proposed model , we want to show that if we consider incoming packets routed to the victim and modify intention-driven iTrace model to generate iTrace message base of incoming rates ,we can accelerate the chances for effective iTrace message in Intention-driven model.

Keywords: Internet Security; Denial of Service; Distributed Denial of Service; ICMP Traceback ,Network Research

1 Introduction

In a denial-of-service (DoS) attack ,a lot of malicious packets will be sent toward one or many victims .In Distributed DoS attack from multiple points it happens and user requests are disrupted and services are denied to the legitimate users . In this kind of attack , hacker usually sends packets with spoofed source IP address so that the victim can not trace source of attack path. DoS/DDoS is one of the major threats and among the hardest security problem in today's Internet [1]. Both DoS and DDoS attacks exploit the vulnerability of today's network infrastructure. In the past several years, DoS/DDoS attacks have increased in frequency, severity, and sophistication. In 2002 CSI/FBI Computer Crime and Security Survey results show that over 55% of respondents have been hit by DoS/DDoS attacks, which caused total losses of \$18,370,500 in the first quarter of 2002. The widely known February 2000 DoS attacks, bringing down commercial websites such as Yahoo, eBay, Amazon, CNN, and ZDNet, are a painful reminder that DoS and DDoS attacks have become some of the severest network security problems.

¹ This research is under process

The objective of IP traceback is to identify the actual origin of attack packets. IP traceback techniques can be defined in following three groups [2].

1) *Packet marking*: Routers mark path information in packets as they pass through the Internet. Victims reconstruct attack paths from path fragments embedded in received packets.

2) *Messaging*: Routers probabilistically send ICMP messages, which has both IP address and MAC address of upstream and downstream routers . Victims build attack paths from received ICMP messages.

3) *Packet Digesting*: Routers store audit logs of forwarded packets to support tracing attack flows. Victims consult upstream routers to reconstruct attack paths.

In this paper we want to show a proposed traceback model which falls in Intention-driven iTrace , one of the messaging techniques, which will be discussed in detail in later sections

This paper is organized as follows: Section 1 gives an introduction to the paper. Section 2 highlights the ICMP related works . Section 3 describes a more effective ICMP traceback messages. Section 4 we describe our proposed model to make effective ICMP message faster to the victim .

2 background

ICMP traceback (iTrace) proposed in IETF .In this case , when forwarding packets, routers can, with a low probability (1/20000),generate a raceback message that is sent along to the destination. With enough Traceback messages from enough routers along the path, the traffic source and path can be determined [3].An ICMP tarceback message has pair IP address and MAC address of both upstream and down stream routers. An iTrace collector (for example IDS) with enough iTrace packets can reconstruct attack source path.

In ICMP traceback with cumulative path [4] , they use various solutions to encode the path traversed by the attack packets into the iTrace message .Instead of encoding the path information in the IP packet header , they use iTrace message to store path information and then send it to the next hop router then this router will process the iTrace message to see that if IP packet and iTrace message has same path or not. Then they decide to send new iTrace message or append it to original IP.

An other approach is Intention-driven ICMP traceback [5] .A router conceptually has two tables : routing information table and a packet forwarding table .Community attribute (32 bit unsigned integer) in BGP routing information exchange protocol is proposed to distribute the intention bit value . A downstream BGP router can pick it up and update the intention value .

3 Effective iTrace packet

IETF iTrace working group has introduced ICMP traceback (iTrace message) toward victim with probability 1 over 20k [3] . In this case we have huge traffic near the victim , and this probability may be sufficient to give enough information to iTrace collector to reconstruct the attack source path. But near source of attack which attack rates is not high , selected packet to create iTrace message may not be related to the victim and provided information by this iTrace packet is not useful to reconstruct the source path .

In intention-driven model [5] ,they introduced 2 module to increase the chance of useful iTrace packets. .Decision module and iTrace generation module (fig.1)

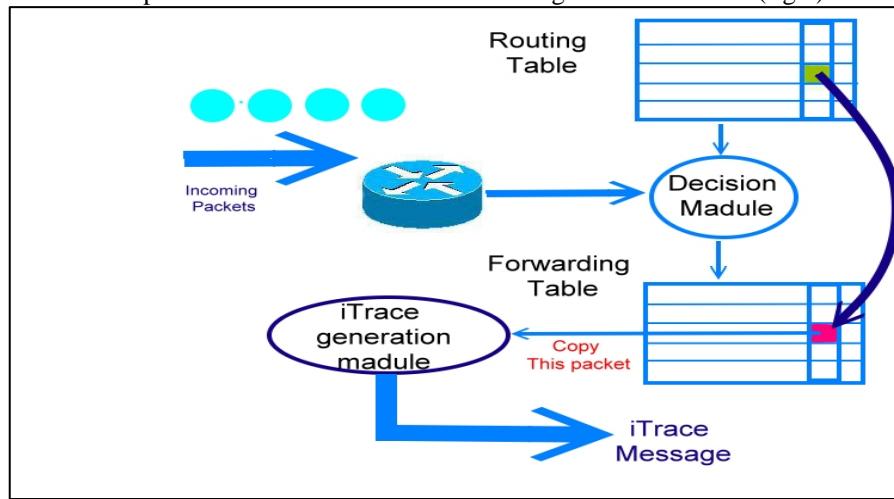


Fig.1 Intention –driven iTrace message

The decision module determine which “entry” in the packet forwarding table should be target for iTrace . Base of this decision one special bit in the packet forwarding table will be set to 1 and the next data packet using this particular forwarding entry will be chosen as an iTrace message, next this chosen will be preceded by the iTrace “generation” module and a new iTrace message will be sent. In this model the selected packet has more effect and can provide enough information to the victim to reconstruct attack source path .

4.Accelerate effective iTrece message

Intention-driven method can increase the chance of useful iTrace packet but we have to consider a problem here ; selected packet to make iTrace from the specific entry in this model may not be belong to the attacker because it is possible a legitimate user also use the victim service with same routing table pattern an same forwarding table.

To solve this problem we proposed a new model base of ID-itaceback model. We assume attack rate is more than normal at further router from the victim. we want to add one more module .In our model when decision module dedicate forwarded interface , generated module first checks the incoming router interfaces to find which interface has the highest rate packets towards the victim , then select the packet to generate iTace message which has to be the base of this incoming interface (Fig.2)

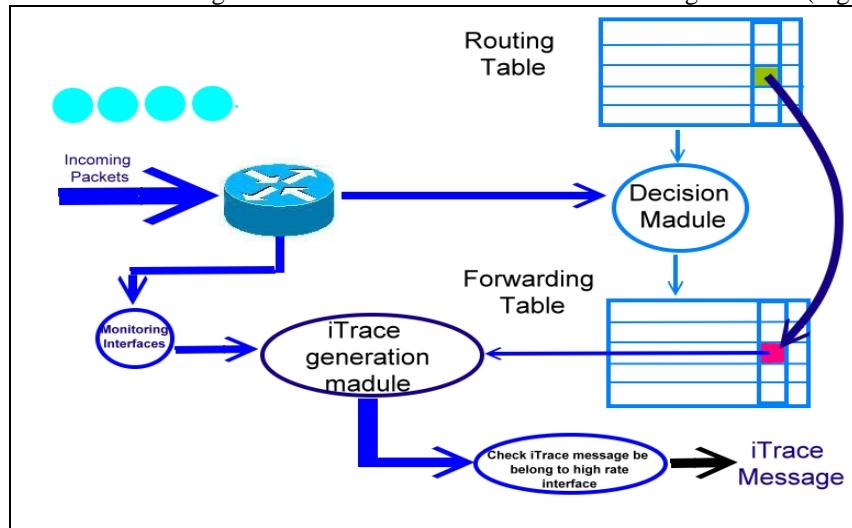


Fig.2 Proposed Model

This selection can increase the chance of effective iTace message specially in the router far from victim when the attack rates is near to normal rates .

We are now in implementation stage in omnet++ [6] simulator for performance measurement .Quantity measurement with checking percentage of effective packets vs. background traffic and quality measurement with definition of value (accumulate iTace message value over time) , the expected result is that both measurement show us better result in compare with pervious models.

References

1. C. Douligeris, A. Mitrokotsa, "DDoS attacks and defense mechanisms: Classification and state-of-the-art," Computer Networks: The International Journal of Computer and Telecommunications Networking. April 2004. vol. 44, Issue 5,pp. 643 – 666.
2. Amin, S.O., Choong Seon Hong, "On IPv6 traceback" Advanced Communication Technology, 2006. ICACT 2006. The 8th International Conference Volume 3, 20-22 Feb. 2006 Page(s):5 pp. 2139-2143,2006.
3. S.M. Bellovin, "ICMP Traceback Messages," Internet Draft, draftbellovin-iTrace-00.txt, Mar. 2000.
4. Henry C.j Lee, (2003), "ICMP Traceback with cumulative path" , Springer Berlin , Volume 2836/2003 ,pp. 124-135 ,2003.
5. Mankin, et al., (2001)"On Design and Evaluation of 'Intention-Driven' ICMP Traceback," Proc. IEEE Int'l Conf. Computer Comm. and Networks, IEEE CS Press, 2001. pp. 159-165.
6. Network Simulator omnet++ : www.omnetpp.org

Proposed Model for Security Enhancement by Algorithm Permutations

Abhilasha Keshariya

Department of Information Technology,
Maulana Azad National Institute of Technology, Bhopal, India
abhilasha_keshariya@yahoo.co.in

Abstract. Data security is an important consideration for technology manager. Various encryption methods have been introduced and are widely deployed for protecting sensitive information. Every encryption method is based on an algorithm and no matter how complicated the algorithm is it is always susceptible to brute force attack since it follows the same monotonous procedure. This paper attempts to introduce a randomness function in the selection of encryption algorithms. We propose an integrated *Giant algorithm* composed of various algorithms which are selected by a secret key. We have also introduced the concept of an algorithm bank and defined Hybrid and Multilevel permutation encryption schemes.

Keywords: Internet security, Encryption algorithms, Algorithm bank

1. Introduction

In the present computer era, security has become a prerequisite for communicating and storing sensitive information. To provide this security various cryptosystems have been defined [1]. Algorithms for encryption (no matter how complicated they are), are monotonous in nature. It is this monotony that makes them susceptible to cryptographic attacks. By introducing randomness in the selection and usage of algorithms, it is possible to build more secure cryptosystem [2]. This scheme has an advantage over introducing randomness only in secret keys, as the hacker will not only have to guess the correct key but also the correct algorithm, whose permutation steps are decided at the real time. In other words, the secret key used for encryption is also used to decide the sequence of the encryption algorithms. Hence, for each key there is a unique sequence of encryption algorithms that cannot be generalised [3].

2. Proposed Model

Figure 1 depicts the proposed model and shows the message flow for encrypting a plain text message. We have extended the generic cryptosystem model to introduce a new entity, Algorithm Bank, described in section 2.2.

2.1 Message Flow

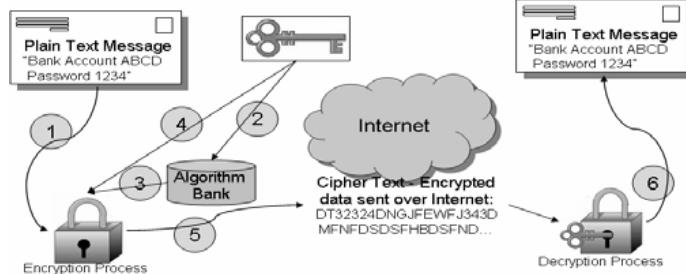


Figure 1: Proposed Model. Message flow to for encryption of a plain text message using an algorithm and a key

To encrypt a message (figure 1), a key is used to generate a sequence of algorithms from the Algorithm bank (2). The generated algorithm sequence (3) could be applied to individual blocks of message (liner block-wise permutation) or a set of algorithms either applied in a sequential pattern (multilevel permutation) or in a hybrid permutation scheme where different block of data are encrypted by different sets of algorithms, described in detail in section 4.1.

The key used for generating the algorithm sequence is used as the secret key (k) for encryption process (4). The generated cipher text is then transmitted through an insecure channel (5) and decrypted at the receiving end (6).

2.2 Algorithm Bank and Transformation Function

An algorithm bank is a collection of encryption algorithms. There are separate banks for encryption algorithms and their corresponding decryption algorithms. Each algorithm has a same fixed length Unique Identification Key (UIK). For encryption, when the secret key (k) is entered, a function $f_{K2A}(k)$ transforms the secret key k to a number string of variable length, which may be more, less or the same size as that of the original key.

This number string corresponds to the order in which the algorithms are applied for encryption. For example: for an algorithm bank of 10 algorithms indexed from 0-9 (figure 3) if the key entered is 123579 and $f_{K2A}(k) = 2k$, then we have a number string 247158 and algorithms indexed as 2, 4, 7, 1, 5 and 8 are selected for encryption through one of the three permutation methods suggested in section 4.2

At the decryption end the same secret key and the same $f_{K2A}(k)$ is used to generate the number string and then decryption algorithms are applied in the reverse order.

3. Proposed Architecture

Figure 2 discusses our proposed architecture. The algorithm engine consists of primarily the encryption and decryption modules, a scheduler and a random generator that has the transformation function of secret key to number string.

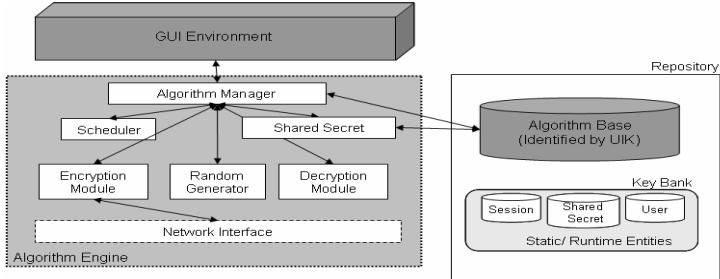


Figure 2: Proposed Architecture

- (i) Algorithm manager signals the Scheduler to select a transformation function, the mapping of algorithms to UIKs and a permutation scheme.
- (ii) Algorithm manager then sends this transformation function along with the secret key to the Random Generator.
- (iii) Random Generator converts the secret key into a string of numbers and sends back to the Algorithm manager.
- (iv) Algorithm manager then initiates encryption through Encryption Module that uses algorithms selected by the Repository in the order decided by the Scheduler.
- (v) The encrypted message is then sent to the user B through network interface.

4. Introducing degree of *Randomness*

4.1 Function variation $f_{K2A}(k)$

Changing the parameters or even constants of the function $f_{K2A}(k)$ results in the change of the number string generated from the secret key k . Also the function $f_{K2A}(k)$ may be altered in such a way that it may even reduce the length of number string or keep it constant. It depends on the function $f_{K2A}(k)$ for which algorithms are used and with what frequency.

4.2 Algorithm Permutations

Different permutation schemes on encryption sequences have been proposed to further enhance the security provided by the system. A *Giant* algorithm G consisting of n different algorithms is used. We may use three types of permutations:

1. Linear Block-wise Permutation: The plaintext is divided into a number of blocks and any algorithm is applied to any block based on some rules decided by the secret key.
2. Multilevel Permutation: In this scheme, the algorithms are applied in nested fashion, one after the other, on the same text message. The order of nesting is deter-

mined by the secret key and thus, the cascading of algorithms is dynamic and random for each key.

3. Hybrid Permutation: This approach is the hybrid of the two approaches discussed above. Here, for each of the n blocks of the plain text, all the selected algorithms are applied but in different sequences of the possible permutations.

5. Prototype Implementation

We have implemented a prototype system (figure 3) using 10 algorithms in the algorithm bank. The function $f_{K2A}(k) = 3k$ was used for the sake of simplicity. The file with the plaintext as shown below was encrypted using an 8 digit secret key.

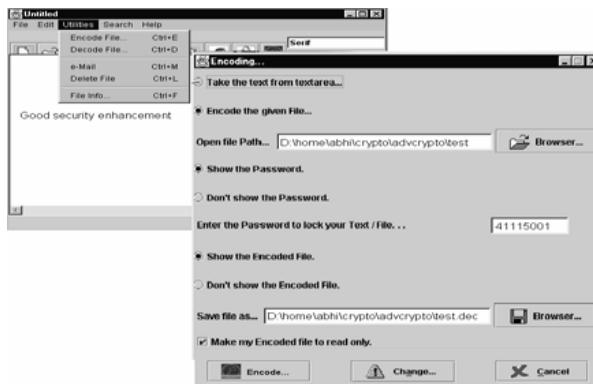


Figure 3: Prototype Implementation

6. Conclusion

In this paper, we have introduced the concept of a randomly permuted *Giant* algorithm which is composed of various algorithms selected by a secret key which is also used for encryption. We have also proposed a new model, introducing components as Algorithm bank and transformation functions and defined different permutation encryption schemes to improve the overall randomness in the proposed system. Further, we have also proposed the architecture for the model and designed a prototype implementation.

References

- [1] A. Beimel, S. Dolev, "Buses for Anonymous Message Delivery," Journal of Cryptology 16(1), 2003
- [2] B. Schneier, "Applied Cryptography," John Wiley and Sons, Inc., 1996
- [3] B. Moller, "Provably Secure Public-Key Encryption for Length-Preserving Chaumian Mixes," Proceedings of CT-RSA 2003, Aril 2003

Linux Kernel development

Ian McDonald

University of Waikato
ian.mcdonald@jandi.co.nz

Abstract. This paper focuses on the process of development for the Linux kernel. This topic is approached both from a technical viewpoint and also the interactions with the open source community. The aim of the paper is for readers not to have the same difficulties the author experienced!

Keywords: Operating Systems

1 Introduction

This paper discusses the social aspects of kernel development, followed by tips for building a kernel. The paper then focusses on releasing code and lastly, but not least important, testing and debugging.

The paper highlights some of the lessons learned by the author and the methodology used in developing within the kernel.

2 Building a kernel

2.1 Maintaining a source code tree

Keeping in synchronisation with the Linux kernel source code tree takes a non-trivial amount of time. It is necessary to keep synchronised if it is intended to have the code merged into the tree or released as an ongoing codebase.

The Linux source code is maintained in a git [1] tree. Git is a new source code management tool created by Linus Torvalds. Git stores every change as a patch addressed by a SHA1 hash. Developers can make copies of git trees and because each patch is unique it is relatively simple for developers to synchronise their code base with other developers.

For developing in the kernel it is recommended to make a copy of Linus Torvalds' source code tree by issuing a statement similar to:

```
git-clone \
  git://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux-2.6.git \
  ~/linuxsrc/linus
```

and then using `git-pull` to keep the git tree up to date. If the maintainer for the area that is being developed keeps a separate git tree, as most do, then you will need to create a copy of it by issuing a statement similar to:

```
git-clone --reference ~/linuxsrc/linus \
    git://git.kernel.org/pub/scm/linux/kernel/git/davem/net-2.6.git \
    ~/linuxsrc/davem
```

When the above syntax is used it uses the local copy where possible making synchronisation quicker and uses a fraction of the bandwidth.

It is often useful to use a patch management tool on top of git such as stgit [2] or quilt [3]. These tools allow a series of patches to be maintained as a “stack” of patches which can be applied against different or updated trees.

Often a maintainer will “rebase” their tree which means they delete their original tree, make a fresh clone of Linus’ tree and then they will reapply any outstanding patches to that tree. If a maintainer rebases their tree then it is usually not possible to update it using the `git-pull` or `stg pull` command. Presuming the use of stgit, issue the command `stg export` to export the patches (this should also be done before any pull in case of a problem), save the files that are under `patches-branch` in a temporary directory, clone the git tree again to the latest tree and then import the patches into the new tree by `stg import`.

2.2 Use of distcc

To speed up compilation if multiple machines are available then `distcc` [4] can be used which is a distributed C compiler, and then specifying to `make` how many parallel threads to run using the `-j` parameter. For kernel development there are a couple of caveats to be aware of. The same version of the C compiler `gcc` must be installed on the machines that are being used as a compile pool. It is also important to specify only one target on the command line or else the kernel will continue to rebuild from scratch each time the `make` command is issued. For example to build a kernel and prepare for it to be installed the following commands could be issued:

```
make -j6 CC=distcc all
make modules_install INSTALL_MOD_PATH=~/tmp
```

2.3 Other resources

For an introduction to Linux development there are excellent resources available such as [5] and [6]. The use of simple tools should also not be underestimated such as the use of `grep -n -r phrase` to find a symbol in the kernel. The author also maintains a Wiki page at <http://wlug.org.nz/KernelDevelopment> which contains other tips.

3 Releasing code

An often cited mantra in the open source community is “release early and release often” [7]. This concept is useful even prior to code release.

It can be productive to talk about what you are planning to implement. This can be useful as other people can inform you if they are working on similar code as is often the case. It also gives people an opportunity to give feedback on whether the ideas are good or need refining.

Once the ideas have been crystallised it is often useful to release a code as a request for comment (RFC) if the code is not ready to be merged. This, again, allows discussion of the ideas being implemented.

The code will probably go through many, many iterations so if the code is rejected then it is important to work on the areas highlighted rather than becoming disheartened.

When developing code for the kernel it needs to be submitted to the maintainer of the subsystem. The **MAINTAINERS** file in the top level directory of the Linux source code tree contains the list of the maintainers. A posting to the `linux-kernel` list or other kernel mailing lists will usually produce a response on how the code is maintained if it is not clear.

Any code released should always be against the latest tree of the maintainer. One of the biggest reasons that code isn't accepted is that the maintainer cannot use it as it applies against an older version of Linux.

When code is submitted it must be able to be applied without problems by the maintainer. Prior to submitting test each patch with:

```
git-apply --check --whitespace=error < mypatch.diff
```

It is also required to choose a mail client to send the patch which does not alter whitespace such as spaces being substituted for tabs, breaking lines up or sending as HTML. The author uses KMail as he has found this is one of the few that will meet these criteria.

The code and patches should also fit into the existing kernel code style. The **Documentation** directory of the Linux source code contains many useful guides to assist with this.

It is recommended for a developer to submit kernel changes in multiple small patches rather than one large patch. This is so the change can be reviewed in a series of steps which are simpler to read and find bugs in. If multiple patches are provided then part of the change can be accepted, rather than all rejected if one large patch is used.

4 Testing and debugging

It is important that code is well tested before it is released.

There are a number of virtualisation tools available to assist with testing software such as `qemu` [8], `UML` [9], `VMWare` [10] and `Xen` [11]. These tools can make testing easier if computers with sufficient processing power are used. Any software developed should also be tested on computers natively as well as this can uncover separate bugs.

There are debugging tools available that can be used on the kernel such as `gdb`. The simplest debugging tool to use for timing dependent code is the `printk`

statement in the code which acts like a `printf` statement, except the output goes to the kernel logging daemon. If it is desired to find which patch caused a bug then `git-bisect` can be used to “divide and conquer” the code base.

In the kernel there is also a mechanism for tracing code execution and capturing data called kprobes which allows hooks to be added into any function entry or exit dynamically without any changes being made to the original function. In our research we have taken an implementation of this for monitoring TCP and implemented it for DCCP.

With testing it is often necessary to transfer newly built kernels to multiple machines. This can be automated through a script such as:

```
#!/bin/bash
# syntax m machine_name directory version
H=$HOME
SRC=$H/linuxsrc/$2
VER=$3
rm $H/tmp/lib/modules/$VER/build
rm $H/tmp/lib/modules/$VER/source
rsync $SRC/System.map root@$1:/boot/System.map-$VER
rsync $SRC/arch/i386/boot/bzImage root@$1:/boot/vmlinuz-$VER
rsync -av $H/tmp/lib/modules/$VER root@$1:/lib/modules
```

5 Conclusion

Development in the Linux kernel is more than simply editing code and typing `make all`. It is the hope of the authour that this paper helps more people develop in the Linux kernel by taking into account other considerations and applying these.

References

1. Web: git. <http://git.or.cz/> (Accessed 2006)
2. Web: Stacked git. <http://www.procode.org/stgit/> (Accessed 2006)
3. Web: Quilt patch management tools. <http://savannah.nongnu.org/projects/quilt/> (Accessed 2006)
4. Pool, M., et al.: Distcc: a fast, free distributed c/c++ compiler, 2002-. URL <http://distcc.samba.org>
5. Love, R.: Linux Kernel Development. Second edn. Novell Press (2005)
6. Web: Lxr. <http://lxr.linux.no/> (Accessed 2006)
7. Raymond, E.: Cathedral & the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary. (2001)
8. QEMU, C.: Emulator. URL: <http://fabrice.bellard.free.fr/qemu>
9. Dike, J.: A user-mode port of the Linux kernel. Proceedings of the Annual Linux Showcase. Atlanta, GA, Oct (2000)
10. VMware, I.: The VMWare software package. See <http://www.vmware.com>
11. Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., Warfield, A.: Xen and the art of virtualization. Proceedings of the nineteenth ACM symposium on Operating systems principles (2003) 164–177

Tartini: The Music Analysis Tool

Philip McLeod

University of Otago, New Zealand,
pmcleod@cs.otago.ac.nz,
WWW home page: <http://www.tartini.net>

Abstract. Tartini is a program designed as a practical music analysis tool for singers and instrumentalists. All you need is a microphone and your computer can become a real-time visual feedback system. It will provide accurate pitch contours for visualising intonation, loudness and vibrato shape. Other experimental views exist, such as one to show a note's harmonic structure.

1 Background

The name of our project comes from Giuseppe Tartini, who discovered the phenomenon of 'difference tones', also now called Tartini tones, in 1714. He used this property to help teach his students to play in tune. The basic idea of this phenomenon is if two related notes are played simultaneously on a violin, a third sound could be heard. For example if a 'C' and a 'G' are played on the same octave, then a 'C' an octave lower can be heard, only if the interval is a perfect fifth. Thus Giuseppe Tartini can be seen as an early example of someone trying to apply scientific principles to understanding and improving musical technique. The development of this project approaches ways to improve musical technique extending from his discovery.

The main goal of this project is to design tools to aid musicians, which could include simplifying or speeding up the learning process, or facilitating refinement of techniques to new levels of accuracy. The primary focus is on the pitch related aspects of the music, as finding pitch is a fundamental problem in a number of fields such as Music Information Retrieval (MIR), speech recognition and automatic score writing. The application of finding pitch has expanded into some other fields recently, such as computer games like "SingStar" [6], and singing tools, such as "Sing and See" [1] and "Melodyne" [2]. The work on pitch recognition of the PhD has been already largely completed. However, this paper discusses the more recent work on pitch analysis of vibrato at a higher level, which includes finding and displaying parameters of vibrato's, speed, width, cycle shape and envelope shape. Vibrato is a cyclic variation of pitch during a note, adding warmth and giving it a more voice-like quality. On a bowed string instrument, such as violin or cello, vibrato can be produced with a rocking motion of the finger on the fingerboard, by movement of the wrist or arm.

The research focuses on fretless String players in particular, such as violinists. It is because with fretless instruments it can take years before a player can play

in tune. However, pitch control is more than just finding the notes. Notes can be decorated using vibrato or glissando¹ for a greater effect for instance.

When a violinist plays a note he or she has to make a decision about the intonation, that is, the fine accuracy to which the player chooses the pitch of a note. This will depend on context and the type of tuning being used, for example “just intonation” in which the notes have frequency ratios that are low whole numbers, such as 3:2, 5:4 or 7:4, or “even tempered” in which the notes are tuned like a piano. Often an individual musician may have his own particular preference for intonation, for example they may prefer slightly sharper major sevenths, or flatter minor thirds. Using Tartini, we have been working with some of the music performance students at Otago, allowing them to experiment with scales and notes as well as vibrato, and compare what they played and heard with a reference scale, displayed on a large screen.

2 Pitch Recognition

The pitch algorithm, as described in [3], uses a Normalised Square Difference Function, in conjunction with a peak picking algorithm. It makes use of the Fast Fourier Transform for efficiency of calculation.

Recent developments involves extending this idea into an incremental method. This means when a new sample is added to the analysis window, and an old one removed, it gives a new pitch value at a small computational cost. The result gives a pitch value for each sample - normally 44100 samples per second. 44100 Hz is a much higher sample rate than necessary, however it will capture all the details of variation in the pitch, such as that during vibrato. Down-sampling can be performed to achieve a lower rate as needed, although this can still be much higher than the rate in existing methods. Vibrato parameters are currently found using a series of Prony’s spectral line estimations [5]. However this is still under investigation.

3 The program

This section discusses the User Interface, and some of the issues that arose in making the parameters of Tartini usable by musicians. Tartini’s interface is made up of a series of views each showing information in real-time about the sound.

The main view that consists of a pitch contour (Figure 1A), and a loudness graph (Figure 1B). It was found that having a slightly different shade background colour above and below the lines enabled quick identification of the lines, especially if scrolling was needed. In addition, an auto scrolling feature was added, but people commented it made them feel sea sick. The main view also has customisable reference lines that show the notes of a selected scale, and tuning. The view in (Figure 1C) shows a very responsive chromatic tuner, displaying instantaneous pitch, with a slider to control averaging as needed. However, most

¹ A continuous slide from one pitch to another

people prefer the pitch contour view, with its ability to show the pitch's recent history.

The other view shows a detailed analysis of vibrato, and consists of three parts. Firstly, a view of the vibrato of an individual note, showing details of precisely how the pitch varied (Figure 1E). Secondly, Figure 1F contains an even more magnified view of the current vibrato period, revealing the cycle shape. Lastly, Figure 1D shows the vibrato's instantaneous cycle rate, and width.

From the idea that seeing a history of values can often be better than only instantaneous values, background shading was implemented using the vibrato width² showing the envelope shape (Figure 1E). Also, the vibrato average is shown as a line. As a vibrato gets wider it often gets slower, causing the sinusoid to be scaled in both dimensions. This scaling makes it look like the same sinusoid, just closer or further away, so one does not notice the change in speed. Alternating light coloured background stripes were added, each showing half a period, so at a glance it can be seen where the vibrato has speeded up or slowed down no matter what its width.

Tartini also has a number of other views, including one to visualise the harmonic structure of a note against a piano keyboard (Figure 2). The current note is depressed and shown in yellow, with the tracks extruding backward, each representing the history of a sinusoidal component in the sound. The alternating colour distinguishes even from odd harmonics. The tracks become closer

² The vibrato width is twice the amplitude, and shown vertically

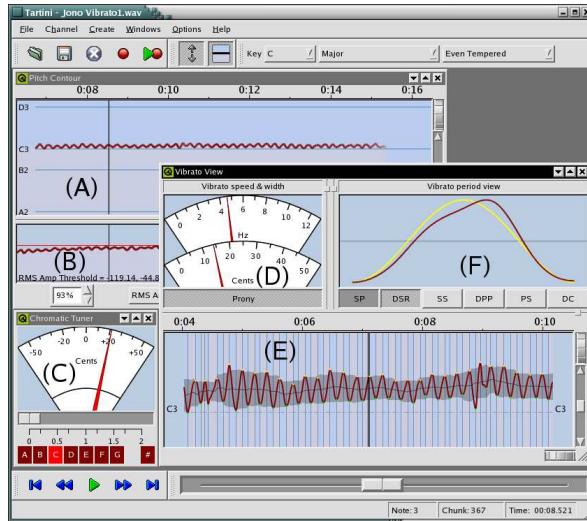


Fig. 1. A screen-shot of Tartini. Here (A) shows the typical pitch contour view, (B) shows the loudness level, (C) shows the instantaneous pitch, (D) shows the vibrato's instantaneous rate and width, (E) shows a zoomed view of the vibrato shape and (F) shows an even more detailed view of the current cycle of vibrato.

together because equal intervals are nearly equal distances on the keyboard but equal multiples of the fundamental frequency. The latest release of Tartini can be found at [4].

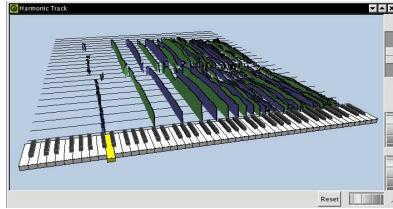


Fig. 2. A screen-shot of Tartini's 'Harmonic Track'.

4 Future Work

Although so far my PhD constitutes the bulk of this project, others have already taken interest in further work. There are a number of possible things that could follow on from this work. Firstly, the pitch algorithm still has some room for improvement, with emphasis on reducing octave errors, a common problem. Secondly, there could be development of more features for practising specific tasks, such as changing bow direction, glissando and pizzicato. These are common tasks every professional violinist has to practise and refine. The hope is this will expand into a tool used by a wide range of musicians, that provides instant objective feedback to a wide range of tasks. It should help detect defects, and point the user in the direction for rapid improvement, meanwhile also allowing for the musician to experiment freely.

Other ideas of continuing research include developing a better model of overtone structure, taking into account that not all overtones are perfectly harmonic, due to non-linearity in the physical nature of an instrument.

Lastly, looking from a higher perspective, now with tools to do analysis on vibrato, there is room to investigate what constitutes good vibrato? Is it a sinusoidal shape? Research could be done in to what the experts do, or how vibrato varies between styles.

References

1. CantOvation Ltd, 2006, *Sing & See - Visual Feedback for singing training* URL <http://www.singandsee.com>, [accessed December 2006]
2. Celemony Software, 2006, *celemony_* URL <http://www.celemony.com>, [acessed December 2006]
3. McLeod, P., Wyvill, G.: A Smarter Way to Find Pitch. Proc. International Computer Music Conference, Barcelona, Spain, September 5-9, 2005, pp 138-141.
4. McLeod, P.: 2006, *Tartini*, URL <http://www.tartini.net>, [accessed March 2007]
5. S. M. Kay and JR. S. L Marple. Spectrum analysis - a modern perspective. In *Proceedings of the IEEE*, volume 69, pages 1380 - 1419. IEEE, November 1981
6. Sony Computer Entertainment Europe, 2006, *SingStar* URL <http://www.singstargame.com>, [acessed January 2007]

