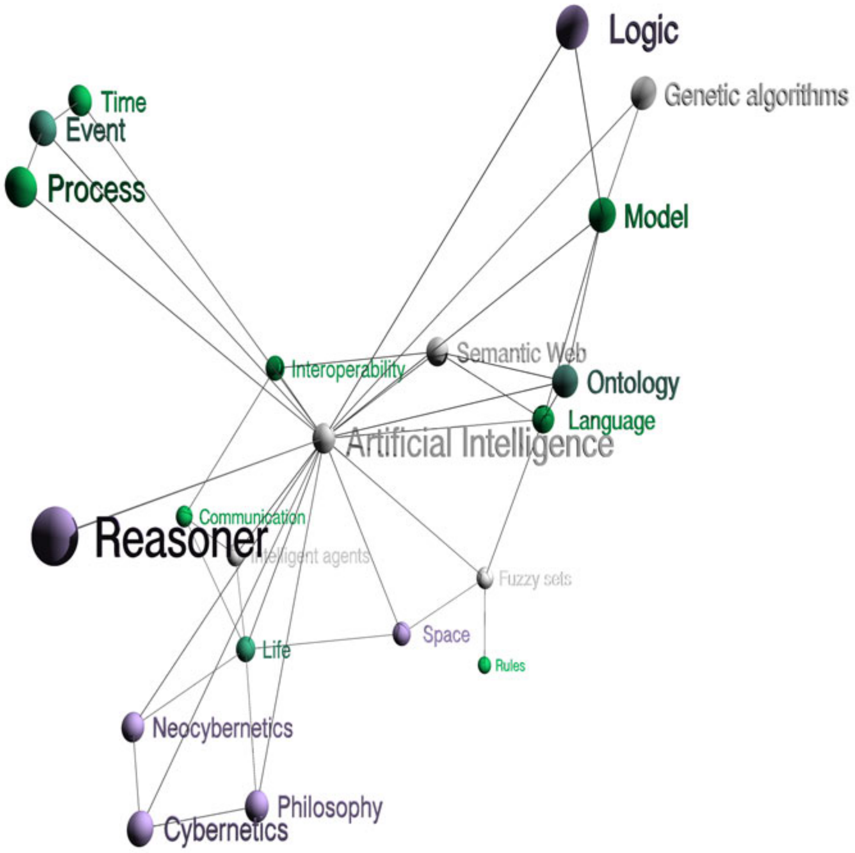
Publications of the Finnish Artificial Intelligence Society - 23



UNIVERSITY OF HELSINKI

New Developments in Artificial Intelligence and the Semantic Web

Proceedings of the 12th Finnish Artificial Intelligence Conference STeP 2006, Helsinki University of Technology, Espoo, Finland, October 26-27, 2006.

Edited by *Eero Hyvönen, Tomi Kauppinen, Jukka Kortela, Mikko Laukkanen, Tapani Raiko,* and *Kim Viljanen*

Finnish Artificial Intelligence Society – Helsinki University of Technology – University of Helsinki

Publications of the Finnish Artificial Intelligence Society

The Finnish Artificial Intelligence Society publishes national and international conference papers on theoretical and applied artificial intelligence research and popular multidisciplinary symposium papers on intelligence and related studies in Finnish and other languages.

New Developments in Artificial Intelligence and the Semantic Web
The 12th Finnish Artificial Intelligence Conference STeP 2006

Helsinki University of Technology, Espoo, Finland, October 26-27, 2006.

Organizers: Finnish Artificial Intelligence Society
Helsinki University of Technology
University of Helsinki
Semantic Computing Research Group

Sponsors: TeliaSonera Finland Oyj
City of Espoo
Nokia Oyj

Program Committee and Editors:
Eero Hyvönen, Helsinki University of Technology, University of Helsinki, Chair
Tomi Kauppinen, University of Helsinki, Helsinki University of Technology
Jukka Kortela, Artificial Intelligence Technologies, University of Oulu
Mikko Laukkanen, TeliaSonera Finland Oyj
Tapani Raiko, Helsinki University of Technology
Kim Viljanen, Helsinki University of Technology

Organizing Committee:
Tapani Raiko, Helsinki University of Technology, Chair
Tomi Kauppinen, University of Helsinki, Helsinki University of Technology
Jukka Kortela, Artificial Intelligence Technologies, University of Oulu
Eero Hyvönen, Helsinki University of Technology, University of Helsinki
Mikko Laukkanen, TeliaSonera Finland Oyj
Kim Viljanen, Helsinki University of Technology
Iina Tarnanen, Helsinki University of Technology
Pentti Haikonen, Nokia Research Center
Harri Valpola, Helsinki University of Technology
Krista Lagus, Helsinki University of Technology
Otto Lappi, University of Helsinki
Matti Miettinen, RD-Tech

Cover: *Tomi Kauppinen, Kim Viljanen,* and *AI*

Series Editor: *Tapani Raiko*, Helsinki University of Technology

Series: Conference Series – No 23 – Proceedings of STeP 2006

Publisher: Finnish Artificial Intelligence Society

Printed by: TSF Vallilan kopiopalvelu, Helsinki

ISBN-13: 978-952-5677-02-7 (paperback)
ISBN-10: 952-5677-02-8 (paperback)
ISSN: 1238-4658 (Print)

ISBN-13: 978-952-5677-03-4 (PDF)
ISBN-10: 952-5677-03-6 (PDF)
ISSN: 1796-623X (Online)

URL: *http://www.stes.fi/step2006/proceedings.pdf*

Sponsors and Partners of STeP 2006

TeliaSonera

NOKIA

ESPOO

THE CAPITAL OF KNOWLEDGE

SeCo

semantic computing

Welcome to STeP 2006!

These proceedings contain the research papers presented at the 12th Finnish Artificial Intelligence Conference (Suomen tekoälytutkimuksen päivät, STeP) that has been held biannually since 1984. In 2006, the twelth conference was organised alongside the ninth Scandinavian Conference on Artificial Intelligence (SCAI 2006) at Helsinki University of Technology, Espoo. The first half of these proceedings is on the minisymposium "Semantic Web at Work" whereas the second half contains topics from various areas of artificial intelligence.

The Semantic Web is a research and application area where artificial intelligence techniques, such as knowledge representation and reasoning techniques are combined with web technologies, and applied to the domain of sharing and reusing knowledge and services on the web. There are already many standards in use, such as RDF and OWL, and lots of tools and methodologies are being developed for creating applications. However, at the moment the number of practical semantic web applications on the web is not very large in comparison with the wide interest in the area.

The idea of this mini symposium was to focus on the practical aspects of making the vision of the Semantic Web through in practice. The papers presented explain and discuss the benefits and drawbacks of using semantic web technologies in real life applications. The theme is discussed from two viewpoints. From the application developers' viewpoint, best practices and methodologies of creating ontologies and semantic web systems are in focus. From the end-user's viewpoint, the semantic services and innovative user interface designs are discussed, as well as issues related to the content creation and harvesting processes on the web.

The symposium programme was opened with a keynote talk by Ora Lassila, Nokia Research Centre Cambridge, and was followed by two consecutive sessions containing thirteen presentations. The second day of STeP 2006 was opened with a keynote talk by Tom Ziemke, University of Skövde, followed by again two consecutive sessions containing eleven presentations. We thank all contributors and participants of the conference and authors of these proceedings for their activity in this exiting field of technology and for smooth co-operation.

The occasion was organized by the Finnish Artificial Intelligence Society together with the National Semantic Web Ontology Project (FinnONTO) 2003-2007, being conducted at the Semantic Computing Research Group at the Helsinki University of Technology (TKK), Laboratory of Media Technology, and at the University of Helsinki, Department of Computer Science.

TeliaSonera Finland Oyj has provided financial support for printing the proceedings.

October 16, 2006

Eero Hyvönen, Tomi Kauppinen, Jukka Kortela, Mikko Laukkanen,
Tapani Raiko, and Kim Viljanen

Table of Contents

Semantic Web at Work

New Developments in Artificial Intelligence

Semantic media application for combining and playing with user created and professional content

Asta Bäck
VTT Media and Internet
P.O. Box 1000, FI-02044 VTT, Finland
asta.back@vtt.fi

Sari Vainikainen
VTT Media and Internet
P.O. Box 1000, FI-02044 VTT, Finland
sari.vainikainen@vtt.fi

Pirjo Näkki
VTT Media and Internet
P.O.Box, FI-02044 VTT, Finland
pirjo.nakki@vtt.fi

Abstract

There are two important trends bringing changes and new opportunities into media consumption: the emergence of user-created content and Semantic Web technologies. In this paper we present an application that shows how these technologies can be combined to create an enjoyable media consumption experience. The application development work served also as a feasibility test of the maturity of Semantic Web technologies for media sector applications. The application contains material relating to the historical Ox road of Häme. The results indicated that users enjoyed using the application. Several ontologies were utilised, most of which were based on existing ontologies or taxonomies. With their help, it was possible to offer multiple views and exploratory routes into the available content. Further development can, among other things, be made in improving search strategies and in utilising user-created metadata for both for enriching ontologies and as an indication of user interests.

1 Introduction

Media sector is undergoing huge changes as the continuously evolving electronic media gets a stronger role in consumers' daily lives. Another important change is the more active role of media consumers. The active role does not only mean commenting and discussing the content that media companies publish but actively interacting with the content - publishing one’s own content and combining self-created content with content from other sources. Neither are users satisfied only with good usability but they expect enjoyable experiences with a considerable element of play and fun.

Semantic Web is an important trend changing the Web. The vision of the Semantic Web is to make the web more intelligent. Semantic Web technologies such as standards and tools relating to ontologies are currently being developed to reach this goal.

The work that we describe here was made in a project that wanted to research what kind of new opportunities these two trends bring to commercial media companies. In our view these trends connect to each other. Semantic Web technologies make it possible to make more enjoyable media content experiences because applications can be made more intelligent, and this way they require less effort from the users. The research approach of the project was prototyping, and a prototype application called StorySlotMachine was developed. The application helps people in choosing a travel destination by letting them explore background information relating to sights. They can also combine different media objects - both their own and others’ - into presentations. The assembled material can be taken along to enrich the actual visit. The aim was to make an application that offers interactivity opportunities for the active users, but also gives an enjoyable user experience for the less active ones. All this should be built utilising rich metadata and Semantic Web technologies to test their applicability.

The text is originally published in Asta Bäck & al. Semantically supported media services with user participation. Report on the RISE-project. VTT Publications 612. Espoo 2006. Reprinted with permission from the publisher.

2 Use scenario

Our initial use scenario was inspired by a slot machine analogy: users are presented with some content in the topic of their interest, and if they are not happy with the results, they can try their luck again. An underlying assumption was that if a person does not know so much about a topic, exploring and browsing a media object collection is more pleasant than making explicit searches. Also, the results should not be shown as a typical list of items as usual in search engines, but as a page or collection where different elements like images, videos and texts may be seen.

The presented material may then be taken as a starting point to explore the topic more, a bit like with a slot machine, where some of the items may be locked and some redrawn to improve the result. The most active users may explore the topic from many different points of view whereas the less active ones are satisfied with what is initially shown them. This way both the more and less active users are taken into consideration.

Our scenario also includes the opportunity to store the presentation and show it to other people, because an important feature that many people appreciate is the opportunity to get feedback from other users. Other opportunities for utilising presentations are either exporting it into a personal devise or printing the content. Different templates may be offered to take into consideration, which media elements are emphasised and for which device the presentation is generated for. If allowed by the original creator, other users may utilise these presentations and their components in their own ones.

We chose location related content for our pilot application with the emphasis on travelling. When preparing for a trip, people often are interested in exploring content to find out about their destination. During a trip, people take photos and videos, which can be used together with content from other sources.

The use scenario can be divided into three separate cases: before, during and after the trip. Before the trip the user can familiarise with potential destinations and their sights to find out more about them. The material can be browsed theme wise, and the user can select and combine the most relevant items into a collection that can be viewed either on the web or printed to be taken along for the trip. After the trip, the user makes his own travel story either utilising his or her own material or by combining it with the material of fellow users or the content that the media company provides. Also after the trip, the user may make theme stories like before the trip as well as also normal word based searches. The users are encouraged to add metadata in the form of keywords or tags, which are utilised to propose additional material. The users may choose any words to describe their content or choose from the ones that the system offers based on relevant ontologies.

3 User interfaces

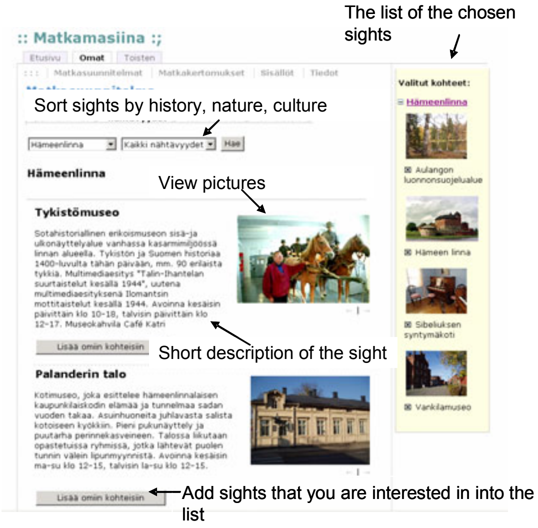
This chapter presents screen shots of the user interfaces and describes their functionality. More detailed descriptions of the implementation and utilisation of underling ontologies are presented in chapter “Ontologies”.

The first step is to select the places of interest from the map or list. The demonstration target area is the Ox road of Häme[1], a historical route between Hämeenlinna and Turku in the South of Finland. After selecting a place, the user is shown a list of the sights located there.

The user can sort the sights by history, nature and culture, read short descriptions of them, view both commercial and user imported pictures and add the sights he or she find most interesting into a personal item list (see Figure 1).

The user can search background information of the selected sights as theme stories (Figure 2). A theme story is a collection of media content from some point of view (Figure 3). Our theme stories are “Life now and then”, “Life stories”, “Nature and animals”, “Historical events”, “Fairytales and stories”, “Wars”, and “Art and culture”. Some of the themes are divided into sub themes. For example, historical events are divided according to historical periods. Only the categories with some content are shown to the user. The user can play with the content: View commercial and user-created pictures and videos, and view and build theme stories. The user may include theme stories into the travel plan to be created for the trip, as well as photos and descriptions of the chosen sights. The travel plan is available as a slide show and as a web page suitable for printing.

[1] http://www.harkatie.net/english/index.html



Tykistömuseo

Sotahistoriallinen erikoismuseon sisä-ja ulkonäyttelyalue vanhassa kasarmimiljöössä linnan alueella. Tykistön ja Suomen historiaa 1400-luvulta tähän päivään, mm. 90 erilaista tykkiä. Multimediaesitys "Talin-Ihantelan suurtaistelut kesällä 1944", uutena multimediaesityksenä Ilomantsin mottitaistelut kesällä 1944. Avoinna kesäisin päivittäin klo 10-18, talvisin päivittäin klo 12-17. Museokahvila Café Katri

Palanderin talo

Kotimuseo, joka esittelee hämeenlinnalaisen kaupunkilaiskodin elämää ja tunnelmaa sadan vuoden takaa. Asuinhuoneita juhlavasta salista kotoiseen kyökkiin. Pieni pukunäyttely ja puutarha perinnekasveineen. Talossa liikutaan opastetuissa ryhmissä, jotka lähtevät puolen tunnin välein lipunmyynnistä. Avoinna kesäisin ma-su klo 12-15, talvisin la-su klo 12-15.

The list of the chosen sights

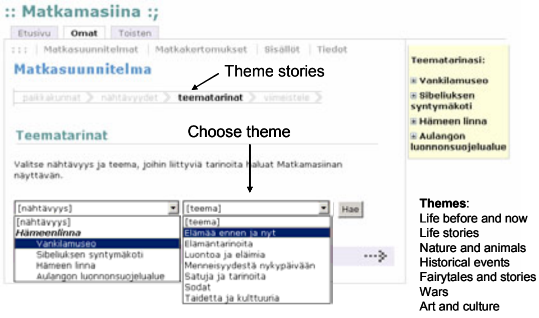
Sort sights by history, nature, culture

View pictures

Short description of the sight

Add sights that you are interested in into the list

Figure 1: Choosing sights to visit.



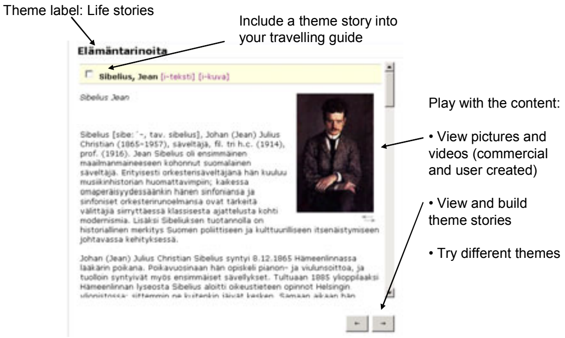
Valitse nähtävyys ja teema, joihin liittyviä tarinoita haluat Matkamasiinan näyttävän.

Theme stories

Choose theme

Themes:
Life before and now
Life stories
Nature and animals
Historical events
Fairytales and stories
Wars
Art and culture

Figure 2: Making theme stories to get background information relating to the selected sight



Sibelius Jean

Sibelius [sibe:´-, tav. sibelius], Johan (Jean) Julius Christian (1865-1957), säveltäjä, fil. tri h.c. (1914), prof. (1916). Jean Sibelius oli ensimmäinen maailmanmaineeseen kohonnut suomalainen säveltäjä. Erityisesti orkesterisäveltäjänä hän kuuluu musiikinhistorian huomattavimpiin; kaikessa omaperäisyydessäänkin hänen sinfoniansa ja sinfoniset orkesterirunoelmansa ovat tärkeitä välittäjiä siirryttäessä klassisesta ajattelusta kohti modernismia. Lisäksi Sibeliuksen tuotannolla on historiallinen merkitys Suomen poliittiseen ja kulttuuriliseen itsenäistymiseen johtavassa kehityksessä.

Johan (Jean) Julius Christian Sibelius syntyi 8.12.1865 Hämeenlinnassa lääkärin poikana. Poikavuosinaan hän opiskeli pianon- ja viulunsoittoa, ja tuolloin syntyivät myös ensimmäiset sävellykset. Tultuaan 1885 ylioppilaaksi Hämeenlinnan lyseosta Sibelius aloitti oikeustieteen opinnot Helsingin yliopistossa; sittemmin ne kuitenkin jäivät kesken. Samaan aikaan hän

Theme label: Life stories

Include a theme story into your travelling guide

Play with the content:

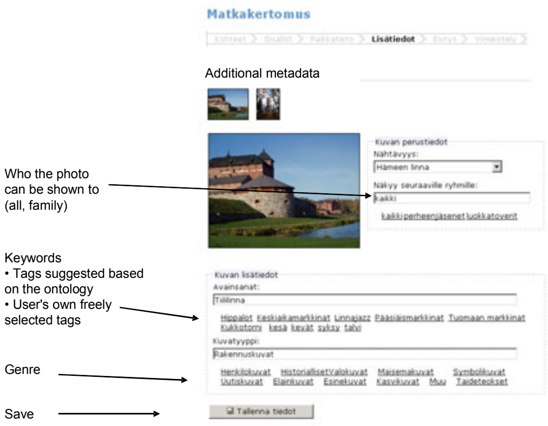
- View pictures and videos (commercial and user created)
- View and build theme stories
- Try different themes

Figure 3: An example of a theme story.

After the trip, the user may create his or her own travel story by utilising his/her own material and the materials in the system. Photos can be uploaded after selecting the visited sights. As part of the uploading process, the user determines whether the photos can be viewed by other users, and accepts the licensing terms.

After uploading the content, the user is asked to add some metadata. As the first step, the photos are connected to the sights by dragging and dropping them to the correct sight. After that, additional metadata can be given in the form of keywords or tags and by indicating the genre of the photo (see Figure 4). The keywords can be written freely or the user may utilise those that are suggested by the system based on relevant ontologies. The user may also add free text to his or her photos and change the visibility of the photos to other users.

Users are offered commercial and other users' content, which they can combine with their own (see Figure 5). There are several ways to search for additional content. The user can browse through the available photos, videos and texts. Content can also be searched with the help of tags, both user's own tags and those suggested by the application based on ontologies, or by making a traditional free text search. The travel story is created automatically out of the content selected by the user. It can be viewed as a slide show or as a web page suitable for printing.


Matkakertomus
Kohteet
Sisällöt
Paikkatieto
Lisätiedot
Esitys
Viimeistely
Kuvan perustiedot
Nähtävyys:
Hämeen linna
Näkyy seuraaville ryhmille:
kaikki
kaikki perheenjäsenet luokkatoverit
Kuvan lisätiedot
Avainsanat:
Tillinna
Hippalot Keskiakamarkkinat Linnajazz Pääsiäismarkkinat Tuomaan markkinat Kukkotorni kesä kevät syksy talvi
Kuvatyyppi:
Rakennuskuvat
Henkilökuvat HistoriallisetValokuvat Maisemakuvat Symbolikuvat Uutiskuvat Eläinkuvat Esinekuvat Kasvikuvat Muu Taideteokset
Tallenna tiedot

Additional metadata

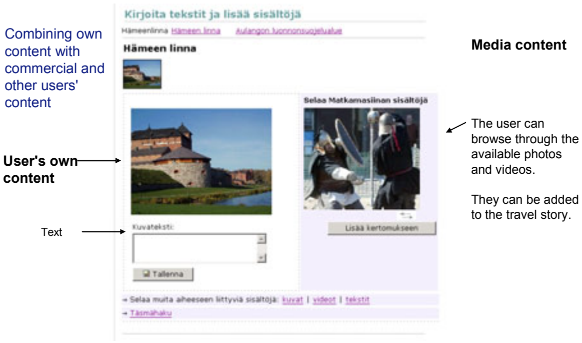
Who the photo can be shown to (all, family)

Keywords
- Tags suggested based on the ontology
- User's own freely selected tags

Genre

Save

Figure 4: Adding metadata


Combining own content with commercial and other users' content
Kirjoita tekstit ja lisää sisältöjä
Hämeenlinna Hämeen linna Aulangon luonnonsuojelualue
Hämeen linna
Media content
Selaa Matkamasiinan sisältöjä
User's own content
Text
Kuvateksti:
Tallenna
Lisää kertomukseen
Selaa muita aiheeseen liittyviä sisältöjä: kuvat | videot | tekstit
Täsmähaku

The user can browse through the available photos and videos.

They can be added to the travel story.

Figure 5: Combining user-created content with commercial and other users' content. Additional content can be found by browsing available photos and videos by media types or tags.

4 Content

We Different types of media content, such as facts, stories and news, are needed in order to be able to create versatile travel plans, theme stories and travel stories. Media content that is directly related to the target area is preferred, but also more general information is usable. A mixture of videos, photos, sounds and texts makes the presentations more appealing and interesting.

The commercial media content of the pilot application consists of newspaper and encyclopaedia articles with images, articles from the Häme Ox road magazines, stories out of a book called "Hämeen Härkätiellä", and photos from the Häme Ox road website. In addition to the commercial content, the application has user-created photos. The content is mostly general background information and not specific travel information like opening hours or prices.

This mixture of content and media formats meant that it was necessary to work with several metadata vocabularies. Different vocabularies are used to describe newspaper, magazine and encyclopaedia articles as well as short stories and users' own content. Also different media formats (text, photos, and videos) have different needs and vocabularies for metadata.

The content was delivered for our prototype in different formats and the amount of metadata varied a lot. The project did not address automatic methods for creating semantic metadata, and adding metadata and converting it into RDF required manual work.

The newspaper articles and images had some metadata that had been generated in the normal newspaper production process, and some more metadata like genre, scene and IPTC subject codes were added by the media company persons for the prototype. We received the metadata in text format and it had a structure that helped us in converting it into XML even though manual work could not be avoided completely.

The encyclopaedia articles were delivered in XML and the structure of the articles could be utilised in converting their metadata into RDF. The encyclopaedia content also has the potential for populating the target ontology, for example with persons relating to the Finnish history.

The content received from the Häme Ox road did not contain any metadata so the metadata was created by hand. The articles of the Häme Ox road magazines were received in PDF format, and that caused also extra work.

5 Ontologies

5.1 The role of ontologies

The prototype utilises a number of ontologies, each of which captures knowledge of some area that is necessary to fulfil the required functionality. Ontologies are utilised when selecting content and also to produce some basic information to be shown to the users. The ontologies are also utilised when users add metadata to their own content such as suggestions to keywords.

The Target ontology describes the knowledge related to places, routes and sights, and contains information that has relevance to them such as persons, events, objects and nature.

The Media ontology describes the media content. Relevant elements were selected out of the Dublin Core (DC) and IPTC Newscode vocabularies. The Media ontology includes the typical metadata fields, such as title, creator, publisher, date, media type, genre, scene, but also relations to the Time and Target ontologies, for example relations to persons, sights, places, routes, events, objects, animals or plants. The subject of media content was described with the YSA ontology (a general-purpose thesaurus in Finnish) whenever possible, but for news articles also the IPTC and for encyclopaedia articles the Facta ontologies were used.

The Presentation ontology contains the information on themes and their subcategories and what kind of content (subject, genre, scene, time) is to be searched for presentations. There are themes like "Life now and then", "Life stories", "Nature and animals", "Historical events", "Fairytales and stories", "Wars", and "Art and culture".

An ontology based on YSA (a general-purpose thesaurus in Finnish) is utilised as a kind of upper ontology for classifying the knowledge. Target, Media and Presentation ontologies are connected to each other via the concepts of this upper YSA ontology. The YSA ontology was created only to a limited extent because the idea was to replace it with YSO (Finnish General Ontology), which was under development and not yet available during the time when the application was made.

The Time ontology defines a taxonomy of time eras and periods by time intervals, and it is based on the ontology developed in the MuseumFinland project[2]. We added some time periods relating to the Finnish history as well as the seasons.

The subject of the media content is determined differently for different content types: the IPTC ontology is used to determine the subject of newspaper articles. The ontology is based on the IPTC ontology[3] that was developed in the Neptuno-project. The content of encyclopaedia uses its own taxonomy (Facta ontology). YSA-ontology is usable as a general subject ontology.

5.2 Searching content for theme stories

Theme stories consist of elements like a title, text, image and fact box, and they are selected on the fly based on the knowledge in the ontologies. The fact box shows information retrieved out of the ontology. It may contain knowledge about how events, like a war, are related to the sight or basic information about a person who has a connection to the sight. Sometimes the user may wonder why a certain article was shown, and the role of the fact box is to give some indication about the connection.

Media content is not linked directly to the various themes. The knowledge in the Target ontology and the search rules are utilised in searching and offering relevant media content. The search rules are determined with the Presentation ontology, Java application and SPARQL queries. The criteria for how the media content is connected to a theme, such as the subject, genre or time, are determined in the Presentation ontology. The advantage is that the search criteria are not hidden inside the Java code, but that they can be changed by modifying the instances of the ontology. Also, themes may be created, changed or deleted by modifying the ontology classes or their instances.

The Java application creates SPARQL queries for searching relevant media content based on the knowledge in the Presentation ontology. Searches utilise the knowledge in the Target ontology (e.g. Life stories -> persons related to the sight) and/or subjects related to themes (e.g. Every day life now and before -> food, professions, clothing etc. or Wars -> Great Northern War, World War I & II etc.). In addition to that, some restrictions may be used, like time (e.g. Historical events), genre (e.g. Stories and fairy tails), place or sight.

The subjects of the different themes are determined as relations to the YSA ontology. Also the subjects of the IPTC and Facta ontologies are connected to themes. Media content that is related to same subjects is searched for. If content that is described with some other ontology were brought into the system, the subjects of this new ontology would need to be connected to the existing themes.

5.3 Adding metadata to user generated content

Users can add metadata to their own content. Users are free to use any words they want to describe their content, but by utilising the available contextual information and the Target ontology, keywords are suggested. These suggestions relate to yearly events, objects, terms, other related sights and seasons. It was made easy to use these terms– it is enough to click a word, and no writing is needed. Users are thus encouraged to use these words that can then be utilised to suggest additional relevant content from the system.

In similar manner, users are encouraged to add keywords relating to the genre based on the knowledge in the Media ontology. Genres have been defined for all media types but only image genres are currently utilised. The genre information is useful when the user generated media objects are utilised with future users.

5.4 Searching commercial content to complement user's own content

Offering media content to complement user's own content is based on the user-given metadata and the knowledge of the Target ontology. First, the media content that is related directly to the sight is searched. After that, more general media content relating to events, persons and places is searched for. The relevance of the media content is determined with the help of the search order starting with from exact searches and then proceeding to more general searches.

Additional content can be searched with the help of tags. The tags suggested by the ontology may also be related persons or events in addition to tags relating to yearly events, objects, terms, other sights relating to sight and seasons. Already existing theme stories made by earlier users might be an additional way to search information also when creating one's own travel story. Theme stories give ready-made text and image/video combinations that can easily be added to a new travel story.

6 Software and architecture

[2] http://museosuomi.cs.helsinki.fi/

[3] http://nets.ii.uam.es/neptuno/iptc/

The ontology editor Protégé 3.1 was used for developing ontologies. Ontologies were developed as RDFS-schema.

The application is implemented as a Java client – server solution using Struts framework and Java Server Pages (JSP). Tomcat 5.5 is used as the web server. The user interfaces were implemented with AJAX (Asynchronous JavaScript and XML) in order to offer good interactivity and usability, and to add new features, like drag-and-drop. Different views of the travel plan and travel story are generated utilising CSS style sheets.

The ontologies and RDF-based data are handled by Profium SIR (Semantic information router). A beta version with support for SPARQL-queries was used. Profium SIR saved the RDF data into a Postgres 7.4 database. Postgres 7.4 was used also for managing user information.

The Sparql4j-jdbc driver[4] was used for quering RDF-data. Profium SIR created the result according to the SPARQL Protocol for RDF specification[5] and forwarded it to a Sparql4j-jdbc driver, which provides the results via the Java ResultSet abstraction.

It is not easy to choose the tools for application development with Semantic Web technologies. There are several open source tools, most of which have been created for research purposes. Semantic Web related standards and recommendations are still under development, and different tools support different subsets of the standards. For example, we used one tool for developing the ontologies and another for handling the RDF-data, and this caused some extra work to overcome the incompatibilities.

Protégé 3.1 is a versatile ontology editor with many useful features. It also has features for managing and developing multiple related ontologies, but we had problems with this feature. Reopening a Protégé project file with connections to other ontologies caused error messages and sometimes even meshed up the instance data.

The development of a standard query language SPARQL for querying RDF repositories is a step to the right direction. We wanted to use such a version of Profium SIR that supported SPARQL even though it was in beta at the time. Java application development was speeded up by using the Sparql4j-jdbc driver with SIR, even though it supported only select and ask type of queries at the time of the application development.

Utilising AJAX made it possible to add impressive features into the web user interface. A downside was the lack of good development tools; debugging JavaScript code is troublesome. We also encountered the well-known problem for developing web applications for different web browsers: what works in one browser does not necessarily work in another.

As a brief summary we can conclude that there already are usable tools for developing semantic web application, but currently many tools only have a partial support for the specifications. There is room and need for further development to make the implementation and management of Semantic Web applications easier.

7 Results

7.1 User tests

The user experience of the application was tested in two phases in the context of real school excursions. The test group consisted of 33 schoolchildren (12–18 years old) and 4 teachers from four different schools. In the first phase, user needs and expectations were studied using artefact interviews, observation, collages, metadata test and prototype tests. The prototype tests were made using a co-discovery method, where two participants used the prototype together and discussed about the decisions they made. Some users tested the travel planning part of the software before an excursion, whereas others had already made a trip and created travel stories with the prototype.

At the end of the project the functional application was tested again with the same user group but with a smaller number of participants (6 schoolchildren, 12 years old). The test users had made a trip to the Häme Ox road and they used the application afterwards to store their own pictures and memories. The users were interviewed both before and after testing. After the test session they also filled out a short questionnaire.

As the result of the first interviews, observation and collages made by users, following requirements for the StorySlotMachine were found. The application should

- *arouse interest* and offer necessary facts before the trip
- enable *experiencing the stories* during the trip
- give *additional information* about the themes studied on the trip, as well as the themes about which no information was available on the trip
- support creating a *personalised travel story*
- enable storing rich metadata about pictures, e.g. memories and feelings, as well as comments and hints for other travellers.

A metadata test was made in order to gather information about the meanings that the users associate with their travel photos. The aim was to find out, how semantic information could be added into the pictures. The users were asked to add captions and keywords into their own travel photos,

[4] http://sourceforge.net/projects/sparql4j

[5] http://www.w3.org/TR/rdf-sparql-protocol/

as well as select applicable tags from a list of keywords. The written captions were generally very short, and the users did not necessarily remember anything about the objects of their photos. The intuitiveness of selecting keywords varied a lot among the users. The users must understand what the purpose of adding metadata is in order to find it easy to do. In addition, the user should see immediate advantage of adding metadata.

The keywords used to describe their photos can be divided into five groups: 1) description of an object or an action, 2) memories and atmosphere, 3) background information about the place or object, 4) questions for additional information (history and/or present), and 5) hints for other travellers.

The application functioned only partially in the first user tests. For that reason many of the test users found the user interface somewhat confusing and the idea of mixing own and media content did not become clear to everyone. In addition, media contents were not presented attractively enough to rouse the interest of users. Nonetheless, half of the users found the application engaging and useful. Schoolchildren appreciated the idea of finding the necessary information easily in one place. Images were regarded as the most interesting part of the content. The complete report of the first user tests can be read in Näkki (2006).

The prototype was developed further after the first user tests and tested again at the end of the project. In the second test, user attitudes towards the functional application were very positive. The system was found useful, quick and easy to use. Users found the StorySlotMachine more pleasant than traditional search machines, because the relevant content could be found easily as stories. The users regarded photos as the core of the application and added both their own and commercial pictures into their travel stories. The users were also eager to write short captions to the photos. Adding metadata into their own pictures was intuitive and did not burden the users. Other users' pictures from the same excursion were found interesting, as well.

Some users wanted to create their travel story quickly, whereas others were ready to use a lot of time to finish their stories. Interestingly, the StorySlotMachine was found to be suitable for both these user groups. All participants of the last tests said that they would like to use the system again. Summary of user experiences in the both test phases can be seen in Figure 6.

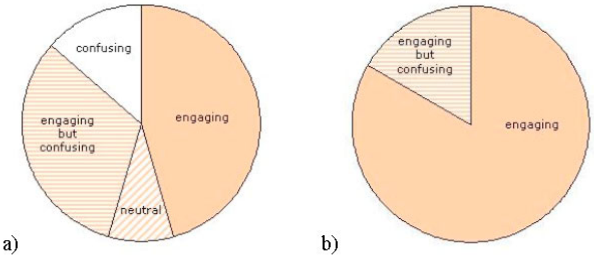


Figure 6: User experiences of the system: a) after the first user test (N=22), b) after the second user test (N=6).

From the users' point of view, the value of semantic content is in the quickness and easiness of information retrieval. The way to find information as stories was something new for the users, but most of them found the idea easy to understand. However, the users did not necessarily want to collect, store and print stories, when planning the trip. The system should therefore better support pure browsing of the content. After a trip it was seen more understandable and useful to create a travel story, where own memories are linked to editorial material.

When users create semantic content, one challenge lies in the process of adding metadata. It was discovered that the travel images and memories include a lot of meanings that are hard to put into words as simple keywords. Users' active participation will be needed, even though automatic semantic reasoning is used for creating presentations. It is the user who decides which content is valuable for her. However, the StorySlotMachine can substantially help the user by offering suggestions about the media content related to the theme of the user's trip. The semantic processing makes it possible to discover interesting and surprising relations between contents that would be hard to find otherwise.

7.2 Ontologies and implementation

Creating, updating and managing ontologies are not easy tasks, but there are clear benefits in this type of an application:

- Ontologies make it possible to search content from multiple directions (sights, events, persons etc.).
- Also general media content can be utilised.
- It is possible to make different thematic presentations or views for people with different interests.
- For example, one user might be interested in the historical places and events of the Ox road during the 19th century and another is only in churches during the trip. They can easily be served with this kind of an application.

- Ontologies contain knowledge that makes it possible to create visualisations such as timelines, cause-effect diagrams, dialogues, trees, and maps of related resources.
- Ontologies support generating aggregations automatically.
- The benefits of being able to link the content automatically into different themes become significant as the number of content items increases and grows continuously.

There already are usable tools for developing semantic web applications, but currently many tools only have a partial support for the specifications. There is room and need for further development to make the implementation and management of Semantic Web applications easier.

Theme stories were the most central part of the application for ontology deployment. Theme stories could be easily generated for sights with a long history, but not so smaller sights. Theme stories should rather be offered at higher level like for a town or a village or as in our case, for the whole historical route, than for a single sight.

There were challenges in creating general search rules for the themes. Every theme had unique requirements and complicated the Presentation ontology. Some examples are listed below:

- "Every day life now and before" has subcategories weekday, celebration and society. Subjects like food, professions, clothing, inhabitation, celebrations, laws, and source of livelihood relate to this theme. These determine the main framework for searching, but to get more relevant content also the time periods and the type of the sight should be determined to find relevant content for a particular sight.
- "Arts and culture" is divided into the following subcategories: persons, art and buildings, and environment. When searching for content for the subcategory 'Persons', it is required to find persons who have a connection to the sight and have or had a profession relating to art and culture, such as composer, writer, painter, or architect.
- "Historical events" are divided into historical periods, and time restrictions are needed in searches. There are several ways to do this: search the media content that relates to the time and the sight/place, utilise terms that describe the time period or events that happened during that time.
- In the theme "Stories and fairy tails" the genre is used to restrict the content selection.

When making ontology-based searches, several search criteria can be used and their priority order must be determined. Here, it is important to find the correct balance between the number of different search criteria to use and the speed of the application. First, the most relevant content is searched and after that, the search can be expanded to more general material. The challenge is to know how deep the searches should navigate into the net of relations of the ontology and still find content that is relevant to the user. We encountered this problem both when making theme stories and searching for additional content to complement users' own content.

When the application has more content, this will probably be less of a problem, and the challenge is the ordering or grouping of the relevant media objects in an interesting way. One of the challenges is to inform user of why certain content is offered to her/him in this context. For example, pictures of historical persons might be confusing, if user does not know who the person is and how he/she is relating to the sight. In connection to the theme stories, a fact box was used to give some indication about the connection by utilising the knowledge in the ontology, and a similar approach should be used elsewhere in the application.

The implementation uses an upper ontology that can be replaced with another one, if needed. This gives flexibility to the application development. We turned selected parts of YSA (a general-purpose thesaurus in Finnish) into ontology, and used it as our upper ontology. There is currently a project in Finland making a comprehensive ontology out of the YSA, and it should be used also here when it becomes available.

We had many different vocabularies for describing the subject of different type of media content in the StorySlotMachine application. Our first idea was to map the IPTC and Facta vocabularies to the concepts of our top YSA-ontology. Mapping different vocabularies to each other turned out to be complicated since it was not always possible to find corresponding concepts in the different ontologies. Also, there was a lack of good tools for ontology mapping.

Instead of mapping ontologies to each other, we decided to keep the different ontologies. As described earlier, the subjects of media objects were described with the concepts of the YSA-ontology when ever possible, but we also stored the original subjects described with IPTC or Facta. The subjects of the different themes were also described with the YSA, IPTC and Facta ontologies. Several searches may be done for media objects during a user session: first utilising the YSA concepts and the subjects from other ontologies. In other words, ontologies are not mapped to each other but to some extent, the mapping is done via the different themes.

In order to better test the feasibility of this approach more media content should be added to the system.

This approach will probably work with media companies' own media services, where they can decide which themes are available. Of course, it might be possible to offer users an interface where they can combine ontology concepts and create their own themes. One idea for the future development of the StorySlotMachine is to let users create new themes with the help of the tags they have used.

In general it is good practice to use one common upper vocabulary or ontology for describing the metadata inside the media house and also use standardised vocabularies as much as it is possible. However it is realistic to assume that a media house will have several vocabularies also in the future and one upper vocabulary or ontology cannot solve all issues and challenges. Better tools are needed to support ontology mappings and even better if mappings could be made automatically by the system.

8 Related work

The key idea in the StorySlotMachine is to aggregate content in a way that lets users explore the content in an enjoyable manner. Related work is being done in the various areas. The first distinction can be made between the aggregation level: is the aim a single story to be created of out the available content, or a collection of independent resources. Geurts et al. (2003) and the Artequakt project (Kim et al. 2002) work at the first area. Geurts et al. (2003) describe the system where the knowledge of ontologies is used to create multimedia presentations like artists' bibliographies. Presentations vary based on the genre (e.g. Biography and CV) and output format that can be selected by the user. The basic idea of their Discourse ontology is same than our Presentation ontology. The ontologies define rules for searching content. They have different genres, whereas we have themes. Our themes use more versatile data than what is needed for artists' bibliographies and we also have more general content which complicated the ontologies and rules. One difference is that they focus more on ready-made multimedia presentations, which contain parts (e.g. address, private life and career) that are determined in the Discourse ontology.

Our work is more related to creating a collection out of independent resources and turning them into presentations. However, we let users combine images and texts in new ways and we do not aim at producing one collection for the user to view but a starting point for further exploration with the content.

Mc Schraefel et al. (2005) have developed an open source framework called mSpace, which is available at mspace.sourceforge.net. The starting point for the mSpace development as well as for our StorySlotMachine is same: to offer an exploratory access to content. The user should be able to browse content according to their interests and associations, to leave tracks on the way by storing the most interesting items, and to get multimedia as a result rather than links to resources.

The original mSpace demonstrator was a Classical Music explorer , and it has since been utilised in other applications and domain.

mSpace is based on the idea of associative exploration of the content and user-defined and manipulated hierarchies. mSpace lets the user explore the material with the help of hierarchical columns like periods, composers, arrangements and pieces: the selection in the first column constrains the selections of the following column. Users can arrange columns according to their preferences and also add new dimensions or remove them.

mSpace provides preview cues (for example audio clips) of some representative example in the various dimensions to help in exploring and deciding whether an area is interesting. This way users may find new interesting areas without prior knowledge of them. mSpace also has info views to show related information like for example a description of a composer. Interesting items may be stored in favourites for future reference.

The preview cues in mSpace have the same aim as the themes in the StorySlotMachine: to give users ideas and hints as to what kind of content is available relating to a topic.

An interesting feature of mSpace is to let users sort and swap dimensions according to their interests. In the current StorySlotMachine version, the users are tied to pre-made themes, but one idea for future development is to let users create new themes with the help of the tags they have used.

Creation of different theme stories in StorySlotMachine is based on associations that are inferred automatically by utilising the knowledge in the ontologies. For example, a theme story may tell about a war that relates to a sight and story may include important historical persons. New theme stories are offered based on these relations and other theme story may tell more about the person. Now this is made by the system, as our guiding idea was to give the users the opportunity to try their luck and get surprised, but as an alternative, users could be given the opportunity to guide the process based on their own associations.

One difference between the StorySlotMachine and mSpace is that the StorySlotMachine offer users the possibility to make exportable packages out of the content and also utilise their own content. The mSpace user interface is more formal in style than in the StorySlotMachine, where emphasis has been put to offering a user interface with the element of play.

The Bletchley Park Text application developed for the Bletchley Park Museum (Mulholland et al.

2005) concentrates on post-visitors of museum. During their visit, people may express their interest by sending text (SMS) messages containing suggested keywords relating to displayed objects. After the visit, they can get a collection of content relating to the selected keywords as a personalised web site. The content can be explored and a number of different views on the collection are provided.

Bletchley Park Text application is made for a specific museum and its specific content. In the StorySlotMachine application, we have several places and sights, and the material is general by nature, since one of the major goals of our project was to study how the general media content can be utilised in new ways with the help of semantic metadata. Both Bletchley Park Text application and the StorySlotMachine share the similar ideas of using the application for learning, but the Bletchley Park Text does not include utilising users' own material like we do.

Bentley et al. (2006) have studied how consumers use photos and music to tell stories. The application is slightly different from ours, as they compare using only music and photos, but there are important similarities and lessons to be learnt. They find that different media formats, photos and music in this case, and commercial and private content should not be stored in separate silos. Instead, they should be available with the help of similar methods. Serendipitous finding utilising available information like contextual cues should be utilised to remind users of what is available and to help them in finding related resources. They also see a need for systems that allow communication by using media.

9 Discussion and future work

This chapter discusses future development opportunities of StorySlotMachine and what are the benefits, opportunities and main challenges of the media companies in creating new semantic media services, particularly in relation to StorySlotMachine type applications.

Easy access to electronic content and users' participation opportunities into the media production cycle are bringing about huge changes in the way that media content is created, offered and consumed. The StorySlotMachine explores the possibilities of letting people explore content in a playful and theme wise way and letting them do the final touch in putting the presentation together. Semantic metadata and ontologies are utilised to offer multiple views into the available content and help the users to explore and learn in a pleasant way of topics that may not be so familiar to them. The application also lets the users import their own content and mix it with other people's and commercial content.

The application is most suited when the content is available as relatively small units. The user tests indicated that photos and videos are important in raising interest, whereas particularly reading long texts requires more effort and is less attractive in a playful use scenario. This implies that text should be written in a way that makes it quick and easy to see what the text deals with to arouse users' interest. In this kind of a context, the user is not looking for a specific answer to a specific question, but looking around and checking if something interesting comes up, and the content that is presented should support this approach.

The application makes it possible for people to make their own narratives about a topic. This is relevant at least in connection to learning and hobbies, like travelling and making memorabilia. The speciality here is the opportunity to combine self-created content with content from other sources. The final result must have an attractive and professional look in order to motivate the use. The electronic format also makes it possible to add live features; for example, a user-made narrative may be updated with current news and developments.

It was not possible to carry out user tests to such an extent that we would have seen how people would like to use the materials and what new ideas come up. More research is also needed to see, how purposefully prepared content users want, or do they enjoy with a mix of material and modifying it according to their own ideas.

Serendipity is a concept that seems to be popping up as a goal in search related application (Leong et al. 2005; Bentley et al. 2006; Lassila 2006) and here Semantic Web technologies come into play. Particularly in consumer applications easy, intuitive and entertaining user interfaces and applications are needed. Also our work aims at providing serendipitous finding of interesting resources. We did not directly explore how this sensation emerges, and which factors contribute to it. One assumption is that if we know what the users' interests are, we'll be able to suggest resources that he or she is interested in, and can offer experiences of serendipitous finding.

Many media companies have extensive archives that are not effectively utilised as end user services. The StorySlotMachine is an example of how content can be offered in a more interesting context than as mere searches and search result lists. If the content is not already modular and if there is not much metadata about the content, an investment is needed to turn the content into more usable format. The current production processes and practices should be changed so that the content is directly processed into a format that supports the reuse in this kind of applications. The best starting point for offering this kind of a new service is an area where the content is already modular and where people may have longer-term interest and personal material. These include areas like travelling, hobbies, encyclopaedia and news.

Other key questions that media companies need to agree on are descriptive metadata and terms for licensing the content for creating the narratives, and how strictly to guard their IPR. On the other hand, networking to freely available network resources such as photos that are licensed under Creative Commons licenses should be considered as a way to add resources for users to choose.

We can see at least following business opportunities with this type of an application:

- some basic features could be free, but access to more content could be available for paying customers
- related materials could be available for buying, such as books or maps, or additional services like back-up storing of images
- co-operation with operators in the application area, for example with local travel associations
- targeted advertising, particularly if people can be encouraged to longer term use, then information about their interests will accumulate and opportunities for effective advertising becomes better.

There are several opportunities for utilising and developing the StorySlotMachine application further. The StorySlotMachine application can be used as a platform to test user expectations and experiences of mixing and playing with media content, and sharing one's own content with other users' content more extensively. More content should be added for additional testing, and the conversion of metadata into RDF format should be made automatically.

The application could be developed into a commercial travel application or a learning application, e.g. for teaching history. For the travel application, some additional features are needed, like exact travel information (opening hours, prices), maps and mobile user interface, and collecting feedback and recommendations from users. Also new features like collaborative storytelling e.g. creating one travel story from the contents of all group members, and real time travel story that is continuously updated with topical information, could be added.

Similar applications could be built relating to other topics such as hobbies or collecting gathering personal memories from past. A new Target ontology may be needed for a new application, if it does not make sense to expand the existing one. The search criteria are not hidden inside the Java code, but they can be changed by changing the instances of the ontology, which makes it easy to develop the current application further, and to adapt it to new areas. Also, themes may be created, changed or deleted by changing the classes of ontology or its instances.

There is always room for improving the search criteria with help of the Presentation ontology, or even a general tool for automatic generation of theme stories could be created. In the future, RuleML (Rule Markup Language) or SWRL (Semantic web rule language) may be the solution to use.

At the beginning of the application development, we considered using the CIDOC CRM cultural heritage ontology that is being developed specially for describing the cultural heritage of museum collections. We decided not to, because the ontology seemed too specific and complicated for our purposes. CIDOC CRM is currently in the final stage of the ISO process as ISO/PRF 21127, and it could be reconsidered as an option in a travel-related application like StorySlotMachine to describe cultural heritage. The challenge is to decide which level of the ontology to include and what to exclude as too detailed. Additional benefits could be gained if content or information can be integrated from various museums with the help of a common ontology.

Automatic methods for creating metadata and converting it into RDF format were not addressed in this project, but they are important particularly when existing media archives are utilised in semantic applications. Once the number of concepts in our YSA ontology has been increased, or the national YSO becomes available, utilising automatic methods will be easier. Additionally, users should be utilised as metadata creators, where feasible.

One of the future needs is to increase the amount of available content. In our application, the content resources were moved to our server, but in a commercial application, content from different sources should probably be utilised. This requires that there is an agreement on what metadata to use and the content should have this metadata.

There are many opportunities to develop the searches and the ways that search results are ordered for presentation. Scene and genre information could be used for ordering images. Images from outside and inside a building, and general images and detailed close-ups could be alternated. New ways grouping media objects could be developed in addition to the current location-based presentation.

User generated metadata could be utilised more extensively. The words that people use to describe their content could be stored in case they are not found in the ontologies, and they could be offered to future users visiting the same sight. In the current system, users can add tags only to describe their content, but tags could be utilised more widely, for example, to describe media content, travel plans, and sights. If we had mechanisms to combine tags with more formal semantics and to analyse the reliability of user generated knowledge, this could be one way of adding knowledge into the ontologies.

To summarise, we can conclude that the application lets users explore and combine various types of media content, as well as virtual and real life experiences. Utilising ontologies helps in making the application more intelligent and gives opportunities to offering enjoyable user experiences.

Acknowledgements

The StorySlotMachine was developed in the project “Rich Semantic Media for Private and Professional Users” (RISE). The two-year project was part of a technology programme “Interactive Computing Technology Programme” (FENIX) run by the Finnish Funding Agency for Technology and Innovation (Tekes).

Besides the main financier, Tekes and VTT, four companies supported the project financially and by giving expertise and tools for the project, and they were represented in the project management group. The project steering group consisted of the following persons: Eskoensio Pipatti (SanomaWSOY Oyj) and Tuomo Suominen (Werner Söderström Osakeyhtiö), Hannele Vihermaa (first half of the project) and Teppo Kurki (the second half of the project) (Alma Media Oyj), Jouni Siren (Yle), Janne Saarela (Profium Oy), Keith Bonnici (Tekes), Caj Södergård (VTT) and Marko Turpeinen (HIIT). The project group members want to thank the steering group for their good support and feedback. We want to express our special thanks to the Häme Ox Road project and project manager Matti Korolainen for their co-operation in the project - the materials and user contacts provided by them contributed to the successful completion of the project.

References

Bentley, F., Metcalf, C., Harboe, G. 2006. Personal vs. Commercial Content: The Similarities between Consumer Use of Photos and Music. In: CHI 2006. Montreal, Canada. April 22–27, 2006. ACM. Pp. 667–676. ISBN 1-59593-178-3/06/0004.

Geurts, J., Bocconi S., van Ossenbruggen, J., Hardman, L. 2003. Towards Ontology-driven Discourse: From Semantic Graphs to Multimedia Presentations. http://homepages.cwi.nl/~media/publications/iswc2003.pdf.

Kim, S., Harith, A., Hall, W., Lewis, P., Millard, D., Shadbolt, N., Weal, M. 2002. Artequakt (2002): Generating tailored biographies with automatically annotated fragments from the web. Proceedings of Semantic Authoring, Annotation and Knowledge Markup Workshop in the 15th European Conference on Artificial Intelligence. Lyon, France.

Lassila, O. 2006. Sharing Meaning Between Systems, Devices, Users and Culture. Symposium on Digital Semantic Content across Cultures. Paris, The Louvre. May 2–5, 2006. http://www.seco.tkk.fi/events/2006/2006-05-04-websemantique/presentations/friday-0900-Lassila-Ora-DSCaC.pdf.

Leong, T., Vetere, F., Howard, S. 2005. The Serendipity Shuffle. Proceedings of OZCHI, Canberra, Australia. November 23–25, 2005. ISBN 1-59593-222-4. 4 p.

Mulholland, P., Collins, T., Zdrahal, Z. 2005. Bletchley Park Text: Using mobile and semantic web technologies to support the post-visit use of online museum resources. http://jime.open.ac.uk/2005/24/mulholland-2005-24-paper.html.

Näkki, P. 2006. Käyttäjäkokemuksen suunnittelu semanttiseen mediapalveluun – tarkastelussa kouluretkien tarinat (in Finnish). Master’s thesis. Espoo: Helsinki University of Technology.

Describing and Linking Cultural Semantic Content by Using Situations and Actions

Miikka Junnila
[⋆]Semantic Computing Research Group
Helsinki Institute for Information Technology (HIIT)
University of Helsinki
http://www.seco.tkk.fi
miikka.junnila@uiah.fi

Eero Hyvönen
[†]Semantic Computing Research Group
Helsinki University of Technology (TKK)
University of Helsinki
http://www.seco.tkk.fi/
eero.hyvonen@tkk.fi

Mirva Salminen
[‡]Semantic Computing Research Group
Helsinki Institute for Information Technology (HIIT)
University of Helsinki
http://www.seco.tkk.fi/
mirva@pieni.net

Abstract

Ontologies have been used to describe cultural objects, such as artifacts, by their physical or media specific properties, or by the life cycle of the objects in collections. In contrast, this paper discusses the problem of creating ontological descriptions that allow describing different kinds of cultural content through the situations and actions that take place in the real world. This point of view is important when semantic metadata is used as a basis for creating intelligent, educational, and entertaining linking of content on the semantic web. The idea is addressed not only in theory but by presenting the first prototype implementation of a cross-domain semantic portal "CultureSampo—Finnish Culture on the Semantic Web". This system is able to automatically link together different kind of cultural resources in meaningful ways with explanations. The content types considered include paintings, artifacts, photographs, videos, cultural processes, and stories .

1 Introduction

This paper investigates possibilities of exploiting semantic cultural metadata in creating intelligent portals. The research is a continuation of the work behind the semantic portal MuseumFinland[1] (Hyvönen et al., 2005a). This work showed that, based on ontologies and associated metadata, semantic search and browsing can be supported to enhance end-user services. The content in MuseumFinland was homogenous in the sense that only artifact metadata conforming to a shared metadata schema and ontologies was used. In this paper the main research problem is to create an ontology that can be used to describe many kinds of cultural resources, so that they can still be searched with a unified logic and be linked together semantically in insightful ways.

We chose processes and actions (events) to be the basis of our ontology. In the context of cultural resources, this proved out to be a fruitful starting point. Many cultural objects of interest, such as paintings, stories, and artifacts have a connection to the processes of human life and the actions of people.

To get a concrete grip of the problems involved, we created the first prototype I of the portal "CultureSampo—Finnish Culture on the Semantic Web" system in 2005. This paper documents and summarizes experiences and lessons learned in this work documented in more detail in (Junnila, 2006; Salminen, 2006). The vision and some other results of the CultureSampo project in the larger perspective 2005-2007, including a later prototype CultureSampo II, is described in (Hyvönen et al., 2006). "Sampo" is a machine that fulfills all the needs of people in the Finnish mythology. We try to fulfill the needs of people interested in getting a good picture of Finnish culture through the semantic web.

The CultureSampo I prototype was built on top of the MuseumFinland architecture and tools (Mäkelä et al., 2004; Viljanen et al., 2006; Mäkelä et al.,

[1] This application is operational at http://www.museosuomi.fi with an English tutorial.

2006). The system is based on metadata of different kind of cultural objects, such as process descriptions, mythic poems, paintings, old photos, artifacts, and educational videos about Finnish culture. These resources have been described with matching metadata schemas and shared ontologies, which enables semantic search and browsing. We introduced a new type of metadata for describing actions and situations that relate cultural objects with each other. The main goal of this paper is to investigate, how such descriptions can be used to bring the cultural context closer to the people looking for information in cultural semantic portals.

2 Describing the Cultural Context of Resources

The motivation for describing cultural resources semantically is to make it easier to search for and automatically link culturally related things together. The more information there is on the semantic web, the more interesting and useful it can be to people. A prerequisite of this is that the information is easy to find, and that the cross-links from resource to resource are really semantically insightful and support the user in finding the right information and connections.

2.1 Towards Event-based Content Descriptions

Information that connects to other information is easier to grasp, as for example constructivist learning theories show (Holmes et al., 2001). In fact, many philosophers from Aristotle to Locke and Hume have stated that all knowledge is actually in the form of associations (Eysenc and Keane, 2002). All this leads us to observe the fact that links can be very useful for a person looking for information about a particular subject.

In our work, actions and processes in the real life and fiction were chosen as the key enabler for connecting different resources semantically. There were two major reasons for this. Firstly, actions and processes connect to many kinds of different resources. Human culture is much about action, about people doing things. People looking for information are also likely to be interested in the theme of action. Actions and processes often tell us more about the life around cultural objects than other points of view. Life, stories, and emotions are things that everyone can easily connect to and are interesting for the end-user. People with varying specific needs are surely interested in other points of view, too, but generally speaking actions are foundational in structuring our knowledge about the world. As a result, event-based representation have been widely developed and applied in the fields of artificial intelligence and knowledge representation (Sowa, 2000). CultureSampo builds upon this research tradition and ideas developed within semantic web research.

The second reason for using actions and processes is that the information about processes themselves is very important to preserve. For example, information about cultural processes, such as "how to farm land with the traditional slash burn method procedure", are important to preserve. By linking other resources through action, the processes themselves are researched and the old know-how preserved in digital format for future generations (Kettula, 2005).

Let us take an example of cultural resources and their connection to each other through their cultural context and actions. Consider a painting depicting people and burning trees. We know that the picture is a part of an old cultural process, slash and burn, where trees are cut down and burned to make the soil richer for nutrition of the crop. There is also a poem in the national Kalevala epic[2], where Kullervo, a tragic hero, is forced to cut down trees in the slash and burn process, but he ends up with cursing the whole forest. An axe displayed in the National Museum of Finland has been used in western Finland for slash and burn, and there may be other related tools, too. We also have some photos of people doing slash and burn, and an educational video about the subject. All these things can be linked together in insightful ways, if they have been annotated with metadata that tells about what is happening during the slash and burn procedure.

To accomplish the needed semantic annotations, we defined a set of domain and annotation ontologies. We then selected a representative set of heterogenous cultural contents of different kinds and annotated them with metadata conforming to the designed ontologies and annotation schemas. The result was a knowledge base in RDF(S)[3], that was homogenized based on the shared action-based knowledge representation scheme of the real world. After this, the view-based semantic search engine and logical recommender system of MUSEUMFINLAND was adapted and applied to the new content set, resulting in the prototype portal CultureSampo I.

[2] http://www.finlit.fi/kalevala/index.php?m=163&l=2
[3] http://www.w3.org/2001/sw/

2.2 Stories as Processes

In addition to describing general process types, we wanted to be able to describe specific process instances, i.e., situations where something actually happens. This leads us to a very classic content medium and type: stories. As processes are chains of actions, where something is done, leading to another action, and so on, stories are the same in many ways. Stories can bring separate facts into life and place in the right contexts for the end-user, and in this way give a better view to the culture that the resources are describing. Furthermore, many cultural content objects, such as historical events and biographies, are actually stories and often relate to other stories.

People want to hear stories (Kelly, 1999): they are an ancient method of spending time, educating and having fun. Stories bring new points of view to the process descriptions, which are on a general level. As Aristotle states in Poetics (Aristotle, 2006), drama is about action. This suggests that describing actions is a good way to describe drama. A process description just needs less information, as it is less precise. For a process description, it may be enough to tell what is done and in what order. A story needs additional information. For example, it may be important to know who does what, what are the relations between the actors, and what goes possibly wrong. Based on this, stories can be seen as a kind of subclass of processes, that need more information and give a richer and more specific description of a situation or happening.

Our main motivation to describe stories with metadata is not to represent the stories, but only to make the actual stories better accessible through better search and links from related resources. A story may be in textual form or maybe depicted in a painting, a photo or a video. The stories lose much in content when they are reduced to metadata, but since the story resources themselves can be put on the semantic web, people searching for stories will find and experience the real stories.

3 The Ontologies

Three ontologies were used to create the action-based metadata for the resources. First, a situation ontology was created in order to define how to describe one situation or moment, the smallest unit of a process or a story. Second, a process ontology was designed. It was used to put the moments in the right order in relation to each other: what is done first, what follows and so on. The third ontology was the content definition ontology, that included the concepts of the real world, so that reasoning could be made also based on the things that appear in the situations. For example, an axe is used in the slash and burn method, so the axe has its own place in the content definition ontology, as a subclass of tools. Also the actions, the verbs, and their relations to each other are an important part of the content definition ontology.

In the following, these three ontologies will next be discussed in some more detail. We concentrate on the situation ontology that was the main focus of the research.

3.1 The Situation Ontology

The situation ontology (cf. figure 1) is used to describe moments, where someone does something. It is actually not much of an ontology in the sense of a hierarchical network, but more of a metadata schema implemented with semantic web technology. The idea is to use instantiated situations for semantic browsing and search.

A *situation*, as defined in the situation ontology, is the moment starting with someone starting doing something, and it ends when the action ends, and another begins, or there is a jump in time or space. These situations can be anywhere: in a process, a story, a painting, a video or in anything else that represents a situation. There is an *actor* who does an *action*. In the same situation, the actor can have other actions going on too, and there can be other actors present, doing their actions. Apart from the actor and the action (that together form an *event*) there is also the surroundings, which consists of absolute and relative *time* and *place*, and possibly other *elements* of the situation. Also the *mood* and *theme* of the situation can be annotated.

When using the situation ontology, the philosophy is to leave the properties open if they don't exist or one doesn't know them. The ontology is meant mostly to be used for creating links and for helping in search.

Culture, for example art, is often much about interpretations (Holly, 1984). This means that when bringing culture resources available to people on the semantic web, interpretations cannot be left out of the system. For example, consider the painting “Kullervo departs for the war” in figure 2 depicting an event in Kalevala. In this case the interpretation that the painting is in particular about Kullervo (and not about some unknown man) departing for the war in Kalevala in order to take revenge on his enemies is of importance. Therefore we have made it possible to dis-

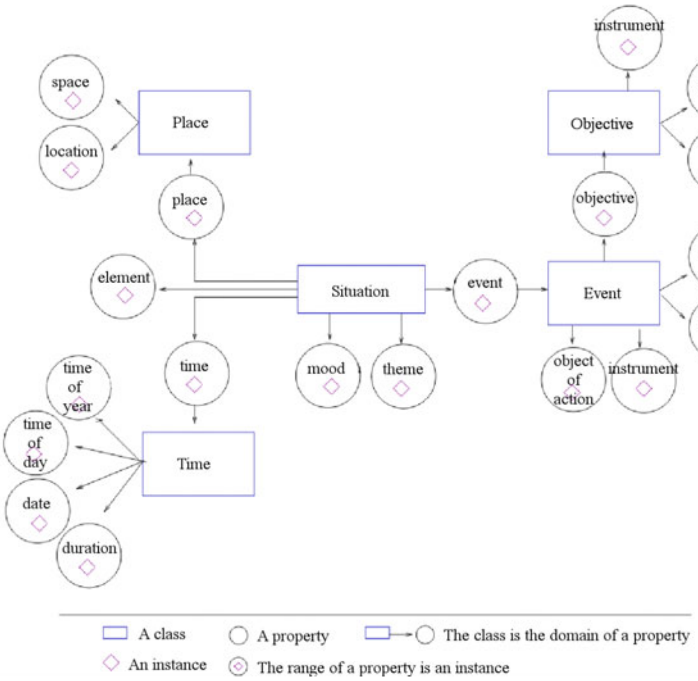
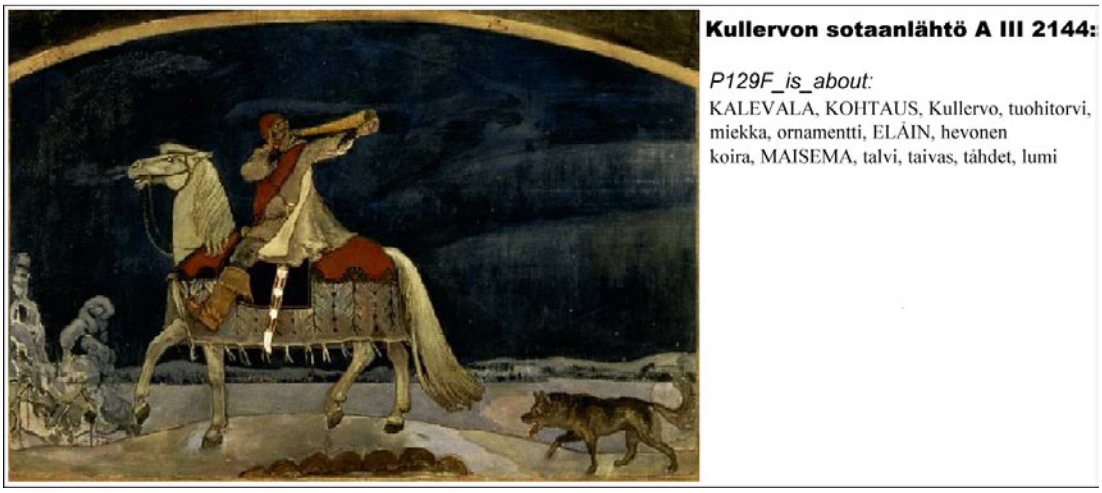


Figure 1: The situation ontology



Kullervon sotaanlähtö A III 2144:

P129F_is_about:
KALEVALA, KOHTAUS, Kullervo, tuohitorvi, miekka, ornamentti, ELÄIN, hevonen koira, MAISEMA, talvi, taivas, tähdet, lumi

Figure 2: Kullervo departs for the war. A painting at the Finnish National Gallery. On the right, the original keywords describing the content are given as the value of the CIDOC CRM (Doerr, 2003) property P129F_is_about: Kalevala, event, Kullervo, birch bark horn, sword, ornament, animal, horse, dog, landscape, winter, sky, stars, snow.

tinguish between factual information and interpretations in the annotations. Interpretations can be shown in a different way in the user interface or even left out, if the person looking for information wants to avoid interpretations.

3.1.1 Describing Action

In one *situation*, there can be many *events*. An event refers to one whole, that consists of an *actor*, an *action*, an *object of action* and an *instrument*. The actor, action and instrument are quite straightforward in their semantics, but the object of action can vary depending on the situation. The object can vary from being an object (e.g., hitting a ball), or a predicate (e.g., try to get up) to being an adverbial (e.g., sit on a couch). Because of this, the object of action is somewhat semantically ambiguous and cannot be used so easily for machine reasoning. However, when people use the system, the natural meaning of the relations can usually be understood easily by the user.

The action can also have an *objective*. The objective is an important part of the event, as what a person does may tell much less about what he is actually doing than knowing the objective. In a generic process description, the objective isn't used, as there it is taken for granted that doing one part of the process always has the objective of getting to the next part of the process. But when we look at stories, the objective becomes important. If we have a painting where a man is riding a horse, it is important to know that he is not only riding, but actually his objective is to depart for the war. We may know this objective from the name of the painting, like in the case of the painting "Kullervo departs for the war". When the objective has been annotated, this painting is found also when looking for information about war, not only about horses and riding.

The question about interpretation is quite apparent in the case of the objective. If the painter of "Kullervo departs for the war" would have given the name as an ironic joke for example, it would already require some knowledge about this to not make the wrong annotation. The same is true if the painting would not have a name at all, but still the objective is obvious by looking at the picture: there is a man with a sword on his side, he's blowing a big horn, etc. Especially when describing pictures much of metadata is based on the interpretations of the annotator. In textual stories, the objectives can actually be explained in the text.

Objectives can also exist on many levels. When we read the story about Kullervo in Kalevala we can see that the objective of him to go to the war is actually to take revenge on the people who killed his family. So he is riding to get to war to get revenge. That's three levels already, and more can sometimes be found.

3.1.2 Describing the Surroundings

Even though our main focus in this research is on the action and the processes, sometimes the surroundings of the situation are also interesting. They should not be left out of the metadata description of the situation. In the situation ontology, the surroundings of the situation are described by the *time* and the *place* of the situation (cf. figure 1). It may be important to know, where and when something happens. This can be relevant to generic process descriptions as well as to story situations.

Time and place can be modeled both on the level of absolute facts and from the relative point of view. For example, a situation described in a news article may have the absolute time (*date*) September 11th 2001, and the absolute place (*location*) New York. However, in many kinds of resources the absolute time and place are not known. For example, generic process descriptions have no absolute time or place, and many stories lack these facts, too. Thus the relative time and space are important. Relative time describes the *time of year* and *time of day* of the situation, and relative place (*space*) describes the surroundings on a more general level. The relative time and space of the example above could be "in autumn", "in daytime", and "in a city". When describing the slash and burn procedure, one could say that the "cutting down the trees" -situation happens in spring time in a forest. The *duration* of the situation can also be modeled by creating an instance of the class *Time*.

3.2 More Features

Apart from modeling events in time and space, the situation ontology allows the annotation of a *theme*, a *mood* and other *elements*. The theme and the mood are properties that have to do with all the non-generic situations, like parts of a story or a painting. The theme reflects the meaning of the situation from an intellectual point of view, whereas the mood reflects the emotional mood of the situation. The elements mean any objects or things of interest in the situation, that are not directly involved in the events, but are still somehow important to mention. These properties add to the flexibility of the model, even though they don't have to be used, and especially the two former need lots of interpretation from the annotator.

To further deepen the model, we have included some more features in a wrapper class called *Situa-*

tionElement. All the instances of the properties mentioned in the ontology are wrapped inside this class. In addition to the link to the content definition ontology, this class has the possibility to mark the information inside as an *interpretation*. With this information explicitly shown, the part of the information that is most subjective can be shown to the user in a different way in the user interface, or it can be left out if someone chooses to only be interested in definite facts.

There is also a property called *attribute* in the *SituationElement*-class. The annotator can add attributes to any elements of a situation. This may bring interesting additional information about some elements and can be used for answering questions such as ”How?” or ”What kind of...?”. Still another property is *symbolizes* that makes it possible to express the fact that something symbolizes something else. As a last addition, we have added the property *freeText*, so that everything annotated can also be given a literal value that can be used in the user interface.

3.3 The Process Ontology

The situation ontology can be used to describe the parts of a process or a story, but there has to be a logic for binding situations together in order to form large processes or stories. The process ontology was created for this. The process ontology was strongly influenced by the OWL Services (Coalition, 2003) processes part. The atomic processes of OWL-S could be seen as a close relative to the situations in our terminology, as these are the units used to build up bigger wholes. It was needed for easy specification of the ordering of the situations. The class *ControlConstruct* with its subclasses made this possible.

Some problems arose when annotating the order of the situations with Protege-2000[4] because this editor did not support the use of ordered lists, even though they are a part of the RDF specification. Some hardcoding was needed to get around this problem, taking away the possibility to reuse subprocesses as a part of other processes, but this can probably be solved in later versions of the ontology.

3.4 The Content Definition Ontology

The content definition ontology is the ontology that defines how things in the world relate to each other. The actions, the objects, the places, indeed everything that is present in situations that are being annotated, should be defined in a this large ontology that tells what are the semantic connections between the concepts.

As the processes are made up of situations, the situations are made up of different properties of the situation. All these properties have a value, that is a reference to a resource of the content definition ontology. For example, consider the situation where the trees are cut down in the slash and burn method. In the action-property of the situation, there will be a reference to the concept of cutting down, and the object of action is a reference to the concept of tree. With all the concepts and there relations defined, we can later create links between different processes that have to do with trees, even though in another process or situation, the annotation would have been oaks. The ontology tells the computer that oaks are a subclass of trees.

3.5 Semantic Difficulties Encountered

Some theoretical problems of using these ontologies were discovered during the ontology design process. One was that of interpretation. Even the same situation can be seen in many ways, depending on the point of view of the annotator. Things get even more subjective when properties such as mood are used. Still, we wanted to include these possibilities, as this metadata is meant primarily for people and not for machines, that can’t as easily cope with the fact that all facts are not really facts. We envision that interpretations of cultural resources can be very interesting to many end-users. It would be too a high price to pay for making the data totally fact-based.

Another interesting dilemma encountered was the relation between the thing being modeled and reality. What is the relation between a story character and a real historical character? There may be almost no differences, but the fact that the other one is only fictional. There is a relation between Alexander the Great and Aragorn of the “Lord of the rings”, but should the difference of the fictional world and the real world be emphasized or not?

The same problem exist with for example toys. A toy aeroplane has lots of semantic connections to real aeroplanes, but still it’s a totally different thing. And if there is an old tractor somewhere, that children use as a playing ground, should it be annotated as a tractor, a playing ground, or a space ship, as the children always think of it as one?

The impact of changes in time and space in the metadata and ontologies is also an interesting question. For example, an old axe may have been used in a

[4] http://protege.stanford.edu

medieval war, but in modern wars axes are rarely used as weapons. Thus the war related properties are only relevant to axes during a certain time in history. Axes are an easy example because axes have been fairly similar objects all through the history. In contrast, some concepts may mean a totally different thing in different eras. For example, the process of healing a wound with modern technology and knowhow is very different from what it was a thousand years ago. Also a change in place or culture can twist the meanings of some concepts very much. Depending on the viewpoint, these differences can be seen either as a problem or as a source of most interesting and complicated semantic connections.

4 CultureSampo I Prototype

In order to test whether the actions and processes indeed provide a useful basis for linking different cultural resources, the theory has to be tested. The CultureSampo I prototype was built on top of the MuseumFinland architecture to test the new point of view, and especially to see how different kinds of resources could be linked together.

The content of MuseumFinland was semantically homogenous, conforming to a shared metadata schema, consisting of mainly artifact metadata originating from heterogenous and distributed museum collections. In contrast, CultureSampo aims at publishing Finnish culture content of many kinds, conforming to different metadata schemas, on the semantic web. The idea is to allow the search of the resources through the actions they are involved in or involve, and provide the end-user with automatic semantic linking of the resources based on the content (Hyvönen et al., 2006).

4.1 Goals

CultureSampo shares the main goals of MuseumFinland:

1. Global view to distributed collections. It is possible to use the heterogeneous distributed collections of the museums participating in the system as if the collections were in a single uniform repository.

2. Content-based information retrieval. The system supports intelligent information retrieval based on ontological concepts, not only simple keyword matching as is customary with current search engines.

3. Semantically linked contents. A most interesting aspect of the collection items to the end-user are the implicit semantic relations that relate collection data with each other. In MuseumFinland, such associations are exposed to the end-user by defining them in terms of logical predicate rules that make use of the underlying ontologies and collection metadata.

4. Easy local content publication. The portal should provide the museums with a cost-effective publication channel.

CultureSampo I prototype was designed especially to deepen the third goal of the system, by bringing in the possibility to relate different kinds of cultural resources through semantically richer annotations, based on actions and processes. The need for this became evident when making MuseumFinland and was actually also echoed in the end-user feedback of MuseumFinland. One of the feedback e-mails brought us this wish:

"Are there any plans to give more detailed information about the objects? Now the items are laid on display: here they are, look at them. But it's not told to what, how and why the artifacts have been used....This kind of extra information would serve me at least. Some of the objects and their use is familiar to me, but not all."

Also the professionals in the museum field are pondering these questions. Trilce Navarrete writes the following (Navarrette, 2002):

"As museums expand the definition of 'education' there is a need to consider alternative models of explanation, such as oral history, mythology, family folklore and other ways to create the context in which stories are told — the story of each museum. How do museums go about fostering their community's narrative construction? The question in museums is not only how to better explain the story of the given object, but also how can museums better create and inspire the context for the public to construct an interest to relate to these stories?"

These two quotes outline nicely the field of problems that we tried to approach with the help of CultureSampo I, using the ideas described earlier in this paper.

4.2 A Use Case

To illustrate the intended usage of CultureSampo, consider the following use case. John is a 12-year-old pupil of a school in Helsinki. The teacher has given him and his friend Michael the task of doing an exercise together. John and Michael should do their work about agriculture in Finland in the nineteenth century. The work should be broad in content, and contain pictures and other materials, too.

The objective of John is to get information about agriculture in the nineteenth century. Michael wants to write about the farm animals, so John has to consider the part about agriculture and people's lives in general. The teacher gives John a link to CultureSampo to find some information. John goes to the site, looks at the search window, ends up with choosing the search categories "Agriculture" and "The nineteenth century", and he gets a link to a scathe that has been used in Helsinki in the nineteenth century. He copies the picture, it's good material. Then he finds a link to the slash and burn process, which is a process where a scathe has been used. He gets information about this kind of farming, and decides to write about that. He also finds a link to a painting where a mythic figure from the Finnish epic Kalevala is using the slash and burn method, and from that he finds a link to the right point in the poem itself, and gets exited about the dramatic twists. The poem also gives him some insight to the life in ancient Finland, and he writes about the jobs he finds people were doing in the poem, like shepherding and fishing. On the basis of the information of CultureSampo, John and Michael manage to make a good school essay about agriculture.

4.3 Illustration of Semantic Linking

Figure 3 illustrates semantic linking in CultureSampo I by a screenshot. The page shows metadata about the painting "Kullervo curses" by Akseli Gallen-Kallela depicting a famous event in Kalevala. The painting itself is shown on the left and its metadata in the middle. The semantic links appear on the right. They include different kinds of links, depending on the logical rules that have created them. The labels of the links and headlines explain why following this link should be of interest of the user. Three semantic recommendation links created by the system are visualized on top of the screenshot. One link points to an artifact in the collections of the National Museum, one to bibliographic information of the artist, and one to the actual point in Kalevala where the event takes place. Different kinds of content items have pages of their own similar to this painting page. They look different because of the differences in content types and content on them, but in all cases the content item itself is shown on the left, the metadata in the middle, and the semantic links on the right.

In addition to semantic browsing, the system also supports faceted semantic search in the same way as MuseumFinland.

4.4 Cultural Content

In the prototype seven kinds of content resources were used. We selected resources that were somehow important for the Finnish culture, and that related to each other, so that links could emerge between them if described correctly with the annotation ontologies. We also wanted to have examples of different mediums present in the resources.

1. *Processes* were one of the core things we wanted to have in the system, as they have a lot to do with action and with culture. Preserving information about cultural processes is also important in itself. We had the slash and burn procedure and seine fishing as examples of old Finnish processes that are not any more very well known to modern Finns. These two processes were built up from about ten situations each.

2. *Stories* were the other resource type that had been much thought of when designing the CultureSampo ontologies, and is an inspiring way to describe the context of old Finnish culture. Kalevala was a natural choice when choosing what stories to include in the prototype. As the Kalevala stories are old, the world they describe is close to the context of life in the past agrarian Finland. Two quite separate entities were chosen from the whole epic: the sad story of Kullervo whose family was killed, and another where the most famous hero of Kalevala, Väinämöinen, learns how to farm land by the slash and burn method. These stories had a start and an end of their own, and they had good references to the other resources we had.

3. *Paintings* were the first resource where no text was included, but the information was in picture form. It was good to add paintings not only because of getting to test describing this medium, but also as visual resources give another kind of life to the context of old Finnish culture. We selected some paintings about Kalevala to fit the stories we had, and also some other paintings describing the old days.

4. *Photographs* are quite similar to paintings w.r.t. their content annotation. Still, the documentary aspect of photographs is much stronger than in

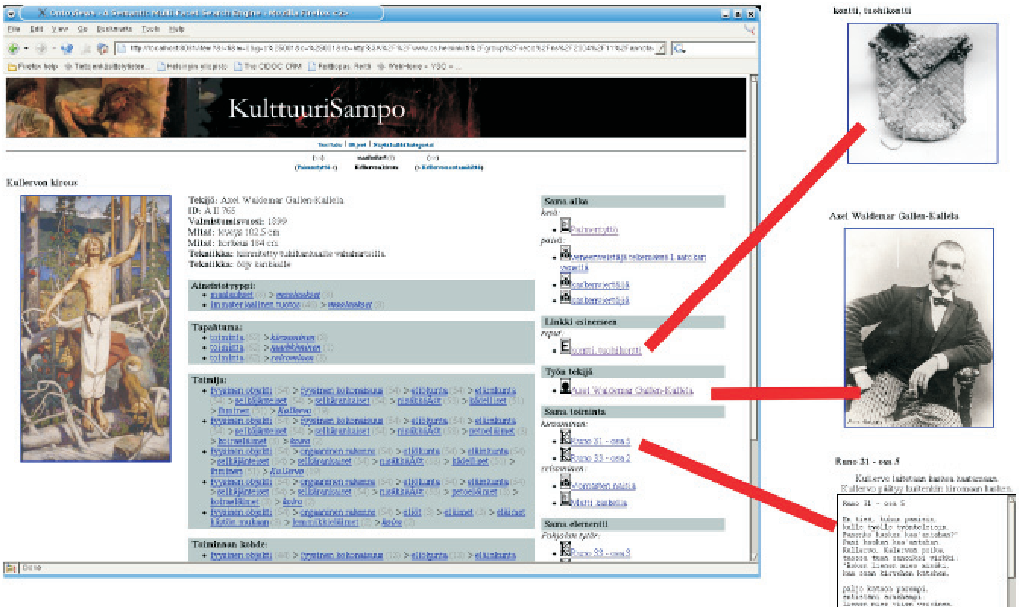


Figure 3: An example of the interface of CultureSampo I illustrating a painting at the Finnish National Gallery about an event in the epic Kalevala. Three semantic recommendation links created by the system are visualized on top of the screenshot.

paintings, so we believe they bring a good authentic point of view to old processes.

5. *Videos* were an interesting and complicated case of resources. At this stage of the process, we didn't yet implement very complicated metadata descriptions of the videos we had, even though using the situation ontology with moving image and sound could be used in a more sophisticated way. At this time it was enough to get some links to the other resources through a simple metadata description, and the logics can later be improved.

6. *People* were added as one kind of "resources", as information about people who lived in the old days can be relevant, when forming the context of cultural resources. Since the core of our metadata schema is action, and the actor's are usually people, it was natural to use people as content items. Most of the persons in the system were artists who had created the artworks included. However, there were also a couple of fictitious people from the Kalevala epic.

7. *Artifacts* were a natural kind of resource we needed to have, because artifacts are important in actions (e.g., as instruments), and we also had lots of metadata about them in the MuseumFinland knowledge base.

The CultureSampo I prototype uses ontologies designed for action-based description of cultural resources. The resources we used in the prototype were 2 processes, which were built from 21 situations, 2 stories of Kalevala in 32 situations, 8 paintings, 4 photographs, 2 educational videos and 14 museum objects. The annotation of the situations was quite slow, as there were no existing action-based metadata available, and no special tools designed to help in the process. Most situations could be described in the way we wanted, so the expressive power of the situation ontology proved out to be satisfactory.

A couple of problems arose in describing the situations of stories. One was a situation where a woman reads a spell to protect the cows when sending them to the forest. This was a long monologue, and the contents of the monologue could not be made into situations, but we could only say the woman is reading a spell. This shows a problem of describing speech in stories and in general: the things that people talk about are not necessarily true or real actions. In our scheme there is no notion of reification, although this is in principle supported by the RDF recommendation. As a result, dialogues or monologues are difficult to describe with our action-based annotations as it is. When describing some parts of the story, we needed to limit the level of detail. All the actions were not described in situations, but only the ones that were interesting and mattered in the whole plot.

In describing processes, there were some problems with representing time. As different situations of a process take different amounts of time, choosing what is important in the process sometimes ended up with situations that are very uneven in their lengths. For example, cutting the trees down is fast but waiting for a year or letting the forest grow back take longer times.

These problems were not critical to the functionality of the system. Most of the situations in different resources were straightforward to annotate with the ontologies. When using the prototype, it could be seen that actions really do link different cultural resources together. Lots of links were created with the logical rules we defined, and the links really had meaning for the end-user. When looking at an axe in a museum, you get a link to the process of slash and burn where axes were used, to a painting where someone is using an axe, and to a poem of Kalevala were Kullervo is sharpening his axe.

The situation ontology was the main object of our interest, but also the other ontologies worked well enough. The process ontology was expressive enough to describe the order of the situations we had. As the content defining ontology we used a preliminary version of the General Finnish Upper Ontology YSO (Hyvönen et al., 2005b), that was being designed at the same time. As it wasn't totally ready, we made a version of it that had the concepts we needed, and it worked at this stage of the process.

5 Related Work

Process models have been researched a lot in computer science, though usually from the point of view of machine-executable processes or business processes. As examples of these include the Process Specification Language (Schlenoff et al., 2000), Business Process Execution Language for Web Services (Andrews et al., 2003) and OWL Services (Coalition, 2003), that we used as a starting point when designing our process ontology. However, these approaches to process modeling were not designed for describing cultural processes, so the end results are quite different, even though some similarities exist. Questions about actions happening in time are always somehow involved in processes.

There are ontologies that have been designed for describing cultural resources. The CIDOC CRM (Doerr, 2003) is an upper level ontology that has been designed for making different metadata schemas and content interoperable. The ABC Harmony (Lagoze and Hunter, 2001) is a related effort intended to provide a common conceptual model to facilitate interoperability among application metadata vocabularies. Our approach is different from these ontologies and other metadata systems that concentrate on describing the cultural resources in collections. Our focus is on describing the actions and processes that relate the resources in the real world (or in fiction), with goal of providing the user with insightful semantic recommendations and enhancing search.

6 Conclusions

This paper is about describing the cultural context of resources. As the common denominator of different cultural resources, a situation including some actions was chosen. Situations describe what happens in a painting, in a story, or in a cultural process. Such descriptions provide a semantic network linking related cultural content items together. This point of view complements the more traditional idea of using classifications and ontologies for defining cultural concepts and metadata schemas.

An experimental situation ontology was created to provide the common annotation scheme to describe situations and their main properties, whatever the medium in which the situations appear is. Also a process ontology and a content defining ontology were taken as a part of model, in order to link different situations together into larger wholes, and in order to define the concepts appearing in the situations unambiguously.

It was clear from the beginning that describing cultural content in terms of real world actions and processes is more complicated than describing cultural objects in collections. The prototype CultureSampo I was designed and implemented in order to find out both the practical and theoretical problems of the approach, and also to see if the benefits of this kind of metadata in use could be as useful as we hoped. The first results seem promising. However, the content set used in this experiment was very small and it was pre-selected according to our goals. Experiments with larger content sets and annotations are definitely needed.

The final goal of the CultureSampo project

(Hyvönen et al., 2006) is be become a demonstration of a nation-wide cross-domain cultural publication for the semantic web. As a first step towards this ambitious goal, the work of this paper shows that a situation ontology and action-based metadata descriptions can bind together different kind of cultural resources in meaningful ways, from paintings and objects to processes and stories.

References

T. Andrews, F. Curbera, H. Dholakia, Y. Goland, J. Klein, F. Leymann, K. Liu, D. Roller, D. Smith, S. Thatte, I. Tricovic, and S. Weerawarana. Business process execution language for web services - version 1.1, 2003.

Aristotle. *Poetics*. Focus Philosophical Library, Pullins Press, 2006.

The OWL Services Coalition. *OWL-S: Semantic Markup for Web Services*, November 2003. http://www.daml.org/services/owl-s/1.0/owl-s.pdf.

M. Doerr. The CIDOC CRM - an ontological approach to semantic interoperability of metadata. *AI Magazine*, 24(3):75–92, 2003.

M. Eysenc and M. Keane. *Cognitive Psychology A Students Handbook*. Psychology Press, Exeter, UK, 2002.

M. Holly. *Panofsky and the foudations of art history*. Cornell University Press, Ithaca, 1984.

B. Holmes, B. Tangney, A. FitzGibbon, T. Savage, and S. Meehan. Communal constructivism: Students constructing learning for as well as with others. In *Proceedings of SITE 2001, Florida*, 2001.

E. Hyvönen, E. Mäkela, M. Salminen, A. Valo, K. Viljanen, S. Saarela, M. Junnila, and S. Kettula. MuseumFinland – Finnish museums on the semantic web. *Journal of Web Semantics*, 3(2):224–241, 2005a.

E. Hyvönen, T. Ruotsalo, T. Häggström, M. Salminen, M. Junnila, M. Virkkilä, M. Haaramo, T. Kauppinen, E. Mäkelä, and K. Viljanen. CultureSampo — Finnish culture on the semantic web. The vision and first results. In *Semantic Web at Work - Proceedings of the 12th Finnish Artificial Intelligence Conference STeP 2006, Volume 1.*, Nov 2006.

E. Hyvönen, A. Valo, V. Komulainen, K. Seppälä, T. Kauppinen, T. Ruotsalo, M. Salminen, and A. Ylisalmi. Finnish national ontologies for the semantic web - towards a content and service infrastructure. In *Proceedings of International Conference on Dublin Core and Metadata Applications (DC 2005)*, Nov 2005b.

M. Junnila. Tietosisältöjen semanttinen yhdistäminen toimintakuvausten avulla (Event-based approach to semantic linking of data content). Master's thesis, University of Helsinki, March 6 2006.

L. Kelly. Developing access to collections through assessing user needs, May 1999. Museums Australia Conference, Albury.

S. Kettula. Käsityöprosessit museossa semanttisen tiedonhaun lähteenä ja kohteena. (Handicraft processes in museum as a source of object for semantic content and search). In L. Kaukinen and M. Collanus, editors, *Tekstejä ja kangastuksia. Puheenvuoroija käsityöstä ja sen tulavaisuudesta. (Views of handicraft and its future)*. Artefakta 17, Juvenes Print, 2005.

C. Lagoze and J. Hunter. The ABC Ontology and Model. *Journal of Digital Information*, 2(2), 2001.

E. Mäkelä, E. Hyvönen, and S. Saarela. Ontogator — a semantic view-based search engine service for web applications. In *Proceedings of the 5th International Semantic Web Conference (ISWC 2006)*, Nov 2006.

E. Mäkelä, E. Hyvönen, S. Saarela, and K. Viljanen. Ontoviews – a tool for creating semantic web portals. In *Proceedings of 3rd International Semantic Web Conference (ISWC 2004), Hiroshima, Japan*, November 2004.

T. Navarrette. Museums in the street: Cultural creation in the community, October 2002. INTERCOM Conference Leadership in Museums: Are our core values shifting?, Dublin, Ireland.

M. Salminen. Kuvien ja videoiden semanttinen sisällönkuvailu (Semantic content description of images and videos). Master's thesis, University of Helsinki, May 2006.

C. Schlenoff, M. Gruninger, F. Tissot, J. Valois, J. Lubell, and J. Lee. The process specification language(psl): Overwiev and version 1.0 specification, 2000. NISTIR 6459, National Institute of Standards and Technology, Gaithersburg, MD.

J. Sowa. *Knowledge Representation. Logical, Philosophical, and Computational Foundations*. Brooks/Cole, 2000.

K. Viljanen, T. Känsälä, E. Hyvönen, and E. Mäkelä. Ontodella - a projection and linking service for semantic web applications. In *Proceedings of the 17th International Conference on Database and Expert Systems Applications (DEXA 2006), Krakow, Poland*. IEEE, September 4-8 2006.

CultureSampo—Finnish Culture on the Semantic Web: The Vision and First Results

Eero Hyvönen, Tuukka Ruotsalo, Thomas Häggström,
Mirva Salminen, Miikka Junnila, Mikko Virkkilä, Mikko Haaramo,
Eetu Mäkelä, Tomi Kauppinen, and Kim Viljanen
*Semantic Computing Research Group
Helsinki University of Technology (TKK), Laboratory of Media Technology
University of Helsinki, Department of Computer Science
http://www.seco.tkk.fi/
first.last@tkk.fi

Abstract

This paper concerns the idea of publishing heterogenous cultural content on the Semantic Web. By heterogenous content we mean metadata describing potentially any kind of cultural objects, including artifacts, photos, paintings, videos, folklore, cultural sites, cultural process descriptions, biographies, history etc. The metadata schemas used are different and the metadata may be represented at different levels of semantic granularity. This work is an extension to previous research on semantic cultural portals, such as MuseumFinland, that are usually based on a shared homogeneous schema, such as Dublin Core, and focus on content of similar kinds, such as artifacts. Our experiences suggest that a semantically richer event-based knowledge representation scheme than traditional metadata schemas is needed in order to support reasoning when performing semantic search and browsing. The new key idea is to transform different forms of metadata into event-based knowledge about the entities and events that take place in the world or in fiction. This approach facilitates semantic interoperability and reasoning about the world and stories at the same time, which enables implementation of intelligent services for the end-user. These ideas are addressed by presenting the vision and solution approaches taken in two prototype implementations of a new kind of cross-domain semantic cultural portal "CULTURESAMPO—Finnish Culture on the Semantic Web".

1 Towards Semantic Cross-domain Interoperability

A widely shared goal of cultural institutions is to provide the general public and the researchers with aggregated views to cultural heritage, where the users are able to access the contents of several heterogenous distributed collections of institutions *simultaneously*. In this way, the organizational and technical obstacles for information retrieval between collections and organizations, even between countries and languages could be crossed.

Content aggregation may occur at the syntactic or semantic level. The basis for *syntactic interoperability* is sharing syntactic forms between different data sources, i.e., the metadata schemas such as the Dublin Core Metadata Element Set[1] or the Visual Resource Association's (VRA) Core Categories[2]. Such schemas make it possible to identify different aspects of the search objects, such as the "author", "title", and "subject" of a document, and focus search according to these. Syntactic interoperabilty facilitates, for example, multi- or metasearch[3]. Here the user types in a query in a metaportal. The query is then distributed to a set of underlying systems and the results are aggregated for the end-user. For example, the Australian Museums and Galleries Online[4] and Artefacts Canada[5] are multi-search engines over nation-wide distributed cultural collections. Here the content includes metadata about museum artifacts, publications etc. represented using shared metadata schemas.

Content aggregation at the *semantic level* means that not only the form of the data is shared and in-

[1] http://dublincore.org/documents/1998/09/dces/

[2] http://www.vraweb.org/vracore3.htm

[3] http://en.wikipedia.org/wiki/Metasearch_engine

[4] http://www.amonline.net.au/

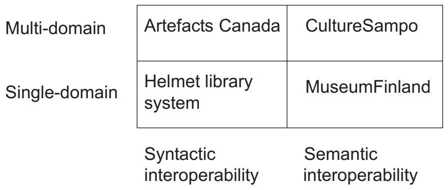
[5] http://www.chin.gc.ca/

teroperable, but also the values used in the metadata schema, and that the meanings of the values are semantically defined in terms of ontological structures. The values of metadata fields, such as authors, material types, and geographical locations are taken from a set of shared vocabularies, i.e., ontologies, or if different vocabularies are used, then the mappings between them are available. At this level of content aggregation, reasoning about the ontological relations between content items becomes possible enabling semantic search, semantic browsing, recommendations, explanations, and other "intelligent" services. A prototypical example of this approach is the portal "MuseumFinland—Finnish Museums on the Semantic Web"[6] (Hyvönen et al., 2005a), where distributed, syntactically heterogeneous museum collection databases are integrated by a set of seven shared ontologies, and semantic search and browsing services are provided to end-users based on the aggregated knowledge base.

Another distinctive feature between cultural content aggregation systems is whether they deal with metadata that conforms to a *single metadata schema* or *multiple schemas*. For example, the Helmet library system[7] aggregates public library collections of neighboring cities for the public by using a single metadata format. In the same vein, an online record shop may deal with CD/DVDs whose metadata is represented in a homogeneous way. On the other hand, in a system such as Artefacts Canada, the underlying databases contain items of different kinds, such as art, furniture, photos, magazines etc. whose metadata conform to different schemas. For example, a metadata field representing physical the material of an object is essential for a piece of furniture or artifact but not for a publication.

Semantic web portals have tackled the problem of semantic interoperability usually by sharing metadata schemas. For example, in MuseumFinland heterogeneous artifact collection databases were made semantically interoperable, but the content was of a single domain (artifacts), and the metadata was based on a single, Dublin core like schema of artifacts. There are paintings and some other forms of art in MuseumFinland collections, but they have been cataloged as pieces of artifacts in the cultural museums participating in the portal, and not as pieces of art. The reasoning routines were based on the annotation schema and the ontologies.

In this paper we investigate the problem of *semantic cross-domain interoperability*, i.e. how content of different kinds conforming to multiple metadata schemas could be made semantically interoperable. The focus is the cultural domain and content types studied in our work include artifacts, paintings, photographs, videos, audios, narratives (stories, biographies, epics), cultural processes (e.g., farming, booth making), cultural sites, historical events, and learning objects. In this case, the content is cross-domain in nature and, as a result, comes in forms that may be quite different from each other. Mapping them into a Dublin Core like generic metadata framework is problematic. Instead, we propose content aggregation at a semantically more foundational and rich level based on events and thematic roles (Sowa, 2000). The research is being carried out not only in theory, but by implementing real portal prototypes. More specifically, we show how the idea of MuseumFinland can be extended into a cross-domain semantic cultural portal called "CultureSampo—Finnish Culture on the Semantic Web". Figure 1 illustrates the positioning of CultureSampo along the distinctions discussed above and its relation to some other portal systems mentioned.



Multi-domain	Artefacts Canada	CultureSampo
Single-domain	Helmet library system	MuseumFinland
	Syntactic interoperability	Semantic interoperability

Figure 1: Portals can be classified in terms of the number of metadata schemas used (vertical axis) and the level of interoperability (horizontal axis).

In the following we first state the vision and goals of creating CultureSampo. After this problems of obtaining semantic cross-domain interoperability are discussed and the idea of using event-based descriptions is proposed as a solution. The discussion is based on experiences gathered in creating two experimental prototypes of CultureSampo. In conclusion, contributions of the work are summarized and directions for further research are proposed.

2 The Vision and Goals of CultureSampo

CultureSampo shares the general goals of MuseumFinland:

[6] This application is operational at http://www.museosuomi.fi with a tutorial in English.

[7] http://www.helmet.fi

Global view to heterogeneous, distributed contents The portal supports the usage of heterogenous and distributed collections and contents of the participating organizations as if there were a single uniform repository.

Intelligent end-user services The system supports *semantic search* based on ontological concepts and *semantic browsing*, where semantic associations between search objects are exposed dynamically to the end-user as recommendation links with explicit explanations. These links are defined in terms of logical rules that make use of the underlying ontologies and collection metadata.

Shared content publication channel The portal should provide the participating organizations with a shared, cost-effective publication channel.

CULTURESAMPO focuses, from the content perspective, especially on material related to the "Golden Era" of the Finnish culture in the 19th century. During this period the notion of Finland as a nation with an original cultural background and history was formed, and the development resulted in the independence of the country in 1917.[8] A central component of the Finnish cultural heritage has been the national epic Kalevala[9]. It was published originally in 1835 and has been translated into some 60 languages. This epic, based on large collections of folklore[10] collected especially in the eastern parts of Finland, Karelia, has been a continuous source of inspiration in Finnish fine arts, music, sculpture, literature, and other branches of culture. The world of Kalevala also nicely relates to the original agrarian Finnish life and artifacts that are available in museums.

In CULTURESAMPO the Karelian culture is central also because one goal of the work is to reunite Karelian collections using semantic web techniques. These collections have now been distributed in several museums due to the result of the World War II where eastern parts of Finland were annexed to the Soviet Union. The semantic web provides means for re-uniting cultural entities virtually on the semantic web. The problem of distributed cultural heritage due to wars and other reasons is very common in Europe. We envision, that the ideas and techniques developed in CULTURESAMPO could later contribute to creation of cross-national and multi-lingual cultural portals, a kind "CultureEurope".

The system will also contribute, in a sense, to the tradition of Kaleva translations. It provides first excerpts of Kalevala translated, not into natural languages for the humans to use but for the machine to "understand" in the formal languages of the semantic web, RDF and OWL.[11]

The latter part of the portal name "Sampo" is the name of the mythical machine-like entity of the Kalevala epic. Sampo gives its owner power, prosperity, everything, but its actual construction and nature is semantically ambiguous and remains a mystery — tens of academic theories about its meaning have been presented. CULTURESAMPO adds still another modern interpretation of what a "Sampo" could be based on the semantic web.

3 Making a Cultural Portal More Intelligent

A major focus of our work in CULTURESAMPO is to study how to provide the end-user with intelligent search and browsing services based on semantically rich cross-domain content originating from different kind of cultural institutions. For example, consider the painting "Kullervo departs for the war" in figure 2 depicting an event in Kalevala. From the end-users' viewpoint, it could probably be interesting, if this piece of art could be linked with other paintings and photos, with the war theme in art museums, with weapons and accessories in cultural museums, with academic studies about Kalevala and Kullervo, with information about dogs and horses in the museum of natural history, with other (external) information on the web about Kullervo, with the actual poem in Kalevala and related pieces of folk poetry, with movies and videos on a media server, with biographies of the artist, and so on. An interesting line of associations could be created by considering the events, processes, and the Kalevala story that takes place in the picture. In this way, for example, the painting could be linked with content concerning the next or previous events in the Kalevala story. Such associations and viewpoints could be insightful, useful, and even entertaining both when searching for content and when browsing it.

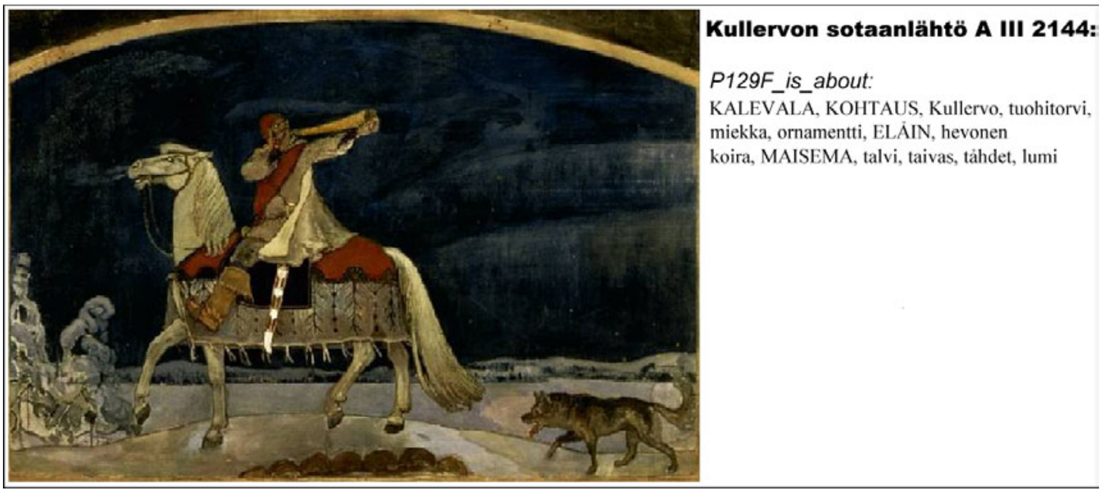
To investigate and test the feasibility of this idea in practise, we are extending the portal MUSEUMFINLAND into CULTURESAMPO by a sequence of new prototypes. In 2005, the first prototype to be called "CULTURESAMPO I" was designed and

[8] Before that Finland had been a part of Sweden (until 1809) and Russia (1809-1917).

[9] http://www.finlit.fi/kalevala/index.php?m=163&l=2

[10] http://www.finlit.fi/english/kra/collections.htm

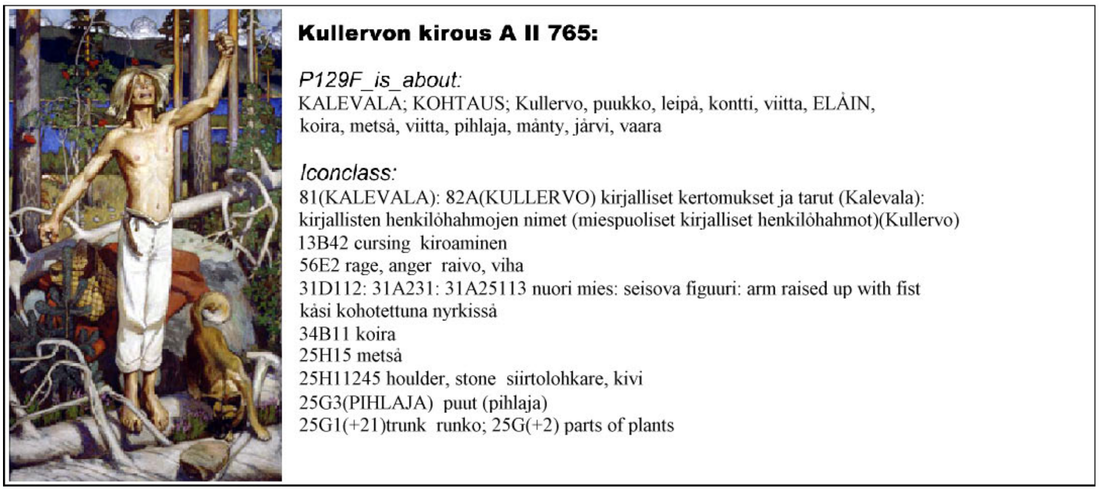
[11] http://www.w3.org/2001/sw/



Kullervon sotaanlähtö A III 2144:

P129F_is_about:
KALEVALA, KOHTAUS, Kullervo, tuohitorvi,
miekka, ornamentti, ELÅIN, hevonen
koira, MAISEMA, talvi, taivas, tåhdet, lumi

Figure 2: Kullervo departs for the war. A painting at the Finnish National Gallery.



Kullervon kirous A II 765:

P129F_is_about:
KALEVALA; KOHTAUS; Kullervo, puukko, leipå, kontti, viitta, ELÅIN,
koira, metså, viitta, pihlaja, månty, jårvi, vaara

Iconclass:
81(KALEVALA): 82A(KULLERVO) kirjalliset kertomukset ja tarut (Kalevala):
kirjallisten henkilȯhahmojen nimet (miespuoliset kirjalliset henkilȯhahmot)(Kullervo)
13B42 cursing kiroaminen
56E2 rage, anger raivo, viha
31D112: 31A231: 31A25113 nuori mies: seisova figuuri: arm raised up with fist
kåsi kohotettuna nyrkisså
34B11 koira
25H15 metså
25H11245 houlder, stone siirtolohkare, kivi
25G3(PIHLAJA) puut (pihlaja)
25G1(+21)trunk runko; 25G(+2) parts of plants

Figure 3: The painting “Kullervo cursing” and its metadata from the Finnish National Gallery.

implemented (Junnila et al., 2006; Junnila, 2006; Salminen, 2006). Figure 3 depicts a painting and its metadata in CULTURESAMPO I. The metadata shown originates from the Finnish National Gallery[12] and describes the subject of the painting in the following way: First, the CIDOC CRM[13] (Doerr, 2003) property `P129F_is_about` lists the following set of keywords (in Finnish): "Kalevala", "event", "Kullervo", "knife", "bread", "knapsack", "robe", "animal", "dog", "forest", "rowan", "pine", "lake", and "mountain". Second, the property "Iconclass" lists a set of ICONCLASS[14] (van den Berg, 1995) notations (categories) describing the subject. This description is partly redundant with the Finnish keywords.

In figure 4 this painting is viewed in CULTURESAMPO I. On the right column, a set of semantic links to other search objects are recommended with explanations created by the logical linking server Ontodella (Viljanen et al., 2006). The figure illustrates a link to a knapsack in the collections of the National Museum of Finland[15], a link to a biography of the artist, and a link to the point in the Kalevala epic where the event of the painting actually takes place.

CULTURESAMPO I was implemented using the same framework as MUSEUMFINLAND, i.e., the OntoViews framework (Mäkelä et al., 2004) including the view-based semantic search engine Ontogator (Mäkelä et al., 2006) and Ontodella (Viljanen et al., 2006). However, in this case much richer cross-domain metadata was used. The test material was limited in size but included examples of artifacts, paintings, photos, videos, biographical information, and narratives such as poems of Kalevala, and descriptions of traditional agrarian processes, such as farming by the slash and burn method.

During this experiment we identified two major obstacles for creating cross-domain semantic cultural portals:

Semantic Interoperability of metadata schemas. The problem of integrating metadata schemas occurs 1) *horizontally* when integrating schemas of different form semantically and 2) *vertically* when integrating content annotated at different levels of granularity.

Expressive power of metadata schemas. A central research hypotheses underlying CULTURESAMPO is that, from the end-user's viewpoint, different processes and events that take place in the society and history should be used as a kind semantic glue by which "insightful semantic links" could be created for the user to browse. This idea was already tested to some extent in MUSEUMFINLAND by creating an artificial event view for the end-user, and by mapping contents of it using logical rules. However, it seemed that a richer and a more accurate knowledge representation method was needed in annotating the contents than traditional metadata schemas.

In the following, our approaches to addressing these problems are outlined.

4 Semantic Interoperability of Metadata Schemas

Re-consider the figure 2. Its metadata may tell e.g. that this painting was created by A. Gallen-Kallela in 1901 in Helsinki. This metadata can be represented, by using RDF triples in Turtle notation[16], in the following way (this example is not based on the actual metadata but is for illustration only):

```
:Kullervo_departs_war
   dc:creator persons:A.Gallen-Kallela ;
   dc:date "1901" ;
   dc:spatial places:Helsinki .
```

The metadata record in a biographical repository, such as the ULAN[17] of the Getty Foundation, may tell us more about the artist in a very different metadata format, e.g.:

```
persons:A.Gallen-Kallela
   :placeOfBirth places:Pori ;
   :timeOfBirth "1865" ;
   :placeOfDeath places:Stockholm ;
   :timeOfDeath "1931" .
```

A problem here is that the number of different properties in metadata schemas easily gets large in cross-domain applications. Furthermore, the meaning of many properties, such as `dc:date` and `dc:spatial` in the metadata schema of paintings and `timeOfBirth/Death` and `placeOfBirth/Death` in the biographical metadata schema of persons may share some meaning, but are still different. We soon realized that when using the schemas for reasoning tasks, the logical rules accounting properly all kinds of combinations

[12] http://www.fng.fi

[13] http://cidoc.ics.forth.gr/

[14] http://www.iconclass.nl

[15] http://www.nba.fi/en/nmf

[16] http://www.dajobe.org/2004/01/turtle/

[17] http://www.getty.edu/vow/ULANSearchPage.jsp

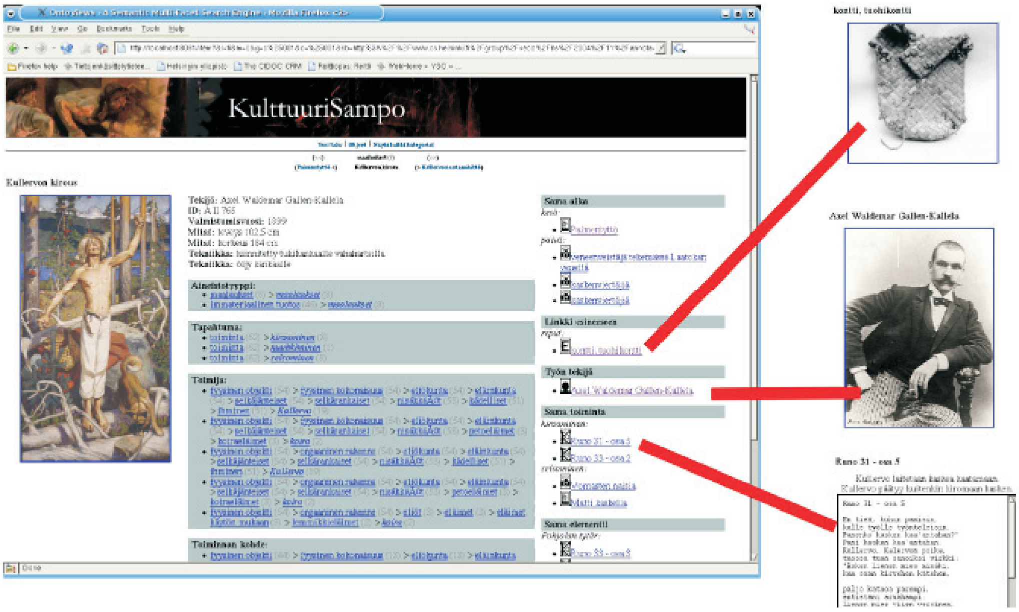


Figure 4: The painting of figure 3 viewed in the semantic portal CULTURESAMPO I. Three semantic recommendation links created by the system are visualized on top of the screenshot.

of properties become complicated, and the number of rules becomes large due to combinatorial explosion. It seems that a more primitive representation of knowledge than traditional metadata schemas is needed for reasoning.

A potential solution approach to solve the problem is to use the CIDOC CRM ontology. The system "provides definitions and a formal structure for describing the implicit and explicit concepts and relationships used in cultural heritage documentation"[18]. The framework includes some 80 classes, such as "E22 Man-Made Object", "E53 Place", and "E52 Time-Span", and a large set of some 130 properties relating the entities with each other, such as "P4 Has Time-Span" and "P87 Is Identified By". Interoperability of cultural content can be obtained by mapping metadata standards to CIDOC CRM.

The focus in CIDOC CRM is in modeling concepts necessary for representing the documentation semantics of different metadata schemas used in the cultural domain, such as Dublin Core. In contrast, in CULTURESAMPO our main focus is to represent *real world knowledge* related to cultural heritage, e.g., the subjects that the paintings in figures 2 and 3 depict. For this purpose, a different kind of knowledge representation scheme and large domain ontologies containing tens of thousands of domain concepts and events are needed.

Our solution to the problem of semantic interoperability is to transform different metadata schemas into a shared, more primitive knowledge representation of the real world. In this way, the meaning of `dc:date`, `:timeOfBirth` and `:timeOfDeath` can be made interoperable. By basing reasoning on the more primitive representation, more generic and fewer rules operating a smaller set of properties can be devised. As for the knowledge representation scheme, the idea of representing knowledge in terms of actions and thematic relations between actions and entities was adopted. This general idea has been applied widely in computational linguistics and natural language processing (cf. e.g. (Zarri, 2003)), in knowledge representation research (Sowa, 2000), and also in CIDOC CRM, where events are of central importance, too.

For example, in CULTURESAMPO the three time-relations of the above examples are reduced into only one time-relation relating an instance of an event type, such as "painting_event", "birth_event", or "death_event" to a time entity. The meaning of semantically complex properties in metadata schemas

[18] http://cidoc.ics.forth.gr/

is essentially represented in terms of different events and related entities. For example, the metadata about the painting "Kullervo departs for the war" means that there was a painting event related with A. Gallen-Kallela, the year 1901, and Helsinki by the thematic roles "agent", "time", and "place":

```
:painting_event_45
   rdf:type cs:painting_event ;
   cs:agent persons:A.Gallen-Kallela ;
   cs:time "1901" ;
   cs:place places:Helsinki .
```

Information about the artist's birth and death dates can be transformed in a similar manner into birth and death events, respectively. In this way, we can not only eliminate various time-related properties from the knowledge base but also aggregate knowledge form different sources on the more primite knowledge representation level. In this case, for example, event-based biographical knowledge about the life events of A. Gallen-Kallela can be enriched with the knowledge about the paintings he painted.

Solving the semantic interoperability problem of metadata schemas by using a primitive event-based knowledge representation scheme was one of the major challenges in creating the CULTURESAMPO II prototype in 2006. This idea will be described, especially from the semantic browsing viewpoint, in more detail in (Ruotsalo and Hyvönen, 2006).

5 Extending Semantic Representational Power of Metadata Schemas

The idea of using event-based knowledge representations in annotation provides also a solution for creating semantically richer annotations. Event-based annotations have been studied before, e.g., in the context of annotating the subject of photographs (Schreiber et al., 2001) and in representing narratives (Zarri, 2003).

To illustrate this idea, re-consider the painting "Kullervo departs for the war" of figure 2. The subject of content is here annotated by a set of keywords (in Finnish) including "Kullervo", "horse" and "dog". A problem from the knowledge representation viewpoint is that the mutual relations of the subject annotations are not known. For example, it is not known whether Kullervo rides a horse, a dog, both of them, or none of them. It is also possible that the dog rides Kullervo, and so on. Events can be used for elaborating the description, if needed, by specifying values for their thematic roles. In this case, for example, Kullervo would be in the agent role and the horse in the patient role in a riding event. This kind of information can be essential when searching the contents (e.g. to distinguish between riders and riding entities) or when providing the end-user with semantic links and explanations (e.g. to distinguish links to other riding paintings in contrast to other horse paintings).

In CULTURESAMPO content comes not only in different forms but is also annotated at different levels of detail "vertically". For example, the metadata from a museum database is given as it is and may contain only minimal metadata while some other content may be described in a very detailed manner by using lots of Iconclass notations or manually annotated events. In our case, detailed semantic descriptions are being created, for instance, when translating the Kalevala story into RDF and OWL. Here each Kalevala part of potential interest to the end-user is annotated in terms of events, thematic roles and other metadata. Furthermore, the events may constitute larger entities and have some additional semantic relations with each other. In CULTURESAMPO I this idea was experimented by representing processes and events of two Kalevala poems, in paintings, photos, and cultural processes (Junnila et al., 2006).

In CULTURESAMPO II this work continues with a new modified event-based model. Furthermore, in the new scheme, annotations can be given at three levels of granularity in order to enable vertical interoperability:

Keywords In some cases only keywords are available as subject metadata. At this level the annotation is a set of literal values. Even if ontological annotations have been used (cf. below), literal keywords may be needed for free indexing words.

Keyconcepts Here the annotation is a set of URIs or other unique references to an ontology or a classification system, such as Iconclass. The additional knowledge introduced by keyconcepts w.r.t. using literal keywords is their ontological connections. This enables semantic interoperability, as discussed earlier.

Thematic roles At this level thematic roles between activities and other entities can be specified. The additional knowledge w.r.t. using only keyconcepts is the distinction of the roles in which the keyconcepts are at different metadata descriptions.

Each new level of annotation granularity only adds new information with respect to the previous level. This means that semantically richer representations can be easily interpreted at the lower level. Event-based descriptions mean at the keyconcept level that only the entity resources that are used in the events are considered, not the properties. At the keyword level, only the literal labels of the annotations at the keyconcept level are considered. This strategy enables, e.g., application and integration of traditional text-based search methods with ontological annotations—a useful feature since much of the content in semantic portals is in textual form in any case (e.g., free text descriptions of collection items, biographical articles, poems etc.).

The main ontology underlying CULTURESAMPO II is the General Finnish Upper Ontology YSO (Hyvönen et al., 2005b) of about 20,000 concepts. This lightweight ontology has been created based on the widely used General Finnish Thesaurus YSA[19]. CULTURESAMPO also makes use of extended versions of the ontologies used in MUSEUMFINLAND.

6 The Portal

CULTURESAMPO II provides the end-user with semantic search and browsing facilities in a way similar to MUSEUMFINLAND. Semantic multi-facet search can be used. Since the ontological model is event-centric, the user is provided with a view classifying verb-like event concepts in addition to more traditional views (persons, collections, etc.). Figure 5 illustrates the search interface.

When a search object is selected to viewing, recommended semantic links with explanations are provided for browsing. Also here the event-centric model is evident: most recommendations are based on sharing events and roles. Figure 6 shows a search object page of a photograph for illustration.

In addition, CULTURESAMPO II includes many new forms of semantic visualization, especially w.r.t. geographical information and time lines (Kauppinen et al., 2006). For visualizing search results on the map, Google Maps[20] service is used (cf. figure 7). It will be used as a search interface, too, later on. In the same vein, the Simile Time Line[21] has been incorporated in the user interface using Ajax-technology (cf. figure 8.

CultureSampo I was implemented on our old OntoViews architecture, based on Apache Cocoon[22]. However, when adding many more cross-linked components to the system in CULTURESAMPO II, such as the timeline, map views, and the new recommendation system, severe limits in the old architecture became apparent.

A major guideline in our work has been to create applications that can be configured to work with a wide variety of RDF data. To accomplish this, we have endeavored to build our applications out of modular components that combine to provide advanced functionality. As CULTURESAMPO II became more complex and started to incorporate components from other projects, there appeared a need for making the individual components smaller and supporting a more complex multidirectional control and configuration flow between them. Apache Cocoon, however, is based on a generally sequential pipeline architecture, which is very limited in its ability to support any multidirectional communication. And while it was possible to make use of modular component libraries on the Java level, there was no architectural support for keeping these components either universal or configurable, which in general resulted in them not being such.

To solve these problems, a new architecture was developed for CultureSampo II based on the well-known Service Oriented Architecture, Inversion of Control and Dependency Injection principles. Specifically, the new platform was built on top of the Apache HiveMind[23] services and configuration microkernel.

7 Discussion

Our work on CULTURESAMPO suggests that using event-based annotations is a promising approach to creating cross-domain semantic portals for several reasons:

1. By using more accurate semantic descriptions semantic search and browsing (recommendation) can be made more accurate and explained in more detail. The semantic accuracy of annotations can be extended in a natural way by the new layer of relational event annotations that explicate the thematic roles between activities and other entities in the description. First tests on CULTURESAMPO I and II seem to indicate that this kind semantic knowledge is vital for semantic information retrieval tasks (search) and for creating insightful semantic linking of contents

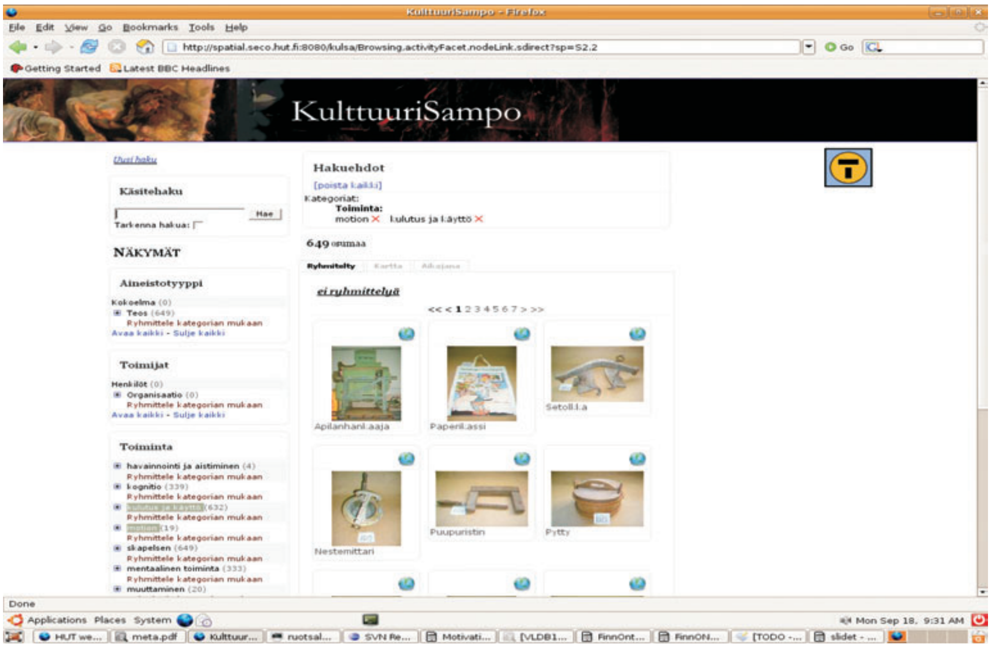
[19] http://www.vesa.lib.helsinki.fi

[20] http://maps.google.com/

[21] http://simile.mit.edu/timeline/

[22] http://cocoon.apache.org/

[23] http://jakarta.apache.org/hivemind/



Uusi haku

Käsitehaku

Hae

Tarkenna hakua: □

NÄKYMÄT

Aineistotyyppi

Kokoelma (0)
⊞ Teos (649)
Ryhmittele kategorian mukaan
Avaa kaikki - Sulje kaikki

Toimijat

Henkilöt (0)
⊞ Organisaatio (0)
Ryhmittele kategorian mukaan
Avaa kaikki - Sulje kaikki

Toiminta

⊞ havainnointi ja aistiminen (4)
Ryhmittele kategorian mukaan
⊞ kognitio (339)
Ryhmittele kategorian mukaan
⊞ kulutus ja käyttö (632)
Ryhmittele kategorian mukaan
⊞ motion (19)
Ryhmittele kategorian mukaan
⊞ skapelsen (649)
Ryhmittele kategorian mukaan
⊞ mentaalinen toiminta (333)
Ryhmittele kategorian mukaan
⊞ muuttaminen (20)

Hakuehdot

[poista kaikki]

Kategoriat:
Toiminta:
motion ✕ kulutus ja käyttö ✕

649 osumaa

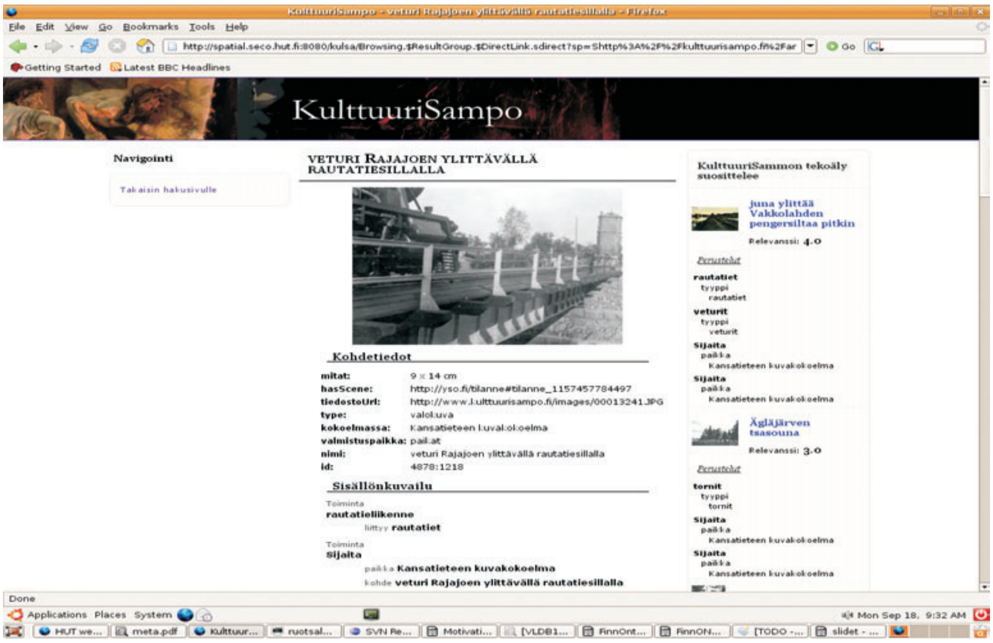
Ryhmitelty Kartta Aikajana

ei ryhmittelyä

<< < 1 2 3 4 5 6 7 > >>

Apilanhankaaja
Paperikassi
Setolka
Nestemittari
Puupuristin
Pytty

Figure 5: CULTURESAMPO II search page. Views are on left and hits on the right.


KulttuuriSampo - veturi Rajajoen ylittävällä rautatiesillalla - Firefox
File Edit View Go Bookmarks Tools Help
http://spatial.seco.hut.fi:8080/kulsa/Browsing.$ResultGroup.$DirectLink.sdirect?sp=Shttp%3A%2F%2Fkulttuurisampo.fi%2Far
Go
Getting Started
Latest BBC Headlines
KulttuuriSampo
Done
Applications Places System
Mon Sep 18, 9:32 AM
HUT we...
meta.pdf
Kulttuur...
ruotsal...
SVN Re...
Motivati...
[VLDB1...
FinnOnt...
FinnON...
[TODO -...
slidet - ...

Navigointi

Takaisin hakusivulle

VETURI RAJAJOEN YLITTÄVÄLLÄ RAUTATIESILLALLA

Kohdetiedot

mitat:	9 x 14 cm
hasScene:	http://yso.fi/tilanne#tilanne_1157457784497
tiedostoUrl:	http://www.kulttuurisampo.fi/images/00013241.JPG
type:	valokuva
kokoelmassa:	Kansatieteen kuvakokoelma
valmistuspaikka:	paikat
nimi:	veturi Rajajoen ylittävällä rautatiesillalla
id:	4878:1218

Sisällönkuvailu

Toiminta
rautatieliikenne
liittyy rautatiet

Toiminta
Sijaita
paikka Kansatieteen kuvakokoelma
kohde veturi Rajajoen ylittävällä rautatiesillalla

KulttuuriSammon tekoäly suosittelee

juna ylittää Vakkolahden pengersiltaa pitkin

Relevanssi: 4.0

Perustelut

rautatiet
tyyppi
rautatiet
veturit
tyyppi
veturit
Sijaita
paikka
Kansatieteen kuvakokoelma
Sijaita
paikka
Kansatieteen kuvakokoelma

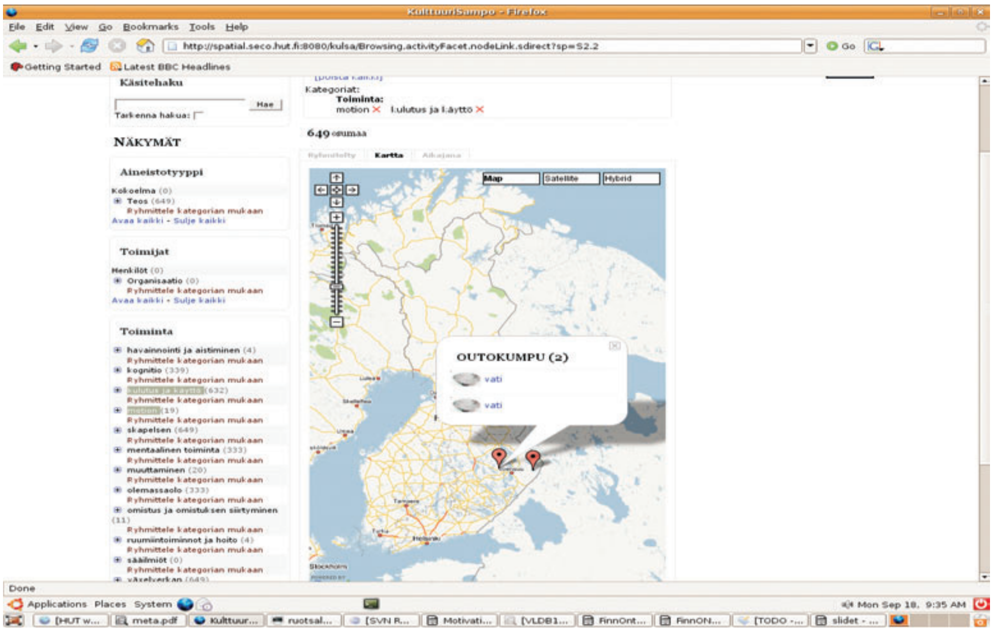
Äglajärven tsasouna

Relevanssi: 3.0

Perustelut

tornit
tyyppi
tornit
Sijaita
paikka
Kansatieteen kuvakokoelma
Sijaita
paikka
Kansatieteen kuvakokoelma

Figure 6: CULTURESAMPO II item page. Metadata is on the left and recommendation links on the right.

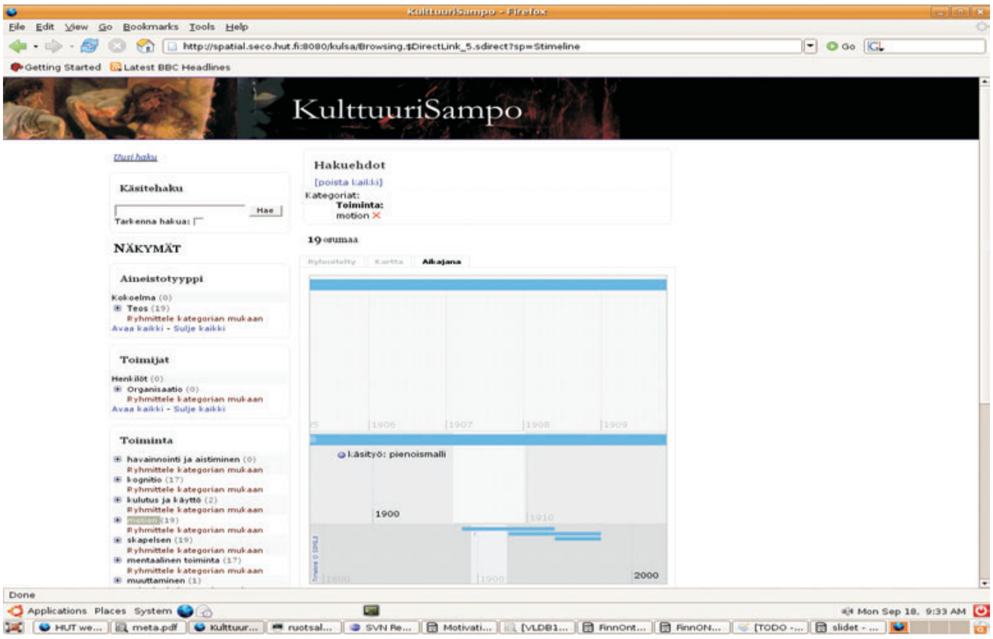

http://spatial.seco.hut.fi:8080/kulsa/Browsing.activityFacet.nodeLink.sdirect?sp=S2.2

Kategoriat:
Toiminta:
motion ✕ kulutus ja käyttö ✕

649 osumaa

KulttuuriSampo - Firefox
File Edit View Go Bookmarks Tools Help
Go
Getting Started Latest BBC Headlines
[poista kaikki]
Käsitehaku
Hae
Tarkenna hakua:
NÄKYMÄT
Aineistotyyppi
Kokoelma (0)
Teos (649)
Ryhmittele kategorian mukaan
Avaa kaikki - Sulje kaikki
Toimijat
Henkilöt (0)
Organisaatio (0)
Ryhmittele kategorian mukaan
Avaa kaikki - Sulje kaikki
Toiminta
havainnointi ja aistiminen (4)
Ryhmittele kategorian mukaan
kognitio (339)
Ryhmittele kategorian mukaan
kulutus ja käyttö (632)
Ryhmittele kategorian mukaan
motion (19)
Ryhmittele kategorian mukaan
skapelsen (649)
Ryhmittele kategorian mukaan
mentaalinen toiminta (333)
Ryhmittele kategorian mukaan
muuttaminen (20)
Ryhmittele kategorian mukaan
olemassaolo (333)
Ryhmittele kategorian mukaan
omistus ja omistuksen siirtyminen (11)
Ryhmittele kategorian mukaan
ruumiintoiminnot ja hoito (4)
Ryhmittele kategorian mukaan
säilmiöt (0)
Ryhmittele kategorian mukaan
växelverkan (649)
Ryhmitelty
Kartta
Aikajana
Map
Satellite
Hybrid
OUTOKUMPU (2)
vati
vati
Stockholm
Done
Applications Places System
Mon Sep 18, 9:35 AM
[HUT w... meta.pdf Kulttuur... ruotsal... [SVN R... Motivati... [VLDB1... FinnOnt... FinnON... [TODO -... slidet - ...

Figure 7: Using Google Maps in CULTURESAMPO II for visualizing search items on the map. The items are positioned based on a place ontology, and interactive to obtain additional information.


http://spatial.seco.hut.fi:8080/kulsa/Browsing.$DirectLink_5.sdirect?sp=Stimeline

Hakuehdot
[poista kaikki]
Kategoriat:
Toiminta:
motion ✕

19 osumaa

KulttuuriSampo - Firefox
File Edit View Go Bookmarks Tools Help
Go
Getting Started Latest BBC Headlines
KulttuuriSampo
Uusi haku
Käsitehaku
Hae
Tarkenna hakua:
NÄKYMÄT
Aineistotyyppi
Kokoelma (0)
Teos (19)
Ryhmittele kategorian mukaan
Avaa kaikki - Sulje kaikki
Toimijat
Henkilöt (0)
Organisaatio (0)
Ryhmittele kategorian mukaan
Avaa kaikki - Sulje kaikki
Toiminta
havainnointi ja aistiminen (0)
Ryhmittele kategorian mukaan
kognitio (17)
Ryhmittele kategorian mukaan
kulutus ja käyttö (2)
Ryhmittele kategorian mukaan
motion (19)
Ryhmittele kategorian mukaan
skapelsen (19)
Ryhmittele kategorian mukaan
mentaalinen toiminta (17)
Ryhmittele kategorian mukaan
muuttaminen (1)
Ryhmitelty
Kartta
Aikajana
1906
1907
1908
1909
käsityö: pienoismalli
1900
1910
1800
1900
2000
Timeline © SIMILE
Done
Applications Places System
Mon Sep 18, 9:33 AM
HUT we... meta.pdf Kulttuur... ruotsal... SVN Re... Motivati... [VLDB1... FinnOnt... FinnON... [TODO -... slidet - ...

Figure 8: Using Simile Time Line in CULTURESAMPO II for visualizing search items on the time line, and for selecting them for a closer look.

automatically (Junnila et al., 2006; Ruotsalo and Hyvönen, 2006).

2. Event-based descriptions can be used for representing the meanings in terms of happenings and entities of the real world based on different metadata schemas. This enables semantic interoperability.

3. The resulting knowledge representation scheme is simpler in terms of the number of properties than the original set of metadata schemas. This makes it simpler to implement reasoning rules needed for the intelligent services for the end-user.

The price for more accurate semantics is the extra cost of creating the annotations. In CULTURESAMPO I all content was manually crafted. In CULTURESAMPO II a semi-automatic process has been used. At the schema level, the content has been enriched automatically by a separate, rule-based knowledge transformation module. This system transforms, e.g., the metadata of paintings into painting events. At the level of enriching the subject descriptions of the content, enriching has been mostly manual by adding thematic role relations between the entities used in the original databases. For example, to tell that Kullervo rides a horse and not vice versa in figure 2, a riding event with Kullervo and an instance of horse in the proper thematic roles has to be created. In principle, the machine and ontologies could help the annotator in her work, if it is known that usually humans ride horses.

The work of annotating narratives, such as the Kullervo poem in Kalevala and the process of farming by the slash and burn method in CULTURESAMPO I (Junnila et al., 2006) has been done completely manually. However, we are also investigating how language technology can be applied to creating semi-automatically annotations for textual contents (Vehviläinen et al., 2006). It is clear, that development of tools that could help in creating annotations will be of outmost importance in the future.

In some cases, like when annotating unique important materials such as Kalevala, the price for detailed annotations can be paid, while in many other cases it is unrealistic to assume that such representations will be available. In CULTURESAMPO this problem of dealing with materials annotated at different levels of semantic accuracy is addressed by using three layers of annotations: keywords, keyconcepts and thematic roles.

The success of the CULTURESAMPO will finally be judged by the end-users. Empirical usability tests are needed in order to evaluate the added value of the semantic approach. The first test, based on the current CULTURESAMPO II, has been scheduled for the autumn 2006. The goal of this experiment is to test whether the end-users really find the semantic recommendations generated by the event-based model feasible and helpful.

CULTURESAMPO II is still a research prototype and its current version contains only a few content types and less that 10,000 search objects. For example, in contrast to CULTURESAMPO I, there are no narratives in the system yet, only events. However, new types of content are being included in the scheme and in the system. Another line of development in the system is designing additional conceptual visualization tools. On the reasoning side, spatiotemporal reasoning under uncertainty is being studied (Kauppinen and Hyvönen, 2006) and is being implemented in the system.

We plan to publish CULTURESAMPO on the public web in 2007.

Acknowledgments

Our research is a part of the National Semantic Web Ontology Project in Finland[24] (FinnONTO) 2003-2007 funded mainly by the Finnish Funding Agency for Technology and Innovation (Tekes).

References

M. Doerr. The CIDOC CRM - an ontological approach to semantic interoperability of metadata. *AI Magazine*, 24(3):75–92, 2003.

E. Hyvönen, E. Mäkela, M. Salminen, A. Valo, K. Viljanen, S. Saarela, M. Junnila, and S. Kettula. MuseumFinland – Finnish museums on the semantic web. *Journal of Web Semantics*, 3(2):224–241, 2005a.

E. Hyvönen, A. Valo, V. Komulainen, K. Seppälä, T. Kauppinen, T. Ruotsalo, M. Salminen, and A. Ylisalmi. Finnish national ontologies for the semantic web - towards a content and service infrastructure. In *Proceedings of International Conference on Dublin Core and Metadata Applications (DC 2005)*, Nov 2005b.

M. Junnila. Tietosisältöjen semanttinen yhdistäminen toimintakuvausten avulla (Event-based approach to semantic linking of data content). Master's thesis, University of Helsinki, March 6 2006.

[24] http://www.seco.tkk.fi/projects/finnonto/

M. Junnila, E. Hyvönen, and M. Salminen. Describing and linking cultural semantic content by using situations and actions. In *Semantic Web at Work - Proceedings of the 12th Finnish Artificial Intelligence Conference STeP 2006, Volume 1.*, Nov 2006.

T. Kauppinen, R. Henriksson, J. Väätäinen, C. Deichstetter, and E. Hyvönen. Ontology-based modeling and visualization of cultural spatio-temporal knowledge. In *Semantic Web at Work - Proceedings of the 12th Finnish Artificial Intelligence Conference STeP 2006, Volume 1.*, Nov 2006.

T. Kauppinen and E. Hyvönen. Modeling and reasoning about changes in ontology time series. In R. Kishore, R. Ramesh, and R. Sharman, editors, *Ontologies: A Handbook of Principles, Concepts and Applications in Information Systems*. Springer-Verlag, Dec 2006. In press.

E. Mäkelä, E. Hyvönen, and S. Saarela. Ontogator — a semantic view-based search engine service for web applications. In *Proceedings of the 5th International Semantic Web Conference (ISWC 2006)*, Nov 2006.

E. Mäkelä, E. Hyvönen, S. Saarela, and K. Viljanen. OntoViews – a tool for creating semantic web portals. In *Proceedings of 3rd International Semantic Web Conference (ISWC 2004), Hiroshima, Japan*, November 2004.

T. Ruotsalo and E. Hyvönen. Knowledge-based recommendation based on heterogenous metadata schemas, 2006. Paper under contruction.

M. Salminen. Kuvien ja videoiden semanttinen sisällönkuvailu (Semantic content description of images and videos. Master's thesis, University of Helsinki, May 2006.

A. T. Schreiber, B. Dubbeldam, J. Wielemaker, and B. J. Wielinga. Ontology-based photo annotation. *IEEE Intelligent Systems*, 16:66–74, May/June 2001.

J. Sowa. *Knowledge Representation. Logical, Philosophical, and Computational Foundations*. Brooks/Cole, 2000.

J. van den Berg. Subject retrieval in pictorial information systems. In *Proceedings of the 18th international congress of historical sciences, Montreal, Canada*, pages 21–29, 1995. http://www.iconclass.nl/texts/history05.html.

A. Vehviläinen, E. Hyvönen, and O. Alm. A semiautomatic semantic annotation and authoring tool for a library help desk service. In *Proceedings of the first Semantic Authoring and Annotation Workshop, ISWC-2006, Athens, GA, USA*, November 2006. To be published.

K. Viljanen, T. Känsälä, E. Hyvönen, and E. Mäkelä. Ontodella - a projection and linking service for semantic web applications. In *Proceedings of the 17th International Conference on Database and Expert Systems Applications (DEXA 2006), Krakow, Poland*. IEEE, September 4-8 2006.

G. P. Zarri. Semantic annotations and semantic web using nkrl (narrative knowledge representation language). In *Proceedings of the 5th International Conference on Enterprise Information Systems, Angers, France (ICEIS 2003)*, pages 387–394, 2003.

Ontology-based Modeling and Visualization of Cultural Spatio-temporal Knowledge

Tomi Kauppinen
University of Helsinki and
Helsinki University of Technology (TKK) Media Technology
Semantic Computing Research Group (SeCo)
`http://www.seco.tkk.fi/`
`tomi.kauppinen@tkk.fi`

Riikka Henriksson
Helsinki University of Technology (TKK)
Laboratory of Geoinformation and Positioning Technology
FIN-02151 Espoo, Finland
`riikka.henriksson@tkk.fi`

Jari Väätäinen
Geological Survey of Finland
P.O. Box 96
FIN-02151 Espoo, Finland
`jari.vaatainen@gtk.fi`

Christine Deichstetter
Johannes Kepler Universität Linz and
Helsinki University of Technology (TKK)
`c.deichstetter@gmx.at`

Eero Hyvönen
Helsinki University of Technology (TKK) and
University of Helsinki
Semantic Computing Research Group (SeCo)
`http://www.seco.tkk.fi/`
`eero.hyvonen@tkk.fi`

Abstract

Geographic knowledge is essential in handling a large variety of resources, including cultural contents such as museum artifacts, maps, books, photographs, and videos. The metadata of such resources often need to refer to a geographic entity or region, for example to the place where an artifact was produced, used, or found. In this paper, we examine how geographical knowledge may be represented ontologically to enable different types of searches, visualization, and inference in cultural semantic portals and other semantic geo-applications. In particular, we show how change in time between historical regions can be explicated as an ontology and be used for reasoning. Regarding search and visualization, we show how maps of different time periods can be visualized as transparent overlays on top of Google Maps, how cultural content can be visualized on them, how geo-based queries can be formulated based on maps, and how additional services, such as a meta-search system, can be integrated in the mash-up system. The work presented is being integrated at the practical level in the cultural semantic cross-domain portal CultureSampo.

1 Introduction

A large proportion of cultural resources such as artifacts, collections, books, photographs, and videos are geographically referenced and thus should be identified by search terms that refer to locations (Jones et al., 2001; Stuckenschmidt and Harmelen, 2004). This is because they are produced, found or used in those locations, or they have some other relationship to the location in question. By georeferencing the resources (Schlieder et al., 2001), different spatial queries can be enabled in order to find interesting connections.

The term georeference refers to the act of assigning locations to geographic objects, which are entities representing some part on or near the Earth's surface. The simplest form of georeferencing is place naming. However, the problem of representing this geographically referenced knowledge is complex due to many reasons. Place names are usually unique only within an area or domain of the Earth's surface (e.g. city name is usually unique within the domain of a state) and they may become obsolete through the time. For example, different countries and other regions are merged, split and renamed due to reorganizations at the social and cultural levels thus causing new territorial shapes. Place names may also have different meaning to different people, and within the different contexts they are used. Well-known systems for georeferencing uniquely across domains are postal codes and geographic coordinate systems (the system of latitude and longitude), from which the former is based on a human activity and the latter one on physical features (Longley et al., 2001; Mark et al., 2001).

Ontology-driven information systems (Guarino, 1998) have been proposed to provide means to represent and manage this kind of complex knowledge. The idea behind modern systems is to (Visser, 2004) "geo-enable" the Web and allow for complex spatial information and services accessible and useful with all kinds of applications e.g. from library systems to museum systems.

In information systems, the term ontology should be considered as a synonym to *conceptual model* in a way that it is independent of its philosophical roots (Welty and Guarino, 2001). The idea is that modeled ontologies capture the properties, classes and their mutual relationships (Guarino and Welty, 2002) — i.e. the essential semantics of the Universe of Discourse (UoD). For example, the *spatial overlap* of regions (Visser, 2004; Stuckenschmidt and Harmelen, 2004) is important in a sense that it affects the way ontological annotation of resources should be made.

This knowledge concerning geographic, spatial entities has to be represented in a machine-understandable, reusable and shareable way so that it can be used, for example, in the Semantic Web (Berners-Lee, 1998; Berners-Lee et al., 2001) as ontologies.

In this paper we examine how ontological knowledge can be used in order to visualize different kinds of georeferenced information and to enable different precise searches and meta searches. We are especially interested in how the results could be used at a practical level in a cultural semantic cross-domain portal such as CULTURESAMPO (Hyvönen et al., 2006) and in creating a prototype of a national geo-ontology server.

In essence, we propose that the following spatio-temporal, ontological issues should be handled in a state-of-the-art-system.

1. Ontological representation of spatial entities and their mutual relationships.

2. Inference based on explication of complex knowledge. We will show how knowledge about historical changes in regions can be modeled and utilized in information retrieval.

3. Visualization of semantic web content on maps. We show how cultural content can be visualized on maps by using Google Maps as an external service.

4. Multi-map visualization by using different kinds of maps simultaneously. It is shown how maps of different time periods can be used by using transparent overlays, which gives the end-user a kind magnifying glass to zoom into history.

5. Polygon-based searches. We will illustrate how to express polygons, and map them into ontological location individuals (instances) such as cities, in order to annotate and search semantic content.

6. Combining map visualization with other services. Visualization on the map can be easily combined with other services. As a demonstration of how this can add value, a meta search service implemented on top of the semantic portal MuseumFinland (Hyvönen et al., 2005) is presented.

In the following we will overview how we have addressed these issues by creating ontologies, reasoning engines and building working demonstrations using existing portals.

2 Representation of Ontological Relationships

Spatial structures and their mutual relations are clearly the most essential form of geographic knowledge. The principles by which the geographic domain - and geo-ontologies - are primarily structured from the theoretical viewpoint are topology (the theory of boundaries, contact and separation), mereology (the theory of part/whole) and geometry (Mark et al., 2001). This is because geographic objects are not merely located in space; they are tied intrinsically to space inheriting many of the structural properties from the Earth's surface. Reasoning in terms of *part-of* relationships is powerful and appears to be well-suited for geographic representation and inference. Part-of relationships are used for containment hierarchies and are closely related to representation of different types of administrative hierarchies. However, mereology alone cannot deal with some very basic spatial relations, such as the relations *within*, *contains*, and *overlaps*, and this is where topological reasoning is required (Casati et al., 1998; Mark et al., 2001; Jones et al., 2002).

There has been an active philosophical discussion about the essential spatial relations (e.g. in (Varzi, 1996)) and different attempts to formalize the relations, such as RCC-8 (Randell et al., 1992) and (Egenhofer, 1989). Furthermore, methods to measure spatial relevance, such as topology (of neighboring regions), directions (of related regions), and distances (between regions) combined with partonomical relevance (cf. (Stuckenschmidt and Harmelen, 2004), chapter 8) have been developed.

Semantic web ontologies represent geographic knowledge (among other types of knowledge) in terms of the Resource Description Framework (RDF) (Brickley and Guha, 2004) and the Web Ontology Language (OWL)[1]. RDF and Uniform Resource Identifiers (URI) form the basis of the Semantic Web. OWL and RDF are used for defining concepts and metadata, and URIs are used for identifying resources used in the descriptions uniquely.

As a part of our research, we are developing[2] a Finnish geo-ontology SUO (Suomalainen paikkaontologia) using Semantic Web technologies RDF and OWL. According to the plans SUO-ontology will model different classes of geographical places - either man-made or natural - and their topological and mereological relations, as well as relations defining coordinate-based geometry for points, curves and polygons. Currently SUO includes approximately 150 classes such as *city*, *province*, *graveyard*, *lake* or *river*. These classes will be populated with thousands of instanses from the Place Name Register[3].

3 Explicating Change of Regions

Geographic regions change over time. Regions, for example, are split and merged due to various reorganizations at the social and cultural levels. For example, in 1991 East Germany and West Germany were merged to form Germany, as depicted in figure 1.*West Germany*, *East Germany* and *Germany* are here individuals of an ontology. The x-axis depicts time and the y-axis the relative areas of the countries. The history is full of events like this (Smith and Brogaard, 2003). Budapest was formed via the unification of former towns Buda and Pest, the Czech Republic and Slovak Republic were formed through the separation of Czechoslovakia, and so on. This kind of spatial changes of individuals at the geographical, social and cultural levels are problematic from the information annotation and retrieval viewpoint. Content is annotated by using historical and contemporary concepts and names. However, when querying, concepts and names from other time periods may be used, which leads to low recall and precision.

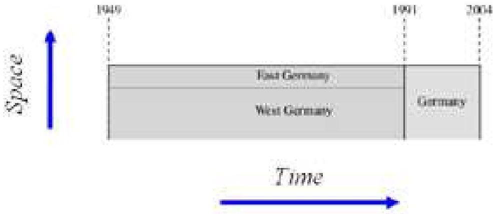


Figure 1: Germany, East Germany and West Germany over time and space.

To address the problem we developed an ontological method (Kauppinen and Hyvönen, 2006, 2005) for representing spatio-temporal changes in regions that essentially define a geo-ontology time series. The figure 2 illustrates the steps of using the method. In the initial data of ontological changes are maintained as a spreadsheet table to ease the editing [4]. Each change, such as a merge or a split of two counties, is represented by a line in the table and is called a "change bridge".

This table is then transformed automatically into RDF-form representing the changes and regions involved. A set of rules is used to construct temporal regions based on the change information. As a simple example, since we know that an administrational region of Viipuri has changed both in 1906 and in 1921, the area of Viipuri has remained the same between (the beginning of) 1906 and (the end of) 1920 and hence a temporal region Viipuri (1906-1920) is created. In a similar manner the rule set is used to construct all the other temporal regions in the ontology time series. At the next phase, the essential properties are inferred for these temporal regions. We declared that the essential property (Guarino and Welty,

[1] http://www.w3.org/2001/sw/

[2] We have used The Protégé Ontology Editor available at (http://protege.stanford.edu/).

[3] Place Name Register was obtained from the National Land Survey of Finland

[4] We have used the freely available OpenOffice Calc (http://www.openoffice.org/) but there are other alternatives available as well.

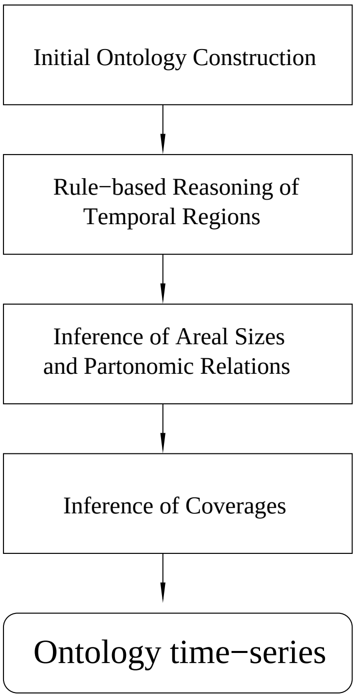

Initial Ontology Construction
Rule−based Reasoning of Temporal Regions
Inference of Areal Sizes and Partonomic Relations
Inference of Coverages
Ontology time−series

Figure 2: Process of creating an ontology time-series.

2002) for each region is the size of the area. For example, since we know that the area of Helsinki has been measured during the year 2005, we can infer that a temporal region Helsinki (1966-NOW) has this areal value as it has remained the same between 1966 and now. Each temporal region in the system has an own identifying URI. This means that if the area of a region changes, there will be a new URI corresponding to a new region with the new value for the size of that area.

An inference engine then reasons the coverages (i.e., how much a region X *overlaps* an other region Y, and vice versa) between all temporal regions of the ontology. The calculation is based on the initial sizes of the regions and their change history. The coverages cannot be calculated directly based the actual polygons of the regions, because these are not known, only the sizes and the changes. In the below, the basic steps of the method are shortly described.

1. Local Bridges. Changes are modeled as individuals of basic change classes, such as split and merged.

2. Local Coverings. The bridges represented in RDF are transformed into a form where the local coverings are made explicit using the sizes of geospatial resources.

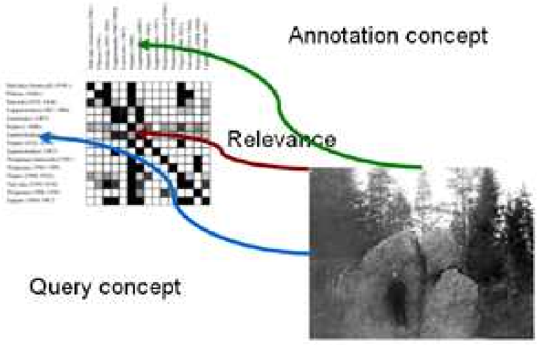


Figure 3: Annotation and indexing concepts matched.

3. Global Coverings. Global overlaps are calculated by chaining local coverings and by considering different change paths between concepts.

4. Visualization of global coverings.

With the method, it has been calculated, e.g., that the temporal region Lappeenranta (1989-) covers 12% of the temporal region Viipuri (-1906). The global coverage information can be used to match annotations and queries, as depicted in figure 3. For example, the photo depicted in the figure is stored in a database of the Geological Survey of Finland GTK (Väätäinen, 2004). According to the attached metadata, the photo is taken within the region Viipuri (-1906). This means that there is a 12% change that the photo is actually within the boarders of the current city of Lappeenranta in Finland. This may be a surprise to many, since Viipuri was annexed to the Soviet Union after the World War II. The explanation is the annexed Viipuri constitutes a temporal region that is different from Viipuri (-1906).

The method is being applied to construct a complete Finnish time-location ontology (Suomen Ajallinen PaikkaOntologia, SAPO) of counties and communes since the mid 1800’s, and to maintain the ontology in the future. It has already been identified (Väätäinen, 2004) that from the beginning of 20th Century there are already over 1100 changes (such as creation, merge, split, and name change) in Finnish communes.

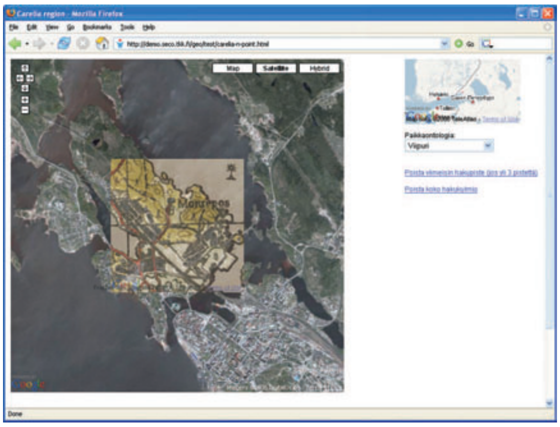

Carelia region - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://demo.seco.tkk.fi/geo/test/carelia-n-point.html
Go
Map
Satellite
Hybrid
Monrepos
Paikkaontologia:
Viipuri
Poista viimeisin hakupiste (jos yli 3 pistettä)
Poista koko hakualue
Done

Figure 4: Using multiple maps simultaneously. A historical Karelian map depicting the park of Monrepos in Viipuri is shown semi-transparently on top of a modern satellite image provided by the Google Maps service.

4 Visualization using Multiple Simultaneous Maps

In order to visualize historical geo-content annotated according to old temporal regions and place names, historical maps are needed. On the other hand, also the current view of places is usually needed at the same time to bridge the conceptual cap between regions of different era. To facilitate this, we created a scheme for using several overlaying maps simultaneously in visualizations. Creation of several layers is a common (de Berg et al., 2000) way to make maps more readable.

The maps and satellite images of the Google Maps service were used as the contemporary view. To provide the historical view, we used a set Finnish maps from the early 20^{th} century covering the area of the annexed Karelia before the World War II. The maps were digitized and provided by the National Land Survey of Finland[5]. In addition, a map of the Espoo region in 1909, provided by the Geological Survey of Finland, was used.

Figure 4 illustrates our scheme. On the left, a satellite Google Maps image of the contemporary Viipuri region is shown. In the middle, a smaller rectangular area is shown with a semi-transparent[6] old Karelian map that is positioned correctly and is of the same scale as the Google Maps image. This smaller view shows the park of Monrepos in Viipuri, a famous Finnish pre-war cultural site that nowadays is a part of Russia. The place cannot be found in current maps as it was, making it difficult for modern users to locate the place geographically. Old maps and names on it could be of substantial benefit when using the visualization in annotating or searching content by maps in systems such as CultureSampo (Hyvönen et al., 2006). The simultaneous views are useful also when comparing geo-information from different eras (e.g., how construction of cities has evolved) or from different thematic perspectives (e.g., viewing a geological map on top of a satellite image).

In order to move around the user is able to use the zooming and navigation functions of Google Maps and the Karelian view is automatically scaled and positioned accordingly. In order to facilitate this, the Karelian maps were processed according to the following steps:

1. Cropping and conceptualization of the map images. The digitized maps had a lot of useful information in addition to the actual maps in the image files. For example, the place name, scale, old coordinates etc. This knowledge was extracted and represented as an ontology. Additional information in the margins of the map images were cropped away.

2. Projection of the images. Projection was done manually by adjusting the images by dragging and stretching them in an image editing program individually using reference points. One could also use a specialized GIS software to do the projection conversion. However, when this was tested, no clear benefits were obtained. For example, relative positional accuracy between the satellite images and the projected map images did not increase remarkably and the process was rather time consuming.

3. Renaming the image files according the Google Maps API specifications.

4. Publishing of the image files on a public web server.

5 Using Points to Explicate a Search Area

Geographic objects are traditionally modeled as points, lines, polygons or fields in spatial databases. Depending on the information a point carries, it can be either an entity point, a label point or an area point.

[5] http://www.maanmittauslaitos.fi /default.asp?site=3

[6] The level of transparency can be altered in the demonstration.

An entity point represents the coordinate position of a point entity such as a tower, a label point carries textual information of the object in question and an area point represents the center point coordinates of a polygon (Zhang and Goodchild, 2002). There has also been discussion (Schlieder et al., 2001) about extending ontologies (gazetteers) with more exact positions like a point or polygon in a given coordinate reference system. For representing areas, polygons would clearly be most satisfying.

However, polygon data is (Schlieder et al., 2001) often 1) proprietary or 2) may be unavailable for e.g. historic regions. Furthermore, 3) detailed polygon data is computationally expensive to process and 4) the exactness of the polygon data should be set to a needed abstraction level. Area points are more widely available. In Finland, names and coordinates are available in the Place Name Register produced by the National Land Survey of Finland.

We have created a method, *n-point search*, for searching this kind of coordinate data. A search query in this method is done by pointing out n points on a map. The user clicks on the map and a polygon is formed, accordingly.

The idea is that the n points define either a simple polygon (without an inner ring) or a complex polygon (with 1...n inner rings) that bounds the search area in question. In other words, the defined polygon can be either complex, concave or convex. If an area point of a certain place is found inside the user-given polygon, it is retrieved and added to the results. Matching places were found using the Jordan Curve Theorem (Haines, 1994).

N-point search would also support the polygon overlays if the regions would be modeled as polygons. This would lead even more precise search results, but in our case such polygon data is not available currently. However, in the future we plan to create polygons and use them as well.

We have also created a special handling for two special cases, namely, for those cases where $n = 1$ or $n = 2$. If $n = 1$ a circle is drawn around the point 1 and the places that are inside the circle are retrieved. An alternative treatment for the $n = 1$ situation would be to find the nearest places. Naturally both of these treatments could be offered for a user. And if $n = 2$, we create a bounding box, where point 1 and point 2 are the opposite corners, e.g. South-West and North-East corners, of the search area.

We have implemented the n-point search as a Google Maps application where Scalable Vector Graphics (SVG) (Ferraiolo et al., 2003) is used for drawing the polygon as an other transparent layer on top of a map. An example of using the n-point search is depicted in figure 5. The n polygon corners are depicted as small crosses. The system has found out three historical communes of the annexed Karelia, Viipuri, Makslahti, and Koivisto, whose center points are situated within the user-specified search polygon.

In the future, we plan to use the idea of user-defined polygons also in ontology population and in annotating contents.

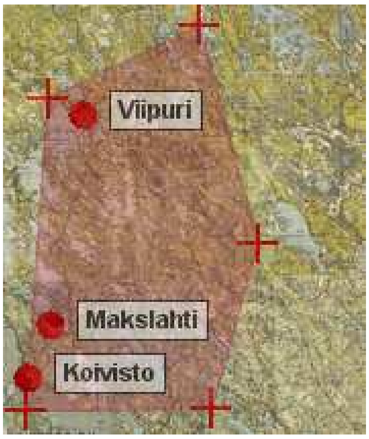


Figure 5: Search results using the n-point search: Viipuri, Koivisto and Makslahti are matched.

6 Combining Visualization with Other Services

Visualizations on the map can be combined interactively with services on the web. In this section it is shown how cultural content can be projected on a map service as interactive elements that can be used to invoke related services, such as search engines.

To test this idea, we have built an application that uses Google Maps API[7] to visualize hundreds of Finnish locations on a zoomable map. User can search information from different portals with location names just by clicking the location on a map.

Figure 6 depicts the idea. On the left pane, a Google Maps view is shown where the red buttons represent Finnish communes and other places of interest. The user has clicked on a button on Helsinki. As a result, the system has queried MuseumFinland (Hyvönen et al., 2005) with the the concept of “Helsinki” as either the place of manufacture or place

[7] http://www.google.com/apis/maps/

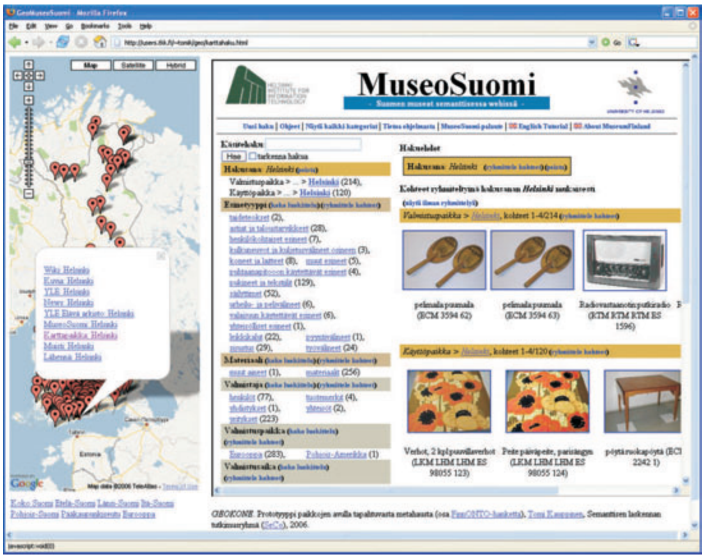

MuseoSuomi
Map
Satellite
Hybrid
Wiki Helsinki
Kuva Helsinki
YLE Helsinki
News Helsinki
YLE Elävä arkisto Helsinki
MuseoSuomi Helsinki
Karttapaikka Helsinki
Muisti Helsinki
Lähemä Helsinki
Hakusana: Helsinki
Valmistuspaikka > ... > Helsinki (214),
Käyttöpaikka > ... > Helsinki (120)
Hakutulokset
Valmistuspaikka > Helsinki, kohteet 1-4/214
Käyttöpaikka > Helsinki, kohteet 1-4/120
Koko Suomi Etelä-Suomi Länsi-Suomi Itä-Suomi
Pohjois-Suomi Pääkaupunkiseutu Eurooppa

Figure 6: A searchable map interlinked with the semantic portal MuseumFinland.

of usage of the artifacts included in the semantic portal. The result is shown on the right pane. In addition, a bubble with several links is opened on the left. The links are search requests to different search engines where the location selected is used as a query parameter.

The search engines engaged in the application include, in addition to the default search engine MuseumFinland, for example Wikipedia[8], Google Images [9], Muisti-project cultural portal[10] and the Old photos service [11] provided by the Geological Survey of Finland.

The search links are created automatically by constructing a search query to the search engines. For example, since the URL's of Wikipedia are simple and permanent, and include the keyword at the end of the URL, the corresponding place name could be used as the keyword. For the capital of Finland, Helsinki, for example, the system guessed that it has a Wiki page *http://en.wikipedia.org/wiki/Helsinki* which happens to be the case. By selecting the link, this page is shown on the right pane instead of the MuseumFinland page. Also an automatic linking to the www-pages of the municipalities in Finland is provided.

However, there are two problems with this approach: First, the automatically constructed link or query may lead to a non-existing page or it might return empty results. For example, some Wikipedia pages may be missing. Nevertheless, this way people can be guided to the corresponding Wikipedia page to contribute their knowledge about places. Second, place names have several meanings that cannot necessarily be disambiguated. For example, “Nokia” is a city in Finland, a telecom company, a person, and an animal in Finnish. Semantic disambiguation could be made, if the services were supporting ontology-based querying based on URIs. For example, when using MuseumFinland this would be possible because the system supports search based on concepts (URIs) in addition to literal keywords.

In the “Geo-MuseumFinland” application of figure 6 places are visualized on the map and used for constructing queries to search engines. We also made a related application for the semantic portal CULTURESAMPO (Hyvönen et al., 2006) where search hits are visualized in the same vein. CULTURESAMPO uses a place ontology that is based on the one used in MuseumFinland and defined in RDF. Language (OWL) (Smith et al., 2004). However, the original place ontology did not have any coordinates. The place ontology was therefore enriched by introducing a *hasCoordinatePoint*-property and by extracting values for these properties from the Place Name Registry obtained from the National Land Survey of Finland. The result is a visualization of the search results on the map. The implementation was done as a Google

[8] http://www.wikipedia.org/
[9] http://images.google.com
[10] http://www.lib.helsinki.fi /memory/etusivue.html
[11] http://www.gtk.fi /palvelut/info/geokuvat/index.php

Maps application where Ajax[12] is used for data interchange of search results between the browser and the server. The figure 7 shows how the search results are visualized as interactive buttons. The user has clicked on a button depicting the hits related to the city of Outokumpu, and a bubble is opened with links to the actual artifacts, in this case two basins ("vati"). By selecting a link, the user will get a more detailed explanation about the item, and semantic recommendations to related materials customary.

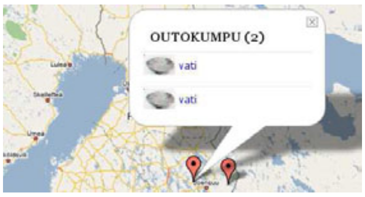


Figure 7: Visualization of search results in semantic portal CULTURESAMPO.

7 Conclusions

In this paper we examined how ontological knowledge can be used in order to represent and visualize different kinds of georeferenced data and to enable searches. We proposed and demonstrated explication of complex spatio-temporal relations between geographical objects, search based on spatio-temporal ontologies, query formulation based on maps, and visualization of historical contents on old and contemporary maps.

The systems overviewed will be used when creating geo-based applications and tools within the the National Semantic Web Ontology Project in Finland[13], especially in CULTURESAMPO and in creating a prototype of a national geo-ontology server.

Acknowledgments

Our research is a part of the National Semantic Web Ontology Project in Finland[14] (FinnONTO) 2003-2007 funded mainly by the Finnish Funding Agency for Technology and Innovation (Tekes).

[12] http://en.wikipedia.org/wiki/AJAX

[13] http://www.seco.tkk.fi/projects/finnonto/

[14] http://www.seco.tkk.fi/projects/finnonto/

References

Tim Berners-Lee. Semantic web road map. Technical report, World Wide Web Consortium, September 1998.

Tim Berners-Lee, Jim Hendler, and Ora Lassila. The semantic web. *Scientific American*, 284(5):34–43, May 2001.

D. Brickley and R. V. Guha. RDF Vocabulary Description Language 1.0: RDF Schema W3C Recommendation 10 February 2004. Recommendation, World Wide Web Consortium, February 2004.

R. Casati, Barry Smith, and A. C. Varzi. Ontological tools for geographic representation. In N. Guarino, editor, *Formal Ontology in Information Systems*, pages 77–85. IOS Press, Amsterdam, 1998.

Mark de Berg, Marc van Kreveld, Mark Overmars, and Otfried Schwarzkopf. *Computational geometry: algorithms and applications*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, second edition, 2000.

M. Egenhofer. A formal definition of binary topological relationships. In *Foundations of Data Organization and Algorithms, 3rd International Conference, FODO 1989*, volume 367 of *Lecture Notes in Computer Science*, pages 457–472, 1989.

Jon Ferraiolo, Dean Jackson, and Jun Fujisawa. Scalable Vector Graphics (SVG) 1.1 specification W3C recommendation. Technical report, World Wide Web Consortium W3C, January 14 2003.

N. Guarino. Formal ontology and information systems. In *Proceedings of the 1st International Conference on Formal Ontologies in Information Systems*. IOS Press, 1998. Trento, Italy.

Nicola Guarino and Christopher A. Welty. Evaluating ontological decisions with ontoclean. *Communications of the ACM*, 45(2):61–65, 2002.

Eric Haines. Point in polygon strategies. In Paul Heckbert, editor, *Graphics Gems IV*, pages 24–46, Academic Press, 1994.

Eero Hyvönen, Eetu Mäkelä, Mirva Salminen, Arttu Valo, Kim Viljanen, Samppa Saarela, Miikka Junnila, and Suvi Kettula. MuseumFinland – finnish museums on the semantic web. *Journal of Web Semantics*, 3(2):25, 2005.

Eero Hyvönen, Tuukka Ruotsalo, Thomas Häggström, Mirva Salminen, Miikka Junnila, Mikko Virkkilä, Mikko Haaramo, Tomi Kauppinen, Eetu Mäkelä, and Kim Viljanen. CultureSampo—Finnish culture on the Semantic Web: The vision and first results. In *Semantic Web at Work — Proceedings of the 12th Finnish Artificial Intelligence Conference STeP 2006*, volume 1, Helsinki, Finland, 2006.

C. Jones, R. Purves, A. Ruas, M. Sanderson, M. Sester, M. van Kreveld, and R. Weibel. Spatial information retrieval and geographical ontologies – an overview of the spirit project, 2002.

Christopher B. Jones, Harith Alani, and Douglas Tudhope. Geographical information retrieval with ontologies of place. In *In Proceedings of the International Conference on Spatial Information Theory*, pages 322–355. Springer-Verlag, 2001.

Tomi Kauppinen and Eero Hyvönen. Modeling coverage between geospatial resources. In *Posters and Demos at the 2nd European Semantic Web Conference ESWC2005*, pages 49–50, Heraklion, Crete, 2005.

Tomi Kauppinen and Eero Hyvönen. *Modeling and Reasoning about Changes in Ontology Time Series*, chapter 11 in book: Ontologies: A Handbook of Principles, Concepts and Applications in Information Systems. Integrated Series in Information Systems, Volume 14. Springer-Verlag, 2006.

Paul A. Longley, Michael F. Goodchild, David J. Maguire, and David W. Rhind. *Geographic Information Systems and Science*. John Wiley & Sons, England, August 2001.

David M. Mark, André Skupin, and Barry Smith. Features, objects, and other things: Ontological distinctions in the geographic domain. *In Spatial Information Theory: Foundations of Geographic Information Science (Lecture Notes in Computer Science 2205)*, pages 488–502, 2001.

D.A. Randell, Z.Cui, and A.G. Cohn. A Spatial Logic based on Regions and Connection. In *Int. Conf. on Knowledge Representation and Reasoning*, Boston, October 1992.

C. Schlieder, T. Vögele, and U. Visser. Qualitative spatial representation for information retrieval by gazetteers. In *Proceedings of Conference of Spatial Information Theory (COSIT)*, volume 2205, pages 336–351, Morrow Bay, CA, 2001.

Barry Smith and Berit Brogaard. Sixteen days. *The Journal of Medicine and Philosophy*, 28:45–78, 2003.

Michael K. Smith, Chris Welty, and Deborah L. McGuinness. W3C OWL Web Ontology Language Guide W3C Recommendation, February 2004.

Heiner Stuckenschmidt and Frank Van Harmelen. *Information Sharing on the Semantic Web*. Springer-Verlag, Berlin Heidelberg, New York, 2004.

A.C. Varzi. Parts, Wholes and Part-whole Relations: the Prospects of Mereotopology. *Data and Knowledge Engineering*, 20:259–286, 1996.

Ubbo Visser. *Intelligent information integration for the Semantic Web*. Springer-Verlag, Berlin Heidelberg, New York, 2004.

Jari Väätäinen. A database containing descriptions of changes of counties in Finland. The Geological Survey of Finland (GSF), Espoo, Finland, 2004.

Christopher A. Welty and Nicola Guarino. Supporting ontological analysis of taxonomic relationships. *Data Knowledge Engineering*, 39(1):51–74, 2001.

J. Zhang and M. Goodchild. *Uncertainty in Geographical Information*. Taylor & Francis, London, 2002.

Improving the Quality of Medication by Semantic Web Technologies

Juha Puustjärvi
Lappeenranta University of Technology
P.O.Box 20 FIN-53851
juha.puustjarvi@lut.fi

Leena Puustjärvi
The Pharmacy of Käpylä
Käpyläntie 8 Helsinki Finland
leena.puustjarvi@kolumbus.fi

Abstract

During the past few years several organizations in the healthcare sector have produced standards and representation forms using XML. For example, patient records, blood analysis and electronic prescriptions are typically represented as XML-documents. This generalization of XML-technologies sets a promising starting point for the interoperability of the various organizations in the healthcare sector. However, the introduction of XML is not enough but many other XML-based technologies have to be introduced in order to achieve a seamless interoperability between the organizations within the healthcare sector. The main goal of this article is to show the gains the interoperability of the health care systems and the deployment of the Semantic Web technologies can provide for electronic prescription systems. In particular, we present an e-prescription ontology and the querying facilities that the deployed ontology provides. We also illustrate how the coordination of the interoperability required by electronic prescription systems can be automated by utilizing XML-based process languages.

1 Introduction

Electronic prescription is the electronic transmission of prescriptions of pharmaceutical products from legally professionally qualified healthcare practitioners to registered pharmacies. The scope of the prescribed products varies from country to country as permitted by government authorities or health insurance carriers. For electronic prescription to be accepted by the physicians, pharmacies and patients it must provide added benefits to all participants.

The problems related to prescribing medication are discussed in many practitioner reports and public national plans, e.g., in (Bobbie, et al., 2005) and (Chadwick and Mundy, 2004). These plans share several similar motivations and reasons for the implementation of electronic prescription systems (EPSs). These include: reduction of medication errors, speeding up the prescription ordering process, better statistical data for research purposes, and financial savings.

A salient trend in medication is that the number of new medications increases every year. As each drug has its unique indications, cross-reactivity, complications and costs also the prescribing medication becomes still more complex every year. However, applying computing technology for prescribing medication this complexity can be alleviated in many ways.

Today there exists a great diversity of competing technological solutions for electronic prescription systems. For example, a citizen may have a memory card, or electronic prescriptions may be transferred via the internet or EDI. There is also diversity in used distribution, e.g., drugs may be transferred to home or they may be picked from pharmacies.

In addition, many prescription writer applications take advantage of internet and other applications such as expert databases, and systems that include information about patients' demographic data and medication history. So, modern prescription writers are internet-based applications that interoperate with many other health care information systems.

During the past few years several organizations in the healthcare sector have produced standards and representation forms using XML. For example, patient records, blood analysis and electronic prescriptions are typically represented as XML-documents (Jung, 2006; Liu et al, 2001; Mattocks, 2005; Stalidis et al 2001; Woolman, 2001). This generalization of XML-technologies sets a promising starting point for the interoperability of the various organizations in the healthcare sector. However, the introduction of XML itself is not enough but also many other XML-based technologies have to be introduced in order to achieve a seamless interoperability between the organizations within the healthcare sector.

In this article we illustrate the interoperability within the healthcare sector from electronic prescriptions point of view. In particular, we illustrate:

- How XML-based technologies can be utilized in modelling the concepts that are related to prescription writing.
- How Web service technology can be used in implementing the interoperability of electronic prescription system and other healthcare systems.
- How the coordination of the interoperability required by electronic prescription systems can be automated by utilizing XML-based process modelling languages. In particular we show how the Business Process Modeling Notation (BPMN) (BPMN, 2006) and Business Process Execution Language for Web Services (BMEL4WS) (BPEL4WS, 2006) can be used for automating the coordination of electronic prescription processes.

The rest of the paper is organized as follows. First in Section 2, we give a motivation by illustrating a paper based prescribing system and electronic prescribing system. Especially we will illustrate the way the physician can utilize the new querying facilities the EPS (Electronic Prescription System). Then, in Section 3, we first give a short introduction to ontologies and then we present a simple e-prescription ontology. We also illustrate how the ontology can be utilized in prescribing medication. Then, in Section 4, we consider the architecture of an electronic prescription system based on service oriented architecture. In Section 5, we illustrate how Business Process Modelling Notation can be used in automating the coordination of electronic prescription processes. Finally, Section 6 concludes the paper by discussing the advantages and disadvantages of our proposed approach.

2 Prescription processes

We now illustrate a scenario of an electronic prescription process that interoperates with other health care systems. The process goes as follows: first a patient visits a physician for diagnosis. In prescribing medication the physician uses a prescription writer. The electronic prescription writer (EPW) used by the physician may interact with many other health care systems in constructing the prescription. For example, the EPW may query previous prescriptions of the patient from the prescription holding store. The EPW may also query patient’s records from other health care systems.

Once the physician has constructed the prescription the EPW may send the prescription to the medical expert system which checks (in the case of multi drug treatment) whether the prescribed drugs have mutual negative effects, and whether they have negative effects with other ongoing medical treatment of the patient. Then the EPW sends the prescription to a medical database system, which checks whether the dose is appropriate.

The medical database may also provide drug-specific patient education in multiple languages. It may include information about proper usage of the drug, warnings and precautions, and it can be printed to the patient. Then the EPW sends the prescription to a pricing system, which checks whether some of the drugs can be changed to a cheaper drug. (This activity is called generic substitution and it aims to promote cost effective drug therapies and to clarify the responsibilities of the prescriber and the pharmacy as well as to enhance both patient autonomy and the efficacy of the pharmaceutical market.)

Once the checks and possible changes have been done the physician signs the prescription electronically. Then the prescription is encrypted and sent to an electronic prescription holding store. Basically the holding store may be centralized or distributed store. The patient will also receive the prescription in the paper form, which includes two barcodes. The first identifies the address of the prescription in the holding store, and the second is the encryption key which allows the pharmacist to decrypt the prescription.

The patient is usually allowed to take the prescription to any pharmacy in the country. At the pharmacy the patient gives the prescription to the pharmacist. The pharmacist will then scan both barcodes by the dispensing application, which then requests the electronic prescription from the electronic prescription holding store. After this the pharmacist will dispense the drugs to the patient and generates an electronic dispensation note. Finally they electronically sign the dispensation note and send it back to the electronic prescription holding store.

3 Deploying ontologies in prescribing medication

The term ontology originates from philosophy where it is used as the name of the study of the nature of existence (Gryber, 1993). In the context of computer science, the commonly used definition is “An ontology is an explicit and formal specification of a conceptualization” (Antoniou and Harmelen, 2004). So it is a general vocabulary of a certain domain. Essentially the used ontology must be shared and consensual terminology as it is used for information sharing and exchange. On the other hand, ontology tries to capture the meaning of a particular subject domain that corresponds to what a human being knows about that domain. It also tries to characterize that meaning in terms of concepts and their relationships.

Ontology is typically represented as classes, properties attributes and values. So they also provide a systematic way to standardize the used metadata items. Metadata items describe certain important characteristics of their target in a compact form. The metadata describing the content of a document (e.g., an electronic prescription) is commonly called semantic metadata. For example, the keywords attached to many scientific articles represent semantic metadata

Each ontology describes a domain of discourse. It consists of a finite set of concepts and the relationship between the concepts. For example, within electronic prescription systems patient, drug, and e-prescription are typical concepts. These concepts and their relationships are graphically presented in Figure 3.

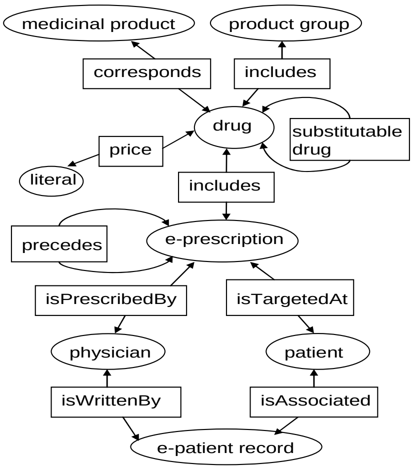


Figure 1. An e-prescription ontology.

In Figure 1, ellipses are classes and boxes are properties. The ontology includes for example the following information:

- E-prescription is prescribed by a physician, and it is targeted to a patient.
- An e-prescription of a patient may precede other e-prescription of the patient, i.e., the e-prescriptions of the same patient are chained.
- Each e-prescription includes a drug
- Each drug has a price, and it may have one or more substitutable drugs.
- Each drug corresponds a medicinal product, e.g., acetylsalicylic acid is a drug and its correspondence medicinal product is Aspirin
- Each drug belongs to a product group, e.g., aspirin belongs to painkillers.
- Each patient record is associated to a patient and it is written by a physician

The information of the ontology of Figure 1 can be utilized in many ways. For example it can be in automating the generic substitution, i.e., in changing meciucinal products cheaper medicinal products within substitutable products. It has turned out that in Finland the average price reduction of all substitutable products has been 10-15 %. In addition the automation of the generic substitution decreases the workload of the pharmacies.

In Figure 2, the ontology of Figure 1 is extended by instances, i.e., it includes the instances of the classes drug, medicinal product, physician, patient and e-prescription. So allows a wide variety of queries including:

- Give me all medicinal products that corresponds the drug asetylsalicylic acid.
- What drugs is included in the medicinal product Aspirin.
- Give me all prescriptions prescribed to Jack Taylor
- Give me all prescriptions prescribed by physician John Smith

- Give me all prescriptions including medicinal product named Panadol.

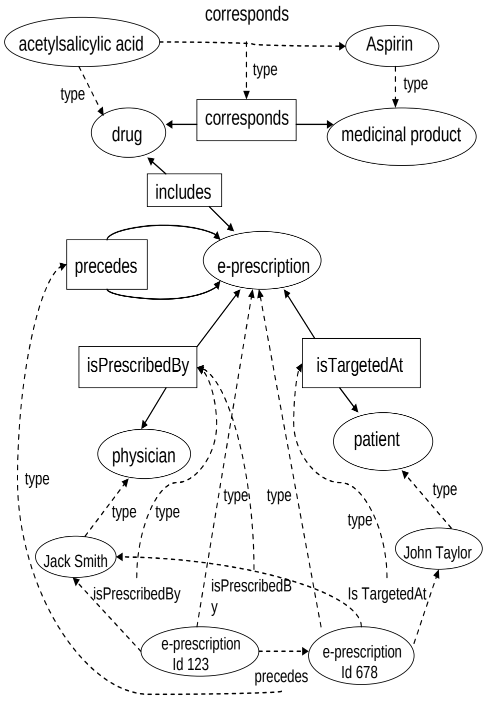


Figure 2. An extension of the e-prescription ontology.

The graphical ontologies of Figure 1 and 2 can be presented by ontology language in a machine processable form. The most commonly used ontology languages are XML (Harold. and Scott Means, 2002), XML Schema XML (Harold. and Scott Means, 2002), RDF (Daconta et al. 2003), RDF Schema (Daconta et al., 2003) and OWL (Singh and Huhns, 2005).

XML (Extensible Markup Language) is a metamarkup language for text documents. It is the syntactic foundation layer of the Se-mantic Web. All other technologies providing features for the Semantic Web will be built on top of XML. Particularly XML defines a generic syntax used to mark up data with simple human readable tags. An important feature of XML is that it does not have a fixed set of tags but it allows user to define tags of their own. For example, various communities have defined their specialized vocabularies (set of tags) for various domains such as MathMl for mathematics, BSML for bioinformatics and GovML (Governmental Markup Language) for government.

4 The architecture of the service oriented EPS

We now describe the architecture that can be used for providing the services described in Section 2 and 3. The architecture is based on the service oriented computing paradigm. Basically, services are a means for building distributed applications more efficiently than with previous software approaches. The main idea behind services is that they are used for multiple purposes. Services are also used by putting them together or composing them. Therefore every aspect of services is designed to help them to be composed.

In the health care sector service oriented computing paradigm provides flexible methods for connecting electronic prescription system to the other relevant health care systems. For example, electronic prescription writer can interact through a Web service with the health care system that supports patient records. There may also be components that are used by different healthcare systems. For example, medical database may provide services for medical information systems as well as for electronic prescription system.

The communication is based on Web services and SOAP-protocol. Originally they provided a way for executing business transactions in the Internet. Technically web services are self-describing modular applications that can be published, located and invoked across the Web. Once a service is deployed, other applications can invoke the deployed service. In general, a web service can be anything from a simple request to complicated business or ehealth processes.

The components of the electronic prescription system are presented in Figure 1

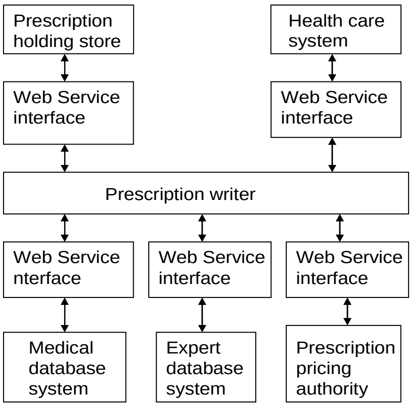


Figure 3. The interaction of a prescription writer.

Each component in the architecture communicates through a Web service interface. Further each message is presented as an XML-document and each XML-document is carried by the SOAP protocol.

5 Using BPMN in modelling e-prescription processes

The Business Process Modeling Notation (BPMN) is a standard for modeling business process flows and web services. The BPMN 1.0 and the UML 2.0 Activity Diagram from the OMG are rather similar in their presentation. However, the Activity diagram has not adequate graphical presentation of parallel and interleaved processes, which are typical in workflow specifications.

We now give an overview of the BPMN. First we describe the types of graphical objects that comprise the notation, and then we show how they work together as part of a Business Process Diagram (BPD). After it we give a simple electronic prescription process description using BPD.

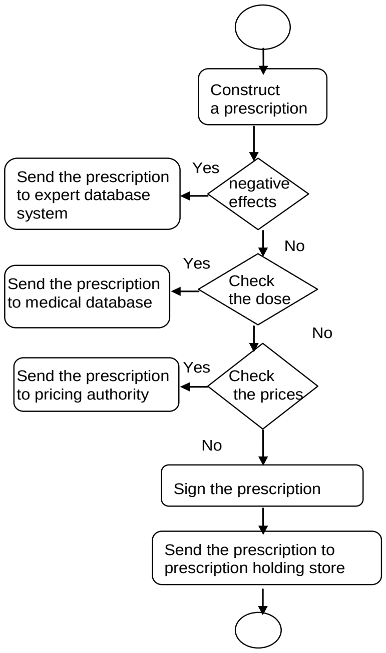
In BPD there are tree Flow Objects: Event, Activity and Gateway:

- An Event is represented by a circle and it represents something that happens during the business process, and usually has a cause or impact.
- An Activity is represented by a rounded corner rectangle and it is a generic term for a work that is performed in companies. The types of tasks are Task and Sub-Process. So, activities can be presented as hierarchical structures.
- A Gateway is represented by a diamond shape, and it is used for controlling the divergence and convergence of sequence flow.

In BPD there are also three kind of connecting objects: Sequence Flow, Message Flow and Association.

- A Sequence Flow is represented by a solid line with a solid arrowhead.
- A Message Flow is represented by a dashed line with an open arrowhead and it is used to show the flow of messages between two separate process participants.
- An Association is represented by a dotted line with a line arrowhead, and it used to associate data and text with flow objects.

In Figure 4 we have presented how the process of producing electronic prescription (described in Section 2) can be represented by a BPD.



Construct a prescription
Yes
negative effects
Send the prescription to expert database system
No
Yes
Check the dose
Send the prescription to medical database
No
Yes
Check the prices
Send the prescription to pricing authority
No
Sign the prescription
Send the prescription to prescription holding store

Figure 4. A BPD-description of the prescription process.

6 Conclusions

In this article we have illustrated the interoperability within the healthcare sector from electronic prescriptions point of view. In particular, we illustrated how XML-based technologies can be utilized in modelling the concepts that are related to prescription writing, and how web-service technology can be used in implementing the interoperability of electronic prescription system and other healthcare systems.

In addition, we have illustrated how the coordination of the interoperability required by electronic prescription systems can be automated by utilizing XML-based languages BPMN and BPEL4WS. The reason for using BPMN is that the BPMN notation is readily understandable for the employees of the health care sector. It is also readily understandable for the business analyst that create the drafts of health care processes as well as for the technical developers responsible for implementing the technology that will perform those processes. Also a notable gain of BPMN specification is that it can be used for generating executable BMEL4WS code.

A consequence of introducing Semantic Web technologies in health care sector is that it significantly changes the daily duties of the employees of the health care sector. Therefore the most challenging aspect will not be the technology but rather changing the mind-set of the employees and the training of the new technology.

References

Antoniou, G. & Harmelen, F., A semantic web primer. The MIT Press, 2004

Bobbie, P., Ramisetty, S., Yussiff, A-L. and Pujari, S. Desgning an Embedded Electronic_prescription Application for Home_Based Telemedicine using OSGi Framework. http://cse.spsu.edu/pbobbie/SharedFile/NewPdfs/eRx-Final2Col.pdf, 2005

BPMN- Business Process Modeling Notation (BPMN), http://www.bpmn.org/ , 2006

BPEL4WS – Business Process Language for Web Sevices. http://www.w.ibm.com/developersworks/ http webservices/library/ws-bpel/

Chadwick, D., and Mundy, D. 2004. The Secure Electronic Transfer of Prescriptions.http://www.healthinformatics.org/hc2004/Chadwick%20Invited%20Paper.pdf, 2004.

Daconta, M., Obrst, L. & Smith, K., The semantic web. Indianapolis: John Wiley & Sons. 2003.

Gruber, Thomas R. Toward principles for the design of ontologies used for knowl-edge sharing. Padua workshop on Formal Ontology., 1993

Harold, E. and Scott Means W., XML in a Nutshell. O'Reilly & Associates, 2002.

Jung, F., XML-based prescription drug database helps pharmacists advise their customers. http://www.softwareag.com(xml/aplications/sanacorp.htm, 2006

Liu, C., Long, A., Li, Y., Tsai, K., and Kuo, H.: Sharing patient care records over the World Wide Web. International journal of Medical Informatics, 61, p. 189-205. 2001.

Mattocks E. 2005. Managing Medical Ontologies using OWL and an e-business Registry / Repository. http://www.idealliance.org/proceedings/xml04/papers/85/MMOOEBRR.html

Singh, M., and Huhns, M., Service Oriented CXDomputing: Semantics Proceses Agents. John Wiley & Sons, 2005.

Stalidis, G., Prenza, A. Vlachos, N., Maglavera S., Koutsouris, D. Medical support system for continuation of care based on XML web technology: International journal of Medical Informatics, 64, p. 385-400, 2001.

Woolman, P. S.: XML for electronic clinical communication in Scotland. International journal of Medical Informatics, 64, p. 379-383. 2001.

RosettaNet and Semantic Web Services

Paavo Kotinurmi
[*]Helsinki University of Technology
Finland
paavo.kotinurmi@tkk.fi

Armin Haller
[†]DERI
National University of Ireland, Galway
armin.haller@deri.org

Abstract

In this paper we present a semantic B2B gateway based on the WSMX semantic Service Oriented Architecture to tackle heterogeneities in RosettaNet messages. We develop a rich RosettaNet ontology and use the axiomatised knowledge and rules to resolve data heterogeneities and to unify unit conversions. We define adaptive executable choreographies, which allow a more flexible integration of suppliers using RosettaNet and make it easy to introduce more competition to the supply chain. The solution is justified with a scenario-based evaluation and by analysing RosettaNet specifications and their implementations.

1 Introduction

B2B integrations offer increased speed, less errors and reduced cost of information exchange between business partners. However, integration efforts still suffer from long implementation times and high-costs [Preist et al. (2005)]. This has lead to business models with simple processes, in which long term rigid partnerships are established [Kotinurmi et al. (2006)]. RosettaNet is a prominent standard for B2B integration that represents an agreement on the message exchange patterns, the message content and a secure transportation mechanism between companies operating in the IT and electronics industries. The message content of structurally valid RosettaNet Partner Interface Processes (PIP) is defined by either DTD for the older PIPs or XML Schema for the recently updated ones. However, the interoperability challenges are only partly solved. DTDs and XML Schemas lack expressive power to capture all necessary constraints and do not make all document semantics explicit. Being able to express constraints in machine-interpretable format is expected by RosettaNet experts as well [Damodaran (2004)].

Companies can use the same PIP messages differently as the messages contain many optional elements. Some companies may support only parts of the enumerated values for packaging-units, unit of measurements and currencies that are in the RosettaNet dictionaries. When the number of partners increases, handling these heterogeneities comes increasingly important and point-to-point syntactic transformations using scripts are not the most lucrative option. This can lead to the situation that buyers use only one supplier as the information systems do not easily support more competitive arrangements. The example scenario that we use in this paper highlights this from the buyer's (requesters) role in a collaboration. Quotes are asked from multiple suppliers (service providers) that use the messages differently. Rules handling the data heterogeneity of their response messages are required to decide on the best supplier for a given order. In the interactions the use of RosettaNet PIPs [1] 3A1 for quotes and 3A4 for purchase orders are handled.

We propose a semantic B2B gateway, which builds upon the Web Service Modelling eXecution environment (WSMX) [Haller et al. (2005)]. This solution applies semantic Web Service (SWS) technologies to RosettaNet collaborations. However, we do not imply the use of SWS technologies to the business partners, as the current knowledge of those technologies is low. In our example scenario only the requester uses SWS technologies, the partners simply use current RosettaNet XML interfaces.

The main contributions of this paper are as follows:

- We encode the information exchanged in RosettaNet PIP3A1 messages in a formal ontology. The ontology includes constraints that cannot be represented in current PIP message schemes, such as dependency between fields.
- We define axioms in the ontology, which constrain the interpretations of certain elements in RosettaNet messages (e.g. units of measurement, packaging size)
- We further add domain specific rules to the knowledge base that are applied at run time to

[1] http://www.rosettanet.org/pipdirectory

resolve data heterogeneities in the RosettaNet messages.

- We define and execute a choreography, which allows to easily integrate new partners to our solution and to handle the data heterogeneities introduced.

The paper is structured as follows: first we give a brief introduction to semantic Web Services in section 2. Section 3 presents the architecture of our semantic B2B gateway and describes its components. Section 4 outlines our use case scenario, its design time modelling tasks and presents evaluation results. In section 5 we position our solution to related work. Finally the paper is concluded in section 6.

2 Semantic Web Services

The Web Service Modeling Ontology (WSMO) [Roman et al. (2005)] provides a conceptual model and a language for semantic markup describing all relevant aspects of general services, which are commonly accessible through a web service interface. However, the semantic mark up provided in WSMO can be grounded to any service (e.g. CORBA, GRID etc.) The ultimate goal of such a markup is to enable the (total or partial) automation of tasks (e.g. discovery, selection, composition, mediation, execution and monitoring) involved in both intra- and inter-enterprise integration. The WSMO conceptual model is formalised by the Web Service Modeling Language (WSML) family of ontology languages, used to describe WSMO elements (goals, ontologies, services, mediators). WSML consists of a number of variants based on different logical formalisms. The different variants of the WSML correspond with different levels of logical expressiveness and are both syntactically and semantically layered. In addition, WSMO defines the underlying model for the Web Service Execution Environment (WSMX). WSMX is an integration platform conforming to the principles of a Service Oriented Architecture (SOA) which facilitates the integration between different systems. The integration process is defined by adaptive operational semantics, defining the interactions of middleware services including discovery, mediation, invocation, choreography, repository services, etc. Thus, WSMO, WSML and WSMX form a coherent framework covering all aspects of semantic Web Services (SWS).

Although WSMX aims to allow a dynamic discovery of partners in the collaboration, our focus lies on how SWS technology tackles the data heterogeneities in e-business collaborations. This has two reasons, first current business practice does not consider an integrated discovery and invocation of services. The “discovery” of business partners is conducted when the infrastructure is set up and are commonly based on well-established and long-running business relations. Second, the technology required for a dynamic discovery of rich semantic descriptions is not mature enough in business settings. The reasoners to perform matchmaking only scale on light semantic descriptions of services with very limited expressivity. Thus we omit the functional description of a service (its capability description which is mainly used during the dynamic discovery) and the formalisation of a requester’s goal. Instead we avail of the WSMO service interface - a description of the communication patterns according to which a service requester consumes the functionality of the service. The WSMO service choreography [Roman and Scicluna (2006)] is based on the formalism of Abstract State Machines (ASMs) [Gurevich (1994)] and allows to model data exchanged in every state of a service execution. It consists of three core elements (1) a *signature*, (2) a *grounding*, and (3) *transition rules*. The signature describes static parts of state descriptions defined in terms of imported ontologies. Each state has a grounding associated defining how concepts in the ontology are linked with messages. Transition rules express changes of states in the choreography by changing the set of ontology instances (adding, removing and updating instances to the signature ontology). An example of such a choreography description is described in section 4.1.

3 Semantic B2B Gateway

In this section we introduce the architectural consideration taken into account for a semantically enhanced RosettaNet B2B integration. We detail the implementation of a generic semantic B2B gateway, being truly agnostic about whether the partners use WSML messages or any ordinary schema language. Our solution is based upon the WSMX platform. WSMX follows a component based design with adaptive operational semantics. Every component provides some service and can be orchestrated within WSMX as required. A semantic RosettaNet B2B Gateway as defined in this paper relies on the following four components:

- The *Adapter Framework* consists of B2B adapters to lift and lower XML instances in the messages sent from the partners to WSML class hierarchies and a middleware adapter connecting to the back-end applications of the requester.

- The *Choreography Engine* executing external service by evaluating a WSMO choreography description. The actual invocation of the service and the choice of the right transport level is performed by the *Communication Manager*.
- The *Resource Manager* constitutes the knowledge base and its interface and provides access to the local component repositories that store definitions of WSMO entities. In our case, these are the domain ontology, the domain specific rules stored in our knowledge base and the choreography specifications stored in the repository.
- The *Reasoner* allows to perform queries on the knowledge base of facts and rules to determine the answer by logical deduction.

WSMX is shipped with these standard components. However, the adapter instances have to be built based on the requirements from RosettaNet interfaces with service providers. The complete architecture of our solution is depicted in figure 1.

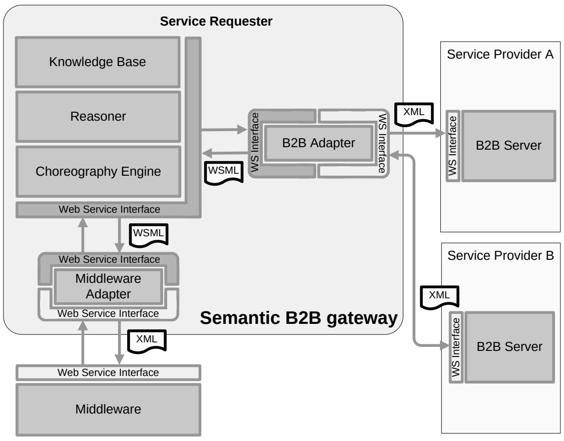


Figure 1: Overview of B2B gateway architecture

3.1 Resource Manager

The resource manager coordinates the access to the repositories of the different components in the architecture and as such gives the illusion of a central repository. This repository acts as a knowledge base and has to be populated with the ontology artifacts used in the different components in our B2B gateway. Most importantly, at least one domain ontology to formally capture the message content exchanged in any of our RosettaNet collaborations is required. Ideally existing *domain ontologies* should be reused. Since an industry wide recognised ontology for business messages has not been established yet, we built the ontology ourselves. It is based on the RosettaNet specification, which itself can be regarded as a lightweight ontology.

However, the task is to not only simply translate the DTDs or XML Schemas to a richer and formal language (i.e. WSML), but to model all the constraints on the semantics of the business documents. We include constraints which are not expressible in DTD or XML Schema and capture implicit knowledge provided in the RosettaNet message guidelines and accompanying documentation to facilitate the promised benefits of automatically resolving data heterogeneities.

An example of the first, a cardinality constraint not expressible in current RosettaNet messages schemas are the constraints imposed on the “BusinessDescription” element used in all RosettaNet PIPs. The “BusinessDescription” element includes business properties that describe a business identity and its location. At least one of the possible three subelements “businessName”, “GlobalBusinessIdentifier” or “PartnerBusinessIdentification” has to be provided in order to be a valid PIP. It is easy to include such constraints in our WSML ontology as shown in listing 1:

```
concept businessDescription
   businessname ofType (0 1) businessName
   globalbusinessidentifier ofType (0 1)
        globalBusinessIdentifier
   partnerbusinessidentification ofType
        partnerBusinessIdentification
   nfp
      dc#relation hasValue validBusinessDescription
   endnfp
axiom validBusinessDescription
   definedBy
      forall ?x ( ?x memberOf businessDescription implies
         ?y memberOf businessName or
         ?y memberOf globalbusinessidentifier or
         ?y memberOf partnerbusinessidentification).
```

Listing 1: Cardinality Constraints spanning multiple elements

Listing 2 shows examples of implicit knowledge we have captured in our RosettaNet ontology. For example, RosettaNet defines a list of 335 possible values for unit of measurements, with the logical relationships between values unspecified. We made these logical relations explicit and included these axiomatisations in our ontology. The first axiom “resolveMeasurementUnitType” in listing 2 shows how the measurement units defined with natural language text in the RosettaNet PIPs can be resolved to its corresponding numerical value. The numerical value can subsequently be used for further calculations (c.f. section 4.1). The second part of the listing defines a function used to relate a kilogram value to its equivalent pound value.

```
axiom resolveMeasurementUnitType
  definedBy
    forall ?x(?x[globalProductUnitOfMeasurementCode
           hasValue ”dozen”] memberOf quoteLineItem implies
           ?x[globalProductUnitOfMeasurementCode hasValue
            ”12”]).
    forall ?y(?y[globalProductUnitOfMeasurementCode
           hasValue ”10−pack”] memberOf quoteLineItem
           implies ?y[globalProductUnitOfMeasurementCode
           hasValue ”10”]).

relation poundKilo (ofType productQuantity, ofType
       productQuantity)
  nfp
    dc#relation hasValue poundKiloDependency
  endnfp

axiom poundKiloDependency
  definedBy
    forall ?x,?y (
      poundKilo(?x,?y) equivalent
        ?x memberOf productQuantity and
        ?x[globalProductUnitOfMeasureCode hasValue
            ”Kilogram”]
          memberOf quoteLineItem and
        ?y memberOf productQuantity and
        ?y[globalProductUnitOfMeasureCode hasValue
            ”Pound”]
          memberOf quoteLineItem and
        ?x = wsml#numericDivide(?y,?x,0.45359237)).
```

Listing 2: Definitional facts in the domain ontology

3.2 Adapter Framework

The adapter framework provides transformation functionality for every non-WSML message sent to the B2B gateway. In our scenario, two adapter instances are required. One for every RosettaNet collaboration partner, who still use XML Schema or DTD based RosettaNet messages and one to connect to the middleware system, providing access to the back-end applications.

The first adapter receives every RosettaNet message and applies the lifting or lowering rules defined in the adapter to map every message instance based on its source schema (in our case XML-Schema) to a WSML instance based on its ontological schema. Listing 3 shows one incoming XML instance and listing 4 its corresponding WSML instance after the transformation rules were applied[2].

It has to be noted that the adapters act for WSMX as the actual service provider. The adapter functionality (i.e. the Web Service endpoint of the adapter) is registered as a service with WSMX and not the RosettaNet service of the partner itself. Thus the adapter is further responsible to execute the correct endpoint of the partner service. However, adapters only perform data manipulation, their interface behaviour replicates the behaviour of the underlying partner service.

The second adapter instance receives every internal message sent from the middleware bus, representing a gateway to every non Web Service compliant back-end application.

3.3 Reasoner

The Reasoner is required to perform query answering operations on the knowledge base, which itself is populated with domain ontologies and domain specific rules at design time and ontology instances at run time. Our reasoner has to handle the WSML-Rule variant and should have built-in predicates for handling basic data-types, basic arithmetic functions as well as basic comparison operators.

Dedicated reasoners for the different WSML variants are still under development. However, there are a number of implementations based on existing reasoners for WSML-Rule available: one is based on the $\mathcal{FLORA}$-2 reasoner, which is also used in the mediation component of WSMX. $\mathcal{FLORA}$-2 is a rule-based knowledge representation formalism and a reasoner for this formalism. It is based on F-Logic [Kifer et al. (1995)], a frame-based logic, WSML is also based upon. Further, there are translations from WSML-Rule to F-Logic available. The reasoner includes a Web Service interface which is used by the service requester to perform queries on the knowledge base. Messages received from the back-end systems of the requester are sent through the middleware. Thus, the middleware adapter has to encode rules how to translate requests from these back end systems to queries on the knowledge base and return the results. This is out of the scope of this paper and will be addressed in future work.

3.4 Choreography Engine

The Choreography Engine as part of WSMX is responsible to control the sending and receiving of messages and update the state of the choreography according to the message content. In contrast to the common use of choreography languages as non-executable descriptions, the engine operates and executes WSMO choreography description such as the one defined in section 4.1.

As mentioned earlier WSMO choreographies are modelled as Abstract State Machines and are processed using standard algorithms during runtime. The state in the machine is represented by ontology

[2] More details on the lifting and lowering of XML Schema to WSML can be found in Kotinurmi et al. (2006).

```
      <QuantitySchedule>
        <ProductQuantity>204</ProductQuantity>
      </QuantitySchedule>
  <GlobalProductUnitOfMeasureCode>dozen
  </GlobalProductUnitOfMeasureCode>
  <SubstituteProductReference>
    <GlobalProductSubstitutionReasonCode>Better product
    </GlobalProductSubstitutionReasonCode>
  </SubstituteProductReference>
  <FinancialAmount>
    <GlobalCurrencyCode>USD
    </GlobalCurrencyCode>
    <MonetaryAmount>198
    </MonetaryAmount>
  </FinancialAmount>
```

Listing 3: XML message instance

```
instance QuoteLineItem1 memberOf rfq#quoteLineItem
  rfq#globalProductUnitOfMeasurementCode hasValue ”dozen”
instance quantitySchedule1 memberOf
    core#quantitySchedule
  core#productQuantity hasValue ”204”
instance substituteProductReference1 memberOf
    core#substituteProductReference
  core#GlobalProductSubstitutionReasonCode
    hasValue ”Better product”
instance totalPrice1 memberOf core#totalPrice
  core#financialAmount hasValue FinancialAmountTot
instance FinancialAmountTot memberOf
    core#FinancialAmount
  core#globalCurrencyCode hasValue USD
  core#monetaryAmount hasValue ”198”
```

Listing 4: and its lifted WSML instance

instances. According to the instance data, a transition rule is selected from the rule base within a choreography. The consequent of the rule is interpreted and the ontology instance is modified accordingly. It is the responsibility of the Choreography Engine to maintain the state of a conversation and to take the correct action when that state is updated. For example, the update of the state of a choreography instance may be the result of a message received from a service provider.

Since WSMO choreography descriptions are expressed in a constraint-based fashion, they are very suitable to be changed and extended once new partners are introduced into a collaboration. It is easy to include transition rules triggering on certain parts of a message sent only by one partner to introduce partner specific data handling (c.f. section 4.1.

4 Evaluation

In our example scenario, we discuss the handling of heterogeneous partners engaged in quoting for a product with a PIP 3A1 Request for Quote (RFQ). We describe the collaboration set up and orchestration through examples from the requester's (buyer's) point-of-view. Then we discuss evaluation results through analysing existing PIP message specifications and example instances and the general benefits from our example scenario.

4.1 Collaboration Set Up

In order to retrieve RFQ messages from different partners at run-time and still being able to process their response messages, the requester has to set up **mapping rules** and define a **choreography description** in the B2B gateway at design time.

Mapping rules developed in the Collaboration Set-Up phase capture any data heterogeneity that is not resolved by the definitional facts in the domain ontology (c.f. section 3.1).

These domain specific rules (conversion relations in our case) define how attribute values in the different WSML instances can be transformed. One such example is given in listing 5. It defines a function to calculate the unit price by taking the “financialAmount” and “productQuantity” given in the RosettaNet ontology instance. This rule can be used by the requester to compare the prices of two or more partners. The “financialAmount” in a PIP3A1 RFQ can refer to different quantities of the product. We made the different packaging sizes and its corresponding value explicit in the ontology as described earlier in section 3.1. Now the requester can query the knowledge base to automatically compare the prices provided as “financialAmount” by the partners transparently on a unit basis.

```
relation unitPrice (ofType financialAmount, ofType
      productQuantity, ofType decimal)
  nfp
    dc#relation hasValue unitPriceDependency
  endnfp

axiom unitPriceDependency
  definedBy
    forall ?x,?y,?z (
      unitPrice(?x,?y,?z) equivalent
        ?x memberOf financialAmount and
        ?y memberOf productQuantity and
        ?z = wsml#numericDivide(?z,?x,?y)).
```

Listing 5: Definition of a conversion relation

Next the choreography description has to be defined. In order to allow a collaboration with its suppliers, the choreography interface of the requester in our scenario has to be compliant with the interface behaviours of the partners providing a RFQ response. Since all suppliers in our Supply Chain use RosettaNet, there is already agreement on the message exchange patterns. However, there are still mismatches on the messages sent and received in the collaboration. Thus, we introduce rules in the choreography (c.f. listing 6) of the requester handling data mismatches in the RosettaNet messages.

It has to be noted that the model defined in listing 6 can actually be regarded as an orchestration in terms of the WSMO conceptual model. Orchestrations in WSMO are modelled in the same way as choreographies. In fact the only difference is on the conceptual level. Whereas WSMO choreographies describe the interface behaviour of one service, orchestrations describe how a service makes use of other WSMO services or goals in order to achieve its capability. In our case, we alter the interface behaviour of the requester by introducing rules, which are for example calling a currency conversion service of an external party. This information in fact would not be part of the choreography description. However, in the WSMX system it is executed in the same engine. Only when the requester wants to publish its interface behaviour to a partner, it is relevant to decide on the abstraction level of what parts of the orchestration he wants to publish.

An extract[3] of such a choreography is shown in listing 6. Please note that the "//..." symbol denotes parts omitted in the listing. The namespace declarations in the first lines of a WSMO choreography definition are also omitted in the listing.

```
31  choreography
32    stateSignature
33      importsOntology {
34        _"http://www.wsmx.org/ontologies/rosetta/coreelements",
35        _"http://www.m3pe.org/ontologies/rosetta/CTRLASM"
36        }
37      out rfq#Pip3A1RFQRequest withGrounding _"http://example
              .org/webServices#wsdl.interface(ServicePortType/
              RFQ/Out)"
38      out curr#currConvRequest withGrounding _"http://www.
              webcontinuum.net/webservices/ccydemo.wsdl#"
39      in rfq#Pip3A1RFQResponse withGrounding _"http://
              example.org/webServices#wsdl.interface(
              ServicePortType/RFQ/In)"
40    transitionRules
41      if (?Pip3A1RequestForQuoteRequest[
42        fromRole hasValue ?fromRole,
43        globalDocumentFunctionCode hasValue
44          ?globalDocumentFunctionCode,
45        quote hasValue ?quote,
46          thisDocumentGenerationDate hasValue
47          ?thisDocumentGenerationDate,
48          thisDocumentIdentifier hasValue ?
                thisDocumentIdentifier,
49          toRole hasValue ?toRole
50      ] memberOf rfq#Pip3A1RFQRequest) and
51      //...
171       update(?controlledState[currentState hasValue 1]
            memberOf ctrlasm#controlledState)
172     endIf
173
174     if (?controlledState[
175       currentState hasValue 1
176       ] memberOf ctrlasm#controlledState) and
177       exists ?Pip3A1RequestForQuoteRespond
178         (?Pip3A1RequestForQuoteRespond memberOf rfq#
              Pip3A1RFQResponse) and
179       //...
357       update(?controlledState[currentState hasValue 2]
            memberOf ctrlasm#controlledState)
358     endIf
359
360     if (?controlledState[
361       currentState hasValue 2
362       ] memberOf ctrlasm#controlledState) and
363       exists ?globalProductSubstitutionReasonCode,
364         ?productIdentification
365         (?substituteProductReference[
366           globalProductSubstitutionReasonCode hasValue
367         ?globalProductSubstitutionReasonCode,
368           productIdentification hasValue ?productIdentification
369           ] memberOf core#substituteProductReference) then
370           add(_# memberOf rfq#Pip3A1RFQRequest)
371         endIf
372
373     if (?controlledState[
374       currentState hasValue 2
375       ] memberOf ctrlasm#controlledState) and
376         exists ?globalCurrencyCode, ?monetaryAmount
377         (?totalPrice[
378           globalCurrencyCode hasValue ?globalCurrencyCode,
379           monetaryAmount hasValue "USD"
380         ] memberOf rfq#totalPrice) then
381             add(_# memberOf curr#currConvRequest)
382         endIf
```

Listing 6: Choreography in WSML

As described in section 2, a WSMO choreography definition consists of a state signature (c.f. line number 32-39 in listing 6), which imports one or possibly many ontologies and transition rules (c.f. line number 40-382), which are basic operations, such as adding, removing and updating on the instance data of the signature ontology. In our example we import two ontologies, the message ontology capturing concepts in RosettaNet messages and a control State ASM ontology, which allows us to define termination and to impose an explicitly modelled control flow for certain parts of the choreography.

The transition rules in listing 6 starting at line 39 capture only a small number of heterogeneities possibly occurring in a RFQ collaboration. New rules can be added when new partners are introduced into the scenario with different data heterogeneities.

The first transition rule (lines 41-172) defines the ontology schema of the message sent by the buyer A (the mode "out" in the "stateSignature" states the passing direction, thus that the message is sent) and that the "currentState" of the machine is updated to the value "1" after a successful transmission of the message. The grounding only defines one WSDL interfaces. However, since the requesters wants to get quote response from multiple providers, the actual grounding list would include more endpoints.

The second rule (line 174-358) checks the control state, to ensure that the "Pip3A1RFQRequest" message was successfully sent and that the

[3] The full listing can be found at: http://www.m3pe.org/ontologies/rosettaNet/

"Pip3A1RequestForQuoteRespond" is received. The constraints in the antecedent of the rule act as a schema validation in the sense that the rule only triggers if the message returned by a partner includes all required fields. For space reasons these parts of the choreography (between lines 179-357) are not shown in listing 6.

The last two transition rules only trigger on a part of the message instance which differs between suppliers. These are rules the requester includes to anticipate possible data heterogeneities. If the requester knows that some partners will provide the amount in USD, the fourth transition rule (lines 373-382) ensures that a currency conversion service is used to calculate the value in Euro. The third transition rule (lines 360-371) fires if a partner provides a substitute product. It results in sending out a new RFQ request.

4.2 Evaluation Results

The evaluation is accomplished by analysing RosettaNet PIPs, real instances obtained from companies and by using a descriptive scenario to demonstrate the utility of our solutions.

We first took a sample 56 PIP message specifications representing 29,5 % of the total 190 messages. The PIPs cover product information, order management, inventory management and manufacturing clusters in RosettaNet classification. Measuring units were used in 24 (43%) and currency information in 28 (50 %) of the PIP messages in the sample. Moreover, we analysed two production message instances and their processing instructions. The PIPs were PIP 3A4 Request Purchase Order and 2A13 Distribute Material Composition Information. The instances use only a part of the whole PIP data model and most of the optional elements are not used. Both messages used unit of measure information, but only one contained currency information. Both supported only kilograms as mass units and Euros as currencies. In both cases, the PIP had just been taken into use with a couple of partners.

The axiomatised message semantics and mapping rules introduced in this paper would have both wide usage across different PIPs and that current real-life XML-based integrations lack this kind of functionality motivates the need for this kinds of solutions. This also provides practical examples of the benefits of SWS technologies to current implementations. Although axiomatised knowledge can also be represented in XSLT scripts or custom codings, the main advantage of expressing these kinds of definitional facts in an ontology is its reusability. The "pound-Kilo" relation captures a conversion that is generally applicable across a wide range of PIP interactions. This reuse is a major advantage over scripts, which are typically rewritten for every schema conversion causing point-to-point integrations that are hard to maintain and test that validation covers the necessary constraints.

The example scenario discussed throughout the paper on quoting and purchasing highlights the requesters need to save on purchasing[4]. Having suppliers from different countries brings heterogeneities as the partners are likely to use different currencies or other measuring or packaging units. Benefits for the buyer result from decreased costs of purchasing as the best value deals can be selected based on best quotes. The suppliers benefit from being able to easier integrate to the buyer without having to make potentially costly changes to their current integration interfaces. In addition, the heterogeneities of using different standards, such as UBL, are easier handled by lifting them to ontologies where mappings can easily be performed.

5 Related Work

There are a number of papers discussing the use of SWS to enhance current B2B standards. Some concentrate on ontologising B2B standards. Foxvog and Bussler (2005) describe how EDI X12 can be presented using WSML, OWL and CycL ontology languages. The paper focuses on the issues encountered when building a general purpose B2B ontology, but does not provide an architecture or implementation. Anicic et al. (2006) present how two XML Schema-based automotive B2B standards are lifted using XSLT to OWL-based ontology. They use a two-phase design and run-time approach similar to our's. The paper is based on different B2B standards and focuses only on the lifting and lowering to the ontology level.

Others apply SWS technologies to B2B integrations. Preist et al. (2005) presented a solution covering all phases of a B2B integration life-cycle.The paper addresses the lifting and lowering of RosettaNet XML messages to ontologies, but no richer knowledge is formalised or used on the ontological level. Trastour et al. (2003b,a) augment RosettaNet PIPs with partner-specific DAML+OIL constraints and use agent technologies to automatically propose modifications if partners use messages differently. Their

[4] See http://www.m3pe.org/ontologies/rosettaNet/ for complete examples of XML instances from which the heterogeneities can be automatically solved.

approach of accepting RosettaNet in its current form and lifting to semantic languages is similar to ours, but we go further by axiomatising implicit knowledge and by providing mappings to resolve data heterogeneities.

6 Conclusion

Our semantic B2B gateway allows a buyer to tackle heterogeneities in RosettaNet interactions. The solution applies axiomatised knowledge and rules to resolve data heterogeneities and to unify various unit conversions using functions to capture definitional facts, such as the relation between pounds and kilograms. Such relations between different enumerations are not specified by RosettaNet. The conversions have potential use in significant portion of the 190 RosettaNet PIP messages as shown in the evaluation of the PIP schemas and the current observed integrations lack the support for such heterogeneity. We further defined adaptive executable choreographies, which allow a more flexible integration of suppliers and make it easy to introduce more competition to the supply chain.

Our future work includes further extending and testing our solution and developing an adapter to the reasoning component, which translates the requests sent from the back-end systems of the requester to queries on the knowledge base and returns the results to the applications.

Acknowledgement

The authors thank Eyal Oren and Tomas Vitvar for fruitful discussions leading to the ideas of this paper. This material is based upon works supported by the Finnish Funding Agency for Technology and Innovation (Tekes) and the Graduate School for Electronic Business and Software Industry and the Science Foundation Ireland under Grant No. SFI/04/BR/CS0694.

References

Nenad Anicic, Nenad Ivezic, and Albert Jones. An Architecture for Semantic Enterprise Application Integration Standards. In *Interoperability of Enterprise Software and Applications*, pages 25–34. Springer, 2006. ISBN 1-84628-151-2.

Suresh Damodaran. B2b integration over the internet with xml: Rosettanet successes and challenges. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 188–195, 2004.

Douglas Foxvog and Christoph Bussler. Ontologizing EDI: First Steps and Initial Experience. In *Int. Workshop on Data Engineering Issues in E-Commerce and Services*, pages 49–58, 2005.

Yuri Gurevich. Evolving algebras 1993: Lipari Guide. In Egon Börger, editor, *Specification and Validation Methods*, pages 9–37. Oxford University Press, 1994.

Armin Haller, Emilia Cimpian, Adrian Mocan, Eyal Oren, and Christoph Bussler. WSMX – A Semantic Service-Oriented Architecture. In *Proc. of the 3rd Int. Conf. on Web Services*, pages 321 – 328. IEEE Computer Society, 2005.

Michael Kifer, Georg Lausen, and James Wu. Logical foundations of object-oriented and frame-based languages. *Journal of the ACM*, 42(4):741–843, 1995.

Paavo Kotinurmi, Tomas Vitvar, Armin Haller, Ray Boran, and Aidan Richardson. Semantic web services enabled b2b integration. In *Int. Workshop on Data Engineering Issues in E-Commerce and Services*, June 2006.

Chris Preist, Javier Esplugas Cuadrado, Steve Battle, Stuart Williams, and Stephan Grimm. Automated Business-to-Business Integration of a Logistics Supply Chain using Semantic Web Services Technology. In *Proc. of 4th Int. Semantic Web Conference*, 2005.

Dumitru Roman and James Scicluna. Ontology-based choreography of wsmo services. Wsmo final draft v0.3, DERI, 2006. `http://www.wsmo.org/TR/d14/v0.3/`.

Dumitru Roman, Uwe Keller, Holger Lausen, Jos de Bruijn, Rubn Lara, Michael Stollberg, Axel Polleres, Cristina Feier, Christoph Bussler, and Dieter Fensel. Web Service Modeling Ontology. *Applied Ontologies*, 1(1):77 – 106, 2005.

David Trastour, Claudio Bartolini, and Chris Preist. Semantic Web support for the business-to-business e-commerce pre-contractual lifecycle. *Computer Networks*, 42(5):661–673, 2003a.

David Trastour, Chris Preist, and Derek Coleman. Using Semantic Web Technology to Enhance Current Business-to-Business Integration Approaches. In *Proc. of the Int. Enterprise Distributed Object Computing Conference*, pages 222–231, 2003b.

CASCOM: Context-Aware Service Coordination in Mobile Computing Environments

Heikki Helin
TeliaSonera Finland Oyj
P.O.Box 970, FIN-00051 Sonera, FINLAND
heikki.j.helin@teliasonera.com

Ahti Syreeni
TeliaSonera Finland Oyj
P.O.Box 970, FIN-00051 Sonera, FINLAND
ahti.syreeni@teliasonera.com

Abstract

The research project CASCOM will implement, validate, and trial value-added support for business services for mobile workers and users across mobile and fixed networks. The vision of the CASCOM approach is that ubiquitous application services are flexibly co-ordinated and pervasively provided to the mobile users by intelligent agents in dynamically changing contexts of open, large-scale, pervasive environments. The essential approach of CASCOM is the innovative combination of intelligent agent technology, semantic Web services, peer-to-peer, and mobile computing for intelligent peer-to-peer (IP2P) service environments. The services are provided by software agents exploiting the co-ordination infrastructure to efficiently operate in highly dynamic environments.

1 Introduction

The essential approach of CASCOM[1] is the innovative combination of agent technology, semantic Web services, peer-to-peer, and mobile computing for intelligent peer-to-peer mobile service environments. The services of CASCOM environment are provided by agents exploiting the CASCOM coordination infrastructure to efficiently operate in highly dynamic environments. The CASCOM intelligent peer-to-peer (IP2P) infrastructure includes efficient communication means, support for context-aware adaptation techniques, as well as dynamic service discovery and composition planning.

CASCOM will implement and trial value-added support for business services for mobile workers and users across mobile and fixed networks. The vision of the CASCOM approach is that ubiquitous application services are flexibly co-ordinated and pervasively provided to the mobile users by agents in dynamically changing contexts of open, pervasive environments.

For end users, the CASCOM system provides seamless access to semantic Web services anytime, anywhere, and using any device. This gives freedom to mobile workers to do their business whenever and wherever needed. For network operators, CASCOM aims towards vision of seamless service experience providing better customer satisfaction. For service providers, CASCOM provides an innovative platform for business application services.

The project will carry out highly innovative research aimed at providing a framework for agent-based data and service co-ordination in IP2P environments. CASCOM will integrate and extend existing technologies in areas such as agent-based mobile computing, service co-ordination, and P2P computing in mobile environments. A generic, open IP2P service environment with its agents and co-ordination mechanisms will be prototypically implemented and deployed in CASCOM mostly as open-source software enabling instant take-up and use within European and world community.

In general, it is expected that the outcomes of CASCOM will have significant impact on the creation of a next-generation global, large-scale intelligent service environment. Both, research results on methods for service provision, discovery, composition and monitoring, and the deployed prototype of an open IP2P service environment in the context of nomadic computing will advance the state of the art of European and world knowledge in areas related to the deployment of services in open systems.

2 Technical Approach

Figure 1 depicts the technologies that we use in the CASCOM project. Software agents will be a key

[1] http://www.ist-cascom.org

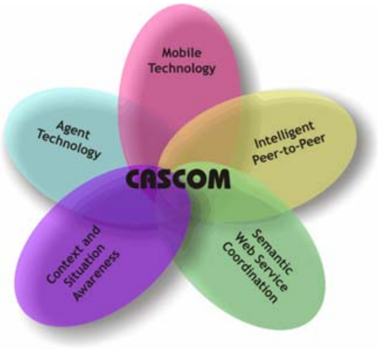

Mobile Technology
Agent Technology
Intelligent Peer-to-Peer
CASCOM
Context and Situation Awareness
Semantic Web Service Coordination

Figure 1: CASCOM Technologies

technology to address the challenges of the CASCOM architecture (see Figure 2). IP2P networks provide an environment for agents to collaborate as peers sharing information, tasks, and responsibilities with each other. Agents help to manage the P2P network complexity, and they will improve the functionality of conventional P2P systems. Our innovations in this domain will concern the development of context-aware agent-based semantic Web services, and flexible resource-efficient co-ordination of such services in the nomadic computing field. Further, context-awareness is investigated in the context of IP2P environment and we will develop context-aware agents which provide various business application services.

Using agents in wireless environments has been studied extensively. We will build on the previous work by using existing agent platforms as a basis of our architecture. However, the P2P aspects are insufficiently taken into account in these platforms and therefore our research represents advancements in this direction. CASCOM will provide solutions for agent communication between agents without assumption of any fixed infrastructure.

Service co-ordination mechanisms of P2P systems can be applied to multi-agent systems to improve their efficiency. Although this may be accepted on a conceptual level, the combination of agents and P2P environments certainly deserves more innovative research, especially regarding nomadic environments. As an example, many P2P overlay network algorithms lacks support for rapid node movements. The dynamic topology of IP2P networks, characteristics of wireless network connections, and the limited capacity or mobile devices pose several challenges that have been addressed inadequately in service discovery architectures. In CASCOM, we will investigate mechanisms for service discovery algorithms for dynamic IP2P environments.

The problem of service co-ordination can be split into several sub problems: discovery, composition planning, execution monitoring, and failure recovery. CASCOM will advance the state of the art by carrying out innovative research on how these problems can be solved in IP2P environments. Especially CASCOM will provide flexible and efficient matching algorithms to be performed in large scale and resource limited IP2P environments.

Using AI planning formalisms in service composition and planning are developed for problems where the number of operators is relatively small but where plans can be complex. In Web service composition for open, large-scale IP2P environments planning methods dealing with huge number of possible service are required. However, plans are not necessarily very complex, and therefore planning methods must follow more closely the structure of the service directories. CASCOM will develop planning mechanisms that establish plan fragments directly on top of the service directory to solve this problem.

Acknowledgements

This work has been supported in part by the European Commission under the project grant FP6-IST-511632-CASCOM. The authors would also like to thank all CASCOM consortium members (DFKI (Germany), TeliaSonera (Sweden/Finland), ADETTI (Portugal), EPFL (Switzerland), URJC (Spain), FrameTech (Italy), UniBAS (Switzerland) and EMA (Finland)).

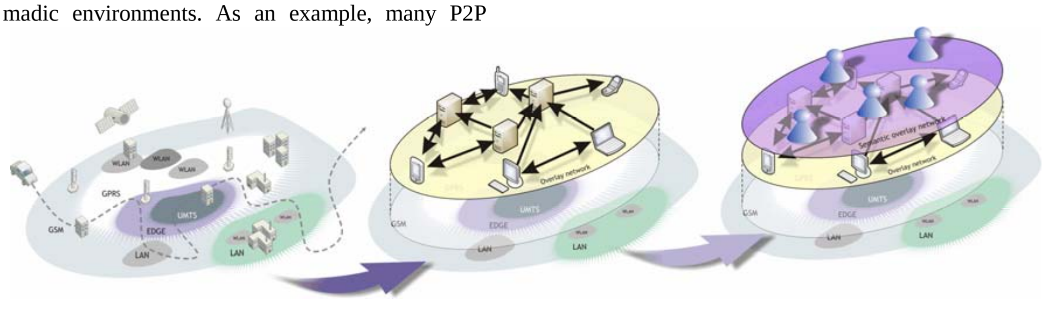


Figure 2: The CASCOM IP2P architecture

Towards reflective information management

Jari Yli-Hietanen[*]
[*]Digital Media Institute
Tampere University of Technology
jari.yli-hietanen@tut.fi

Samuli Niiranen[*†]
[*]Digital Media Institute
Tampere University of Technology
[†]Visiting at Decision Systems Group
Brigham and Women's Hospital
Harvard Medical School
samuli.niiranen@tut.fi

Abstract

We discuss the evolution of information management and argue that the current paradigm is reaching its limits in terms in ability to provide more utility. Our proposition is that this is due to the nature of current information management tools being built around the paradigm of a document, in the abstract sense of the word, sharing many of their limitations. We discuss reflective information management breaking free from this through a native ability to directly reflect on information.

1 Introduction

Proponents of evolutionary psychology, Steven Pinker being perhaps the most famous example, argue that the mental evolution of primates towards the sentient man was pushed forward by the environments faced by early hunter-gatherers (e.g., see (Pinker, 1999)). The result, Homo Sapiens, is the ultimate hunter-gatherer, a social animal who's mental and physical skills are tuned at co-operation and collaboration at the small scale of hunting parties.

Considering the role of information management in this context, humans are adept at processing and storing information natively when it comes to matters close to the hunter-gatherer life style. Natural spoken language as a tool of collaboration, especially when the collaborating parties share a common ground and history, is an evolutionary adaptation providing an extremely efficient information management tool for the hunter-gatherer (see (Pinker, 2000)). The emergence of complexity, as we currently understand the term, in human societies beyond the ad-hoc organizations of the hunter-gatherer coincided with the growing populations and related urban settings facilitated by the surplus of food created by advancing agriculture. The resulting large, complex societies needed new information management tools to handle the large amounts information related to their management. Written language and its practical manifestation, the written document, emerged as the tool to handle the information overflow created by advancing societies.

The use of documents to convey and store the large mass of information created by advancing agricultural, and later on industrial, societies represented a fundamental break from innate information management. Although documents are written in expressive natural language, they fail to reflect the innate linked nature of information management native to the human mind. When we document something, we almost always lose a part of the original piece of information. Typically some part of the original context, of which we are not necessarily even conscious of at the time of the documentation, is lost in the process. With documentation, the worst case is that relevant information is lost into a sea of unrelated data without meaning and context. The key question arising from these considerations is if continuing on the documentation-centric path is the optimal choice in the contemporary world of the digital computer. Also, although highly expressive, the interpretation of natural language compositions is highly context-sensitive and currently relies on our native disambiguation facilities and a shared common ground with the composer.

What comes after digital documentation in information management? Looking at the contemporary developments of information technology, one key trend is the emergence of linkage. For example, a key power of the World Wide Web is that it provides a way to dynamically link documents together. The benefits of linkage in the case of digital documents are obvious, especially when the goal is to search and sort them. Linkage is to documents what roads are to cities. Still, linked documents do not provide a panacea for the complex information management needs sprouting around us in business and elsewhere. This is true even if we expand the concept of a digital document to include rigidly coded and structured information in order to solve the problem of ambiguity inherent to natural language as we will discuss later on. It can argued that information management systems built around the concept of a digital document, even a linked and conventionally structured one, have failed to fulfill the promises made in the early age of computing. Could the role of linkage in information be brought to a new level and could this change bring about advantages for the management of information?

Looking at this question from the perspective of business or other organizations provides a number of interesting insights. Organizations targeting complex activities have traditionally been hierarchical with formalized responsibilities and roles. Still, modern management science has raised the question whether this organizational approach is the optimal one in all situations (Wilkinson, 1998). So what is holding back the trend towards decentralization in organizations targeting complex activities? Complex activities involve many pieces of information which current information management tools, based primarily on electronic documents, cannot reflect in a natural way. The result is that a hierarchy is required to gradually transform management decisions down to practical work orders and, on the other hand, to provide reporting of factory floor activities back to the decision makers.

As a practical example, a single work order document, even if available in the context of an enterprise resource planning system, cannot natively reflect on the activity it relates to. Not all the linkages, relevant to the purpose and meaning of the work order, are available. Native means here the ability to obtain the relation between this piece of information and the goals of the organization. Comparing large organizations to single-person ventures it can be argued, that a single-person venture has a relative advantage in information management. Namely, the entrepreneur can natively understand the entire value chain.

2 Reflective information management

A viewpoint we denote ‘reflective information management’ approaches the presented problem by introducing a way to manage information with many dependencies always relevant to complex activities. How can we briefly characterize the approach? There are two key aspects to it. First of all, information is natively handled as an unrestricted linkage of atomic reflections, collected ideally directly from transactions resulting from financial or other concrete decisions made by people and including other information relevant to the activity at hand. The descriptive power of a diverse and large set of direct observations cannot be stressed enough. We argue that the resulting associative power of the linkage can create a foundation for reducing the need for layered, manual information management. Secondly, looking at the operational side of reflective information management, the emphasis is put on human abstraction level, iterative collaboration between humans and information management tools. Simply put, reflective information management tries to replace the middle men reading and writing documents produced in a layered way with the ability to directly obtain the relation between each piece of information and the goals of the activity in question.

How does reflective information management relate to the numerous attempts at intelligent computing and artificial intelligence? In contrast to attempts at a generic intelligent machine, using approaches such as neural networks or expert systems (Russell and Norvig 2003), reflective information management attempts to construct a mechanism supporting people in their information management activities. The focus is moved from standalone, unsupervised intelligent systems towards incrementally and cumulative built tools, which continuously collaborate with users through the efficient and flexible use of abstraction level traversal. Within this collaboration, the key tool is support for reflective communication among collaborating people instead of support for formal documentation.

As stated, the capability of the approach is to enable seamless traversal within different levels of abstraction and different dimensions of an available information mass. The goal is not to automate decision making but to provide people making decisions with context-optimized access to information. The key point is that access to information is provided at human abstraction level in a way that people can concentrate on decision making instead of formatting and interpreting information to and from machine-readable formats. Even more important goal is the capability to provide the people with the information (e.g., in the form of seemingly simple but highly contextualized messages) currently relevant to their decision making.

Going back to the presented historical review towards the use of documents, in the abstract sense of the word, from natural collaboration, we note that ultimately reflective information management takes the way of organizing human collaboration, also in complex activities, back to its natural form, which is more close to communication among tribesmen than to sterile documentation in hierarchical settings.

3 Unrestricted linkage

Considering the described characteristics of reflective information management, the goal of the approach is human abstraction level information management in settings where the innate human ability is insufficient due to the variety or volume of possibly relevant pieces of information.

Hampton (2003) has discussed how humans use abstraction and context in concept representation. Picking up some key points from his discussion, native human information management involves a flexible abstraction mechanism. Characteristic of this flexibility is that context heavily influences how, for example, a specific heard word is interpreted. Another characteristic he lists is the ability to flexibly add new dimensions to previously learned concepts without hindering the ability to pick the currently most relevant dimensions during a cognitive process. Also, human learning is characterized by the ability to fluidly leverage previously learned concepts and their relations while facing new phenomena.

Considering implications of these characteristics for reflective information management, the key observation is the unbounded nature of human abstraction level information management. This leads into the conclusion that design-while-use is the only possible basic paradigm for reflective information management tools. However, we also limit the scope of the tools by not setting the goal at autonomous or unsupervised operation but rather at the capability of the tools to serve as collaborative agents. The key capability of the tools is thus the ability for abstraction traversal and dimension surfing in an unbounded concept space. This is in contrast to the approach used, for example, in Semantic Web research where concepts are modeled using pre-constructed ontologies.

Expanding on this, conventionally information has had to be rigidly and strictly coded and structured to be machine-processed. This formalization sets a hindrance for information processing as typically the structure for the information has to be designed before processing. This leads at minimum to increased complexity in the case where the operation environment evolves during use. In practice, such information processing systems work satisfactorily only in bounded cases where the boundaries of the case space can be predicted reliably. For these reasons, reflective information management does not rely on structures fixed before use but rather on an evolving, unrestricted linkage of direct observations and other information. Thus reflective information management is suited also for unbounded cases.

We will next describe one possible physical manifestation for the described unrestricted linkage where a set of concept relations with practically unbounded expressive power using natural language vocabulary for concept and relation labels provides the basis for concept modeling. The following is an example of concept relations from health care:

```
12892 "low-density lipoprotein"
(is) "cholesterol"

12893 "LDL" (is) "low-density lipo-
protein"

...

44137  "laboratory  result"  (for)
"patient" 1146045393687

44138 *44137c (is) "290245-1234"

44139 *44137a (is) "LDL value"

44140 *44139c (is) "89 mg/dL"
```

As seen from this example a concept relation has a syntax similar, for example, to the one used in the W3C RDF standards (see http://www.w3.org/RDF/). Specifically, each relation is a <*unique identification, A, B, C, timestamp*> 5-tuple. Basically, the member B relates the member A and C to each other with an optional timestamp indicating the absolute markup time.

The couplings between the concept relations are either implicit or explicit. For example, an implicit coupling exists between all relations containing a "LDL value" labeled concept through the existence of the similar string. Explicit couplings are defined through labeling A, B and C members with a reference notation. This notation uses relation sequence numbers and A, B or C membership as points of reference. For example, when the member C is labeled “*441437c”, it indicates a reference to the C member of the relation with identification 441437. It is important to note that the representation does not make any difference between data and metadata.

One mechanism for operation in an unbounded case space is the implicit coupling between natural language labeled concepts. This enables coupling of a new concept relation to already existing concept relations without the need to modify them or to even verify their existence. It should also be noted that since pre-defined ontologies cannot be used with unbounded case spaces, natural language is the only practical source for concept labeling. The unavoidable ambiguity of natural language has conventionally hindered its use in applications involving machine-processing. We propose that instead of formalizing the used language, ambiguity should be solved by firstly utilizing context as available through the associative power of the linkage and the related descriptive power of the direct observations contained in it and secondarily by making a clarifying query to appropriate user. Naturally, the ability to learn from these interactions is a critical facility here.

4 Case: health care

Health care work illustrates especially well the characteristics of a field involved in complex activities where documents have had an important role in information management. Attempts at fully digitalizing the patient record, where the goal has been to improve health work efficiency, have run into major problems, especially if one considers the original goal of increased efficiency (Dorenfest, 2000), (Heeks, 2005), (Wears and Berg, 2005), (Ash et al, 2004), (Kuhn and Giuse, 2001). The introduction of an electronic patient record has perhaps removed the conventional archivist and document courier from the picture but something has been lost at the same time. It can be argued that the electronic record has increased formalization of health care work, due to the use of strictly structured and inflexible electronic patient documents (Berg and Toussaint, 2003). This increased formalization has broken up natural paths of information sharing and work organization. The value of tacit knowledge, which is by definition impossible to formalize (Nonaka and Takeuchi, 1995), and a shared common ground between collaborating health care workers have an essential role in successful and efficient clinical work (Stefanelli, 2001). Of course, it cannot be denied that early information processing systems in health care, as well as in other industries, have been successful at the automation of actuarial work exemplified by repetitive computation of tabular numerical data. To clarify this, processing of documents can be enhanced by digitalization but not all work can be made more efficient by digitalizing documents.

Expanding on the set of concept relations presented in the section 3, we will next present a simple example on how information management operations can be carried out in this context. Specifically, we look at how presenting a simple goal via natural language fragments can provide for contextual information access to information and semantic sensor and action interfacing.

Let us assume that for a patient group laboratory test results for LDL cholesterol are made available in the described representation through a semantic sensor interface mapping a laboratory system’s structure and semantics for the LDL values directly to concept relations. Now, a medical professional searching for a specific patient’s LDL values initiates a search with the query `"cholesterol 290245"`. Search and traversal algorithms as well as an iterative query process can be applied to provide the professional with the value `"89 mg/dL"`.

Assuming that we describe action interfaces with the same relation mechanism, we can furthermore directly couple the found LDL value and the recipient’s address information, for example, to a physical software component capable of sending an e-mail message again with the help of an iterative query process. This way to relate the pieces of information and the actions operating on them enables, at least in principle, a native ability to directly reflect on information.

5 Discussion

The current discussion represents merely a starting point and a vision for more detailed work in the area of human abstraction level information management. Important open research questions include, but are not limited to, performance issues related to the utilization an unrestricted linkage and the repeatability of operations carried out in the context of the described paradigm.

References

Steven Pinker. *How the Mind Works*. W.W. Norton & Company, New York, 1999.

Steven Pinker. *The Language Instinct: How the Mind Creates Language.* Harper Perennial Modern Classics, New York, 2000.

Adrian Wilkinson. Empowerment: theory and practice. *Personnel Review*. 27(1): 40-56, 1998.

Stuart Russell, Peter Norvig. *Artifical Intelligence: A Modern Approach.* Prentice Hall, 2nd edition, Upper Saddle River, 2003.

James A. Hampton. Abstraction and context in concept representation. Philosophical Transactions of the Royal Society of London B. 358: 1251-1259, 2003.

Sheldon Dorenfest. The decade of the '90s. Poor use of IT investment contributes to the growing healthcare crisis. *Health Care Informatics*.17(8):64–7, 2000.

Richard Heeks. Health information systems: Failure, success and improvisation. *International Journal of Medical Informatics*. Aug 19, 2005.

Robert L. Wears, Marc Berg. Computer technology and clinical work: still waiting for Godot. *Journal of American Medical Association*. 293(10): 1261-3, 2005.

Joan S. Ash, Marc Berg, Enrico Coiera. Some unintended consequences of information technology in health care: the nature of patient care information system-related errors. *Journal of American Medical Informatics Association*. Mar-Apr;11(2):104-12, 2004.

Klaus A. Kuhn, Dario A. Giuse. From hospital information systems to health information systems. Problems, challenges, perspectives. *Methods of Information Medicine*. 40(4):275-87, 2001.

Marc Berg, Pieter Toussaint. The mantra of modeling and the forgotten powers of paper: A sociotechnical view on the development of process-oriented ICT in health care. *International Journal of Medical Informatics*. Mar;69(2-3):223-34, 2003.

Ikujiro Nonaka, Hirotaka Takeuchi. *The Knowledge-Creating Company. How Japanese Companies Create the Dynamics of Innovation*. Oxford University Press, New York, 1995.

Mario Stefanelli. The socio-organizational age of artificial intelligence in medicine. *Artificial Intelligence in Medicine*. 23(1): 25-47, 2001.

Elastic Systems: Role of Models and Control

Heikki Hyötyniemi*
*Helsinki University of Technology
Control Engineering Laboratory
P.O. Box 5500, FIN-02015 TKK, Finland

Abstract

Neocybernetics, and specially the framework of elastic systems, gives concrete tools for formulating intuitions concerning complex systems. It can be claimed that a cybernetic system constructs a *model of its environment*, and it applies *model-based control* to eliminate variation in its environment. There exist various valuable control intuitions that are available for understanding cybernetic behaviors, including experiences with *adaptive control.* It turns out that even the evolutionary processes making the systems more and more complex are *thermodynamically consistent* when seen in the control perspective.

1 Introduction

It seems that artificial intelligence is returning to its roots: It used to be cybernetics, as proposed by Norbert Wiener that was one of the cornerstones of early AI, and, still, the same ideas hold. The original reason for intelligence is not to do fancy reasoning, but to survive, and to constitute a functional whole in its environment.

Behaviors in cybernetic systems are based on interactions among low-level system components, constituting structures of *feedback.* In another perspective, feedbacks can be seen as *control.* This understanding has been exploited, and, indeed, cybernetics has also been defined as being study of *communication and control in an abstract sense.* The claim here is that the framework of neocybernetics makes it possible to make the abstract intuitions about communication and control concrete and analyzable.

Since the introduction of cybernetics back in 1950's, control theory, or, more generally *system theory* has matured. Whereas there was plenty to explore in traditional control structures during the last decades, it seems that by now the easy intuitions have been exhausted. The limits of the control paradigm are now visible, and it is evident that there are problems.

It needs to be recognized that control theory is not in all respects an appropriate framework to understand natural complex systems: The main counterargument is that control practices are based on extremely reductionistic approaches and centrally operating control structures. The real cybernetic populations — economies, ecologies, etc. — are based on distributed operations, and it is difficult to see how the centralized structures could be relaxed and reorganized.

Getting distributed by definition means loosening the controls, and major shifts in thinking are needed in engineering and AI. Indeed, traditional centralized control is a prototype of Western ways of structuring and mastering the world — but when applied in modeling of complex systems, there is the unpleasant feel of finalism and purpose-orientedness in the models[1].

There is a two-way contribution here: Control theory gives powerful tools to understand complex systems, but, at the same time, neocybernetic models may give fresh perspectives to the control community. Today, one cannot even imagine the approaches that someday perhaps become possible what comes to distributed agents and complex networks.

What is more, even the most fundamental notions of system theory are challenged by cybernetic considerations. Before having some *function,* a complex system intuitively does not deserve to be called a system. Whereas in system theory there are some crucial concepts, like the distinction between the inside of the system and the outside environment, in cybernetic systems such boundaries become blurred, the environment equally reacting to the system. A system could be defined not as being characterized by some formal structures, etc. — a system is a *functionally complete sustainable entity.*

[1] Perhaps because of the monoteistic view of natural order, one searches for the underlying *primum movens* — Jean-Paul Sartre has said that *even the most radical irreligiousness is Christian Atheism*

2 Cybernetics as control

New useful intuitions can be reached when the contents of familiar concepts are “refilled” with fresh semantics. When studying complex systems, control engineering seems to offer just the right connotations.

2.1 Information vs. noise

Ross Ashby coined the *Law of Requisite Variety* in 1952:

> The amount of appropriate selection that can be performed is limited by the amount of information available.

This is a deep observation. The concept of *information* is, however, left somewhat vague here, and the consequences remain obscure. To make it possible to efficiently apply mathematical tools, such basic concepts necessarily have to be defined in an accurate manner. How is information manifested in data?

When looking at the neocybernetic models in (Hyötyniemi, 2006a), one can see how the models see the data: When studied closer, it turns out that the weighting matrix in the pattern matching cost criterion becomes

$$W = \mathrm{E}\{\Delta u \Delta u^T\}. \tag{1}$$

This means that data is weighted by the correlation matrix when evaluating matches among patterns: The neocybernetic system must see information in variation. Traditionally, when doing parameter fitting applying maximum likelihood criteria for Gaussian data, the approach is opposite — variation is interpreted as something top be avoided — and the weighting matrix is the *inverse* of (1).

When applying Shannons information theory, or Kolmogorov / Chaitin (algorithmic) information theory, the definition of information is strictly syntactical. There is no domain area semantics involved, and thus extreme universality is reached. However, some paradoxes remain: What you expect, contains no information, and *noise* has the highest information content. When applying the neocybernetic view of information, semantics (in a narrow, formalized way) is included in manipulations, making the analyses non-universal — but there is universality among all cybernetic systems. This semantics is based not only on correlations, but on *balanced tensions* among variables. What is expected, is the most characteristic to the system; uncorrelated noise has no relevance whatsoever.

The neocybernetic models are fundamentally based on correlation matrices — principal subspace analysis is just a way of formally rewriting and redistributing this correlation information. The correlation matrices contain atoms of information, entries $\mathrm{E}\{\bar{x}_i \bar{u}_j\}$ revealing cumulated (co)variations among variables. Covariances and variances — such measures for information are easily expressed and exploited, and they are also the basis of modern identification and minimum-variance approaches in control engineering.

2.2 Model-based control

It turns out that a cybernetic system is a “mirror” of its environment, optimally capturing the information there is available. This is not merely a metaphor — note that the formulas in the neocybernetic model (see Hyötyniemi (2006a)) can be given very concrete interpretations:

- **Model.** It turns out that the neocybernetic strategy constructs the *best possible* (in the quadratic sense) description of the environment by capturing the information (covariation) in the environmental data in the mathematically optimal principal subspace based latent variables

$$\bar{x} = \left(\mathrm{E}\left\{\bar{x}\bar{x}^T\right\}\right)^{-1} \mathrm{E}\left\{\bar{x}\Delta u^T\right\} \; \Delta u. \tag{2}$$

- **Estimate.** It turns out that the neocybernetic strategy constructs the *best possible* (in the quadratic sense) estimate of the environment state by mapping the lower-dimensional latent variable vector onto environment applying the mathematically optimal least-squares regression fitting

$$\hat{u} = \mathrm{E}\left\{\bar{x}\Delta u^T\right\}^T \left(\mathrm{E}\{\bar{x}\bar{x}^T\}\right)^{-1} \; \bar{x}. \tag{3}$$

- **Control.** It turns out that the neocybernetic strategy integrates modeling and estimation to maximally eliminate variation in the environment:

$$\tilde{u} = u - \hat{u} \tag{4}$$

The issue of modeling Δu rather than u directly is studied in Sec. 3.1 (when q increases, u and Δu approach each other what comes to the n most significant eigenvalues). Note that in the case of “intelligent agents” that are capable of explicitly taking the competition into account, so that explicit feedback is constructed, original u rather than Δu can directly be modeled.

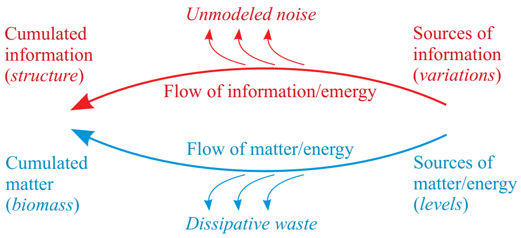

Unmodeled noise
Cumulated information (structure)
Sources of information (variations)
Flow of information/emergy
Flow of matter/energy
Cumulated matter (biomass)
Sources of matter/energy (levels)
Dissipative waste

Figure 1: Abstract flows in a cybernetic system

This all means that a cybernetic system implements *model-based control* of its environment. In terms of information as defined above, this control is the *best possible.* The implemented control is far from trivial: It constitutes a multivariate controller where the n most significant variation directions are equalized (or nullified). It needs to be emphasized that the presented control scheme is just an emergent phenomenon, as there are no centralized control units or "master minds", but everything is based on the local, mindless agents that know nothing about the "big picture". The symmetric structure of the modeling / estimation loop reminds of Heraclitus' words: "The way up and the way down is the same".

2.3 Flows of information and matter

The feedback part in the closed-loop structure above is only an abstraction: It does not correspond to some separate real processes because it only represents the non-ideality of the information transfer. It is interesting to note that for the closed loop structure to emerge, two different kinds of processes need to cooperate — first there is the information flow into the model, and then there is the material flow dictated by the model. Without the other flow the other could not exist either. One could say that a cybernetic system constitutes a *marriage mind and matter,* combining these two incompatible dualistic viewpoints (see Fig. 1).

In the figure, there are the two flows shown separately: On top, there is the flow of information (or emergy), and on bottom, there is the flow of matter (and energy). Most of the flows are wasted — in information flow, the uncorrelated noise becomes filtered, whereas in material flow, it is the dissipative losses that do not get through into the higher-level system. Note that it is often assumed that it is these dissipative flows that are the manifestation of complex system dynamics (Prigogine, 1997) — now these are just a side effect. It is the information in the environment (or *variations* in the data) that dictates the structures within the higher-level system, whereas it is the matter (or actual *levels* in the data) that cumulate as some kind of biomass within this predestinated structure of some kind of *niches.* One could even say that the cybernetic model to some extent captures the Platonian *idea* beyond the changing world.

2.4 Hunger for information

A cybernetic system sees information (emergy) as *resources* available in the environment. In other words, variation is the "nourishment" for systems, and being capable of exploiting these resources is a prerequisite of surviving in an environment. That is why, it seems that evolutionary surviving systems are "hungry" for more and more information. Again, this sounds teleological — but if some system applies this strategy by accident, it immediately has evolutionary benefit in terms of increasing resources. There is no guiding hand needed — but it is like with Gaia: Even though all behaviors can be reduced to lower levels, simplest models are found if stronger emergent-level assumptions are applied.

It turns out that this eternal hunger has resulted in very ingenious-looking solutions for reaching more information, and, to achieve such sophistication, the systems have typically become ever more complicated. First, the variable values visible in the environment can be actively changed by the system: When an organism develops the ability to move, changing ones environment also changes the variable values. Second, there exist an infinity of environmental variables to choose from, and with enough ingenuity, the resource vector can be augmented in different ways, or the environment can be seen in new ways. To illustrate this, see Fig. 2 — there it is shown how the information content of a signal can reside in different frequency regions. The mathematically compact definition of information as being interpreted as variance makes it possible to exploit frequency-domain methods for analysis. Assume that the system concentrates on band-limited signals, so that the signals are filtered as

$$\frac{du_{\mathrm{s}}}{dt} = -\mu_{\mathrm{s}} u_{\mathrm{s}} + \mu_{\mathrm{s}} u_{\mathrm{in}}, \qquad (5)$$

and, similarly, there is an exponential "forgetting horizon" what comes to the covariance estimates:

$$\frac{d\hat{\mathrm{E}}\{\bar{x}_{\mathrm{s}} u_{\mathrm{s}}^T\}}{dt} = -\gamma_{\mathrm{s}} \hat{\mathrm{E}}\{\bar{x}_{\mathrm{s}} u_{\mathrm{s}}^T\} + \gamma_{\mathrm{s}} \bar{x}_{\mathrm{s}} u_{\mathrm{s}}^T. \qquad (6)$$

The parameters μ_s and λ_s are filtering coefficients. Then it turns out that only variation in the darkest area in the figure becomes cumulated in the model (or in the covariance matrix), whereas higher-frequency signals are only filtered by the system. Too high frequences are invisible altogether to the current system, leaving there room for other systems to flourish. As the world gets older, even slower-scale behaviors become statistically modellable — meaning that there is room for ever increasing number of coexisting systems.

The systems are hungry, but they are not greedy. Whereas a system exhausts variation in its environment, there is the same variation inherited in the system itself (remember that PCA model maximally relays variation to latent variables). This gives rise to a *cascade* of trophic layers: Another system can start exploiting the variation that is now visible in the prior system. When the next trophic layer has been established, there is room for a yet higher trophic layer, etc. This kind of succession of systems can be represented as a sequence of “ideal mixers” of information. When new layers are introduced, the ecosystem becomes more and more continuous and smooth – becoming a partial differential equation (parabolic PDE) diffusion model filtering the incoming variation. All loose information seems to give rise to new systems.

Heraclitus said that the underlying principle in nature is *fire* — in the cybernetic perspective, it is this fire that is the driving force, but it seems that the goals of nature could best be explained in terms of a *fire extinguisher.*

There are the physical (chaotic) processes (planets orbiting and rotating, followed by climatological phenomena, etc.) that shuffle the originally “non-informative” flow of solar energy, originally generating the information for other systems to exploit. The input variables on the lowest level are temperatures, nutrients in the soil, diseases, rainfall, etc., and on the level of herbivores, it is then the spectrum of plants to forage on.

The higher-level systems can appear in very different phenospheres, starting from very concrete systems and ending in very abstract memetic ones (see Fig. 3). The interprtetation of signals changes from concrete resources to available/required functionalities, and explicit formulations become impossible. For example, take *politics* — also there exist interesting possibilities for applying the cybernetic thinking.

> Why democracy seems to prosper even though it is less efficient than a dictatorship? Assuming that there is complete information available in the society, democracy represents the *most cybernetic political system*, information on the bottom (needs of citizens) being maximally transferred to the top (decision makers). Parties determine profiles of opinions; party popularity (number of votes x_i) reflects needs u_j in the society, and this is reflected in its possibilities of implementing party visions.

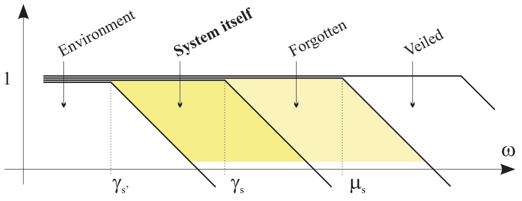


Figure 2: Different systems concentrate on different time scales, seeing the environment in different ways

2.5 Towards optimum?

In cybernetic systems the critical resource is information. In human-made “constructivistic” systems (technical, scientific, ...), the same principle seems to apply, but the variables are more difficult to quantify; the critical resource can be said to be *knowledge*, and one is always “at the edge of understanding”. As soon as there is some new understanding about relationships among variables, it is exploited — this becomes manifested in industrial plants, for example, where new controls are introduced to make the system remain better in balance. These developments are implemented by humans, but, after all, the system follows its own evolution where individual human signal carriers have little to say.

Complex systems seem to develop towards becoming more and more cybernetic. Regardless of the domain, the limiting factor in this evolutionary process seems to be related to extracting and exploiting information (or knowledge). Typical examples are found in working life. The other prerequisite for “cybernetization” — better understanding of the system and gaining more information — is implemented through supervision, questionnaires, and more paper work in general, and the other — applying more efficient controls based on the acquired information — is implemented through increasing administration, organizational changes, missions and visions, and even “developmental discussions”. This is all good, isn’t it?

Unfortunately, the same efficiency pursuit has also come to universities. The role of science is to question and find alternative explanations, avoiding fixed

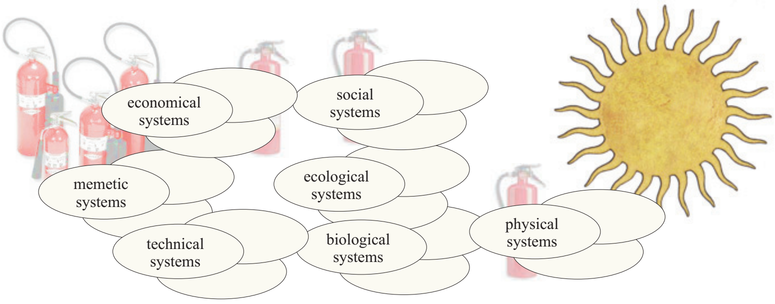


Figure 3: Systems in different phenospheres try to extinguish the Heraclitus' fire

frameworks and controls. This is, of course, far from being efficient, and cybernetization is going on. Whereas the scientific problems being studied are — by definition — missing compact representation, different kinds of variables are needed to make research work better quantifiable. This means that deep problems have to be trivialized, and new measures for "good research" are defined. Earlier the scientific work was evaluated in terms of how well the observed phenomena can be described, but nowadays it is the concrete practical value that is assessed. The traditional ideal in scientific work was self-organization and low hierarchies, but now one is getting back towards hierarchies.

3 Control intuitions

When the control notions are employed, there are many intuitions directly available concerning the behaviors in cybernetic systems. For example, the control system can become *too* good.

3.1 Adaptive control

Adaptation is the key property in truly cybernetic systems, meaning that they are *adaptive control systems,* trying to implement more efficient controls based on simultaneous observations of their environments. If one has control engineering background, one can understand what happens in a truly cybernetic system: Adaptive controllers are notorious in control engineering, as they can behave in pathological ways. The reason for the "explosions" is *loss of excitation.* Good control eliminates variation in data — and after this there is no information where the model tuning can be based on, and gradually the model becomes corrupted. After that, when the model is no more accurate, the variation cannot all be eliminated, and there will exist information in observations once again. The model starts getting better, and after that the control gets better, and the cycle of good and bad closed-loop behavior starts again; the collapses in control performance can be quite catastrophic. This kind of behavior is typical in loops of simultaneous model identification and model-based control. This result is paradoxical: Good balance on the lower level results in high-level instability.

In complex cybernetic systems, the model adaptations can be more complex than in typical adaptive controls. For example, the sampling rate can become fast as compared to the system dynamics (compare to "quartal capitalism" in economy), but increase in sensitivity in any case follows. The evolutionarily surviving systems are on the edge of chaos where variation is no more information but only noise.

Extreme optimization results in "stiffness" of the system, and worsened fitness in changing conditions. It is again easy to see connections — compare to ancient empires: It seems to be so that there is a lifespan for all cultures, after which even the strongest civilization collapses. For example, during Pax Romana, there were no enemies, and the army was gradually ruined – and there was a collapse after a severe disturbance[2]. And this does not only apply to human societies: For some reason, massive extinctions seem to take place in 62 million year cycles (Rohde and Muller, 2005). Do you need some meteors to explain extinctions — or is this simply because of evolution dynamics?

However, as it turns out, the cybernetic strategy where the feedback is implemented implicitly through the environment, results in "gentle" adaptive control, form of *buffering,* where the variation is not

[2] But explicit emphasis on the army results in the Soviet-type collapse: If there is no real need at some time, such investments are cybernetically non-optimal, meaning that the system cannot outperform its competitors in other fields in the evolutionary struggle

fully eliminated, and the closed loop behavior does not become pathological. This is because it is Δu rather than the estimate u itself that is being eliminated from the input data, and some level of excitation remains, making the overall system evolutionarily stable and sustainable.

However, being too ambitious, implementing extreme optimization, and full exploiting the information completely wiping out excitation, is also a possible scenario in a cybernetic system. This happens if the agents are "too smart", implementing the feedbacks explicitly. An agent can not only see the resources but also the competitors and take their actions into account — implementation of such explicit feedback results in combined Hebbian/anti-Hebbian learning (see Hyötyniemi (2006b)).

3.2 Inverting the arrow of entropy

The second law of thermodynamics states that in a closed system entropy always increases (or remains the same if the system is reversible). This means that the the system goes finally towards "heat death", the state of maximum probability, where all variations are eliminated. All physical systems fulfill this principle. However, it can be claimed that cybernetic systems are *thermodynamically inconsistent* as it seems that they operate *against the arrow of entropy:* As new structures emerge, the probability decreases. This difference between "normal" physical systems and "abnormal" cybernetic ones causes an uneasy feeling: Even though there are no outright contradiction here (the cybernetic domain is not closed), different laws apply, and it seems that there must exist different sciences for evolutionary systems.

In the neocybernetic framework, this paradox seems to vanish altogether. Remember that the target of entropy is heat death where everything is in balance — but it was balances that were asumed to be the goal of the neocybernetic systems, too. When the boundaries between the system and its environment are set appropriately, it turns out that *all processes go towards entropy increase,* including the evolutionary ones. There is a minor decrease in the overall entropy when the model of the environment is constructed off-line, once-for-all, but thanks to this model there is huge ever-lasting increase in entropy caused by the on-line variation suppression due to the model-based control. When the environment is seen as residing *inside* the higher-level control system, it turns out that at all levels maximization of entropy takes place. Such observations perhaps offer tools for modeling principles: If it is assumed that all systems go towards entropy *as fast as possible,* an extremely efficient conceptual framework is available (this principle could be called "maximum entropy pursuit").

The inversion of the arrow of entropy makes it possible to attack many mysterious phenomena where it seems that improbability cumulates beyond all limits — in the new perspective, such cumulation is no more a paradox. For example, the challenge of *origin of life* can be attacked. The essence of understanding the life processes is, again, in the functions; semantics of life is buried in the notorious concept of *elan vital.* In the neocybernetic setting, such finalistic arguments become issues of entropy production.

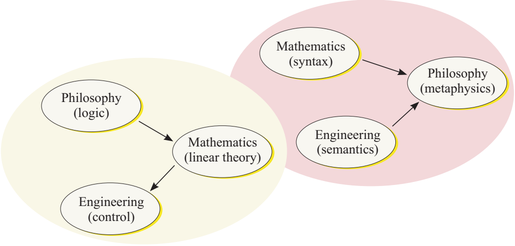


Figure 4: Hierarchy of disciplines. Traditional view, on the left, sees engineering as simple application of the fine theories; the new view, on the right, sees control engineering as delivering substance to philosophies

3.3 Rehabilitation of engineering

Since 1960's, after the great discoveries of modern control theory, there have been no real breakthroughs in the branch of control engineering. It seems that this stagnation does not need to last long: There is a Golden Age of control engineering ahead. Control theory and tools can be applied not only in technical applications, but also in understanding really complex system — biological, social, economical, etc. There do not necessarily exist explicit controls in such systems, but understanding the natural dynamics in such systems is still based on control intuitions.

It is traditionally thought that philosophy is the basis of all science: Logic is part of philosophy determining the rules of sound thinking. Mathematics if "applied logic", implementing the logical structures and manipulating them according to the logical rules. Natural sciences, on the other hand, can be seen as "applied mathematics", where the ready-to-use mathematical formulas are exploited to construct models. Finally, the engineering disciplines are "applied science". Engineering is inferior to the more fundamental ways of structuring the world.

This is a formal view of how deductive science is done, how new truths are derived. However, also these viewpoints need to be reconsidered: If the presented neocybernetic modeling can cast some light onto the mysteries of what is the essence of complex systems, the deepest of the philosophical branches, *metaphysics,* is addressed. It is mathematics that offers the *syntax* for discussing the issues of what is there beyond the observed reality, and it is control engineering that offers the *semantics* into such discussions. It can be claimed that control knowledge is necessary for understanding complex systems, natural or artificial (see Fig. 4), involving intelligent ones.

4 Mastering the environment

The whole cybernetic machinery can be studied in the framework of control and entropy maximization. The feedback needs not be implemented in such an implicit way as in the prototypical cases of "static control", where time axis is abstracted away, but the control intuition extends to dynamic, transitory cases.

4.1 Artificial reflexes

It is *reflexes* that can be seen as atomary manifestations of intelligence, representing reasonable behavior with no brains involved, and automated sensor/motor reactions (conditioned reflexes) can be seen as extensions of that, being learned rather than hardwired. When variations in the environment are interpreted as resources (or threats), low-level intelligence means immediate benefit: Reaching towards resources (or away from them) can be seen as control towards zero activation, or "heat death" of the local environment.

To study "artificial reflexes" the originally static model framework needs to be extended to dynamic cases. As the system was previously seen as a mirror of the environment, now it is the controller that implements *current state as a mirror between the past and the future,* and, what is more, an adapted control should implement some kind of *balance between the past and the future.* One needs to have a model not only of the current environment but also of how it can be changed; one has to be capable of *simulation,* or estimation of the future. In the control engineering perspective, one could speak of *model-predictive control* combined with *dead-beat control* (see Åström and Wittenmark (1997)).

Assume that the observation vector u is coded as a "perception" or "mental view" $\bar{x}$. These variables denote deviations from the nominal reference values, so that in the goal state there should hold $\bar{x} \equiv 0$, meaning extreme loss of variation in the environment. The perceptions are transformed into control signals c that are connected to actuators (muscles). The resulting responses in the environment are then observed and transformed to another set of perceptions. Based on such assumptions, the control scheme in Fig. 5 can be presented. The right-hand side in the figure represents the process of model construction, where the dependencies between the control signal and resulting changes in the observations are recorded, and the left-hand side represents model usage, or on-line construction of the control signals. The model construction is carried out applying the neocybernetic principles for the "input" $\bar{\bar{x}}(k+1)$ and "latent variable" $\bar{\bar{c}}(k)$, now assuming that the control and its effects are in balance. The signals with "double bars" are the balance values after the previous-level signals $\bar{x}$ are further exploited as inputs in the control construction process.

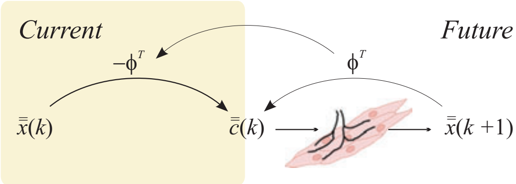


Figure 5: Control as a (antisymmetric) mirror between the past and the future

The intuitive idea of this control is to make the change in the state *inverse* of the current state, meaning that, when the new state is reconstructed as a sum of the old state and the change, the outcome will be zero state — meaning successful control. The past and the future perceptions are "folded" on top of each other. Whereas adaptation of the actuation model can only take place after the effects are visible, the on-line control construction is strictly causal. Coupling the successive perceptions together implements the trick of collapsing the time structure back to singularity.

The above control scheme is based on a very simple view of the environmental dynamics: The assumption now is that if nothing is done, nothing changes in the observed environment. This makes prediction simple. Yet, the causal structure is no more self-evident, as it is no more physical exhaustion of the variation taking place as a side-effect, feedbacks being explicitly implemented, and control structures have to be explicitly initialized. The minor extension of the assumed environment presupposes that different kinds of extensions are implemented in the basic neocybernetic model structure. In general, better control means that new structures need to be employed, and there is more need for explicit control of this control.

4.2 From reactivity to proactivity

When looking at truly complex systems, it turns out that the environment itself consists of other systems. When one has a coevolving system of systems, the role of external disturbations becomes smaller and smaller, and it is interaction among subsystems that mainly takes place. When one knows in advance how the environment will react, controls can be designed beforehand. When the subsystems share the same design instructions, changing the environment can turn from reactive to proactive. Indeed, the genetic (memetic) codes are used for bootstrapping the biological (cognitive) systems, activating dynamic attractors one by one, until the final phenotype is reached.

It seems that no matter how sophisticated structures have been developed during evolution, nature has not found the way to put up a running system without repeating the whole sequence. As observed already by Ernst Haeckel, "ontogeny recapitulates phylogeny".

When a biological system is to be put up in a new organism (or memetic system in another mind), the information can only be stored and transmitted in the form of sequential genetic code (memetic scriptures or spoken tradition). When the environment is favorable, the codes can be interpreted, and the high-dimensional dynamics are started in a hierarchic manner, more complex structures being based on simpler ones, making the system functional and "alive".

5 Conclusion

In today's AI, specially in modern robotics, the approaches are typically based on input/output orientation, "intelligent" systems being black boxes, actually in the spirit of the age-old Turing test. Following the developments in cognitive science, perhaps this Brooksian "artificial behaviorism" will someday change to "artificial cognitivism" or constructivism with emphasis on the internal models and model-based control. Neocybernetics is a candidate offering such model structures.

Good control is based on a good model — it is this model that is the key issue assumedly in all cybernetic systems, however abstract they happen to be. The basic idea does not change if the intuitions cannot easily be turned into concrete numbers. Even all human activities can be interpreted in this framework of finding models and exploiting them for control. For example, scientific activities try to find models for the world — and technology is thereafter employed to exploit this understanding and exhaust the new resources, thus bringing the resource variations to zero.

Arthur Schopenhauer once said that *art* is the only field of human activity that is free of struggle, aesthetic experience making it possible to temporarily escape the "rat-race". In the cybernetic setting, this activity, too, is in line with other goal-directed activities: If art helps to see the world in new perspectives, it makes it possible to construct alternative models. In all, the purpose of life is to understand the world — to find a model, and then exploit the resources.

Indeed, neocybernetic considerations have close connection to philosophies. Along the lines of Eastern wisdom, neocybernetics emphasizes the role of balances in all kinds of living systems. However, in some sense neocybernetics goes deeper than that: Good life is not only about finding the balance. To find the model, to be capable of identifying the environment, there has to exist enough excitation around that balance. It can be assumed that autoimmune diseases, for example, are caused by an incomplete model caused by absence of natural microbial attacks. Full life is not characterized by loss of challenges; rather, *happiness is one's knowledge of being capable of coping with any challenge one can face.*

According to Eastern philosophers, the reason for suffering is missing knowledge and understanding — with appropriate models the world and its constituents become a meaningful whole:

> Before Zen, men are men and mountains are mountains, but during Zen, the two are confused. After Zen, men are men and mountains are mountains again.

References

K.J. Åström and B. Wittenmark. *Computer-Controlled Systems*. Prentice Hall, New Jersey, 1997.

H. Hyötyniemi. *Elastic Systems: Another View at Complexity*. SCAI'06, Helsinki, Finland, 2006a.

H. Hyötyniemi. *Neocybernetics in Biological Systems*. Helsinki University of Technology, Control Engineering Laboratory, 2006b.

I. Prigogine. *End of Certainty*. The Free Press, 1997.

R. Rohde and R. Muller. *Cycles in fossil diversity*. Nature 434, 2005.

$\leq$

$\leq$

≤

≤

≤

≤

≤

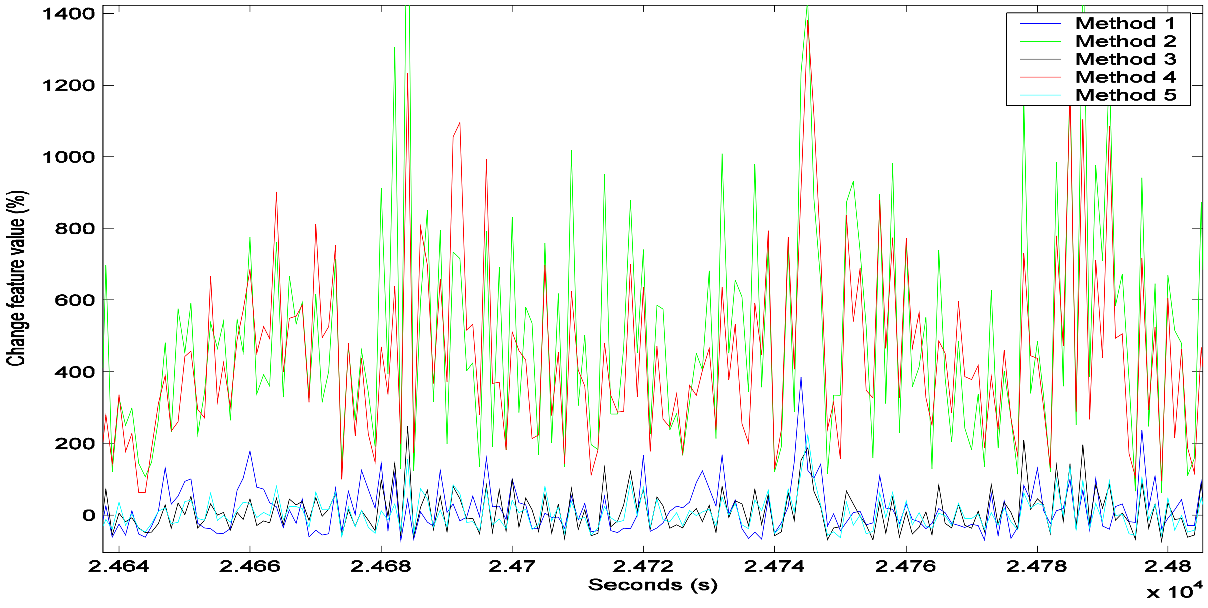
≤

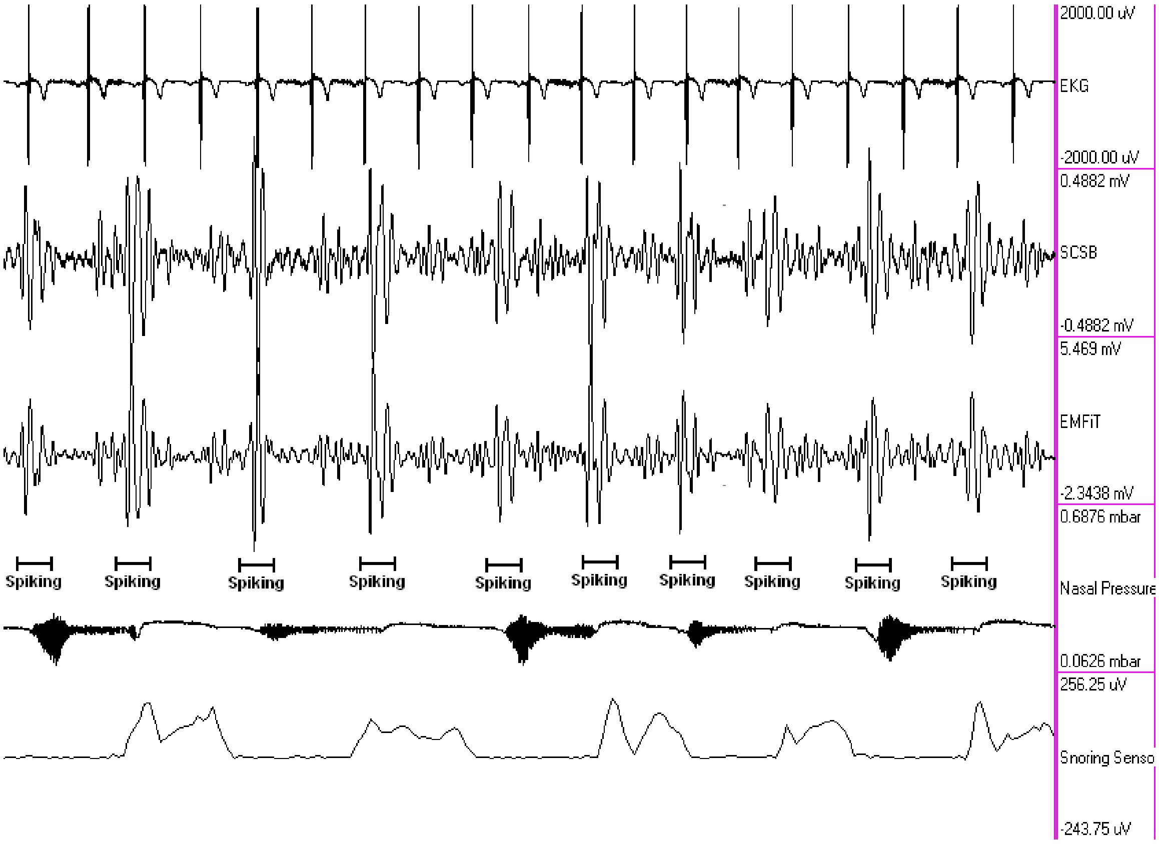
≤

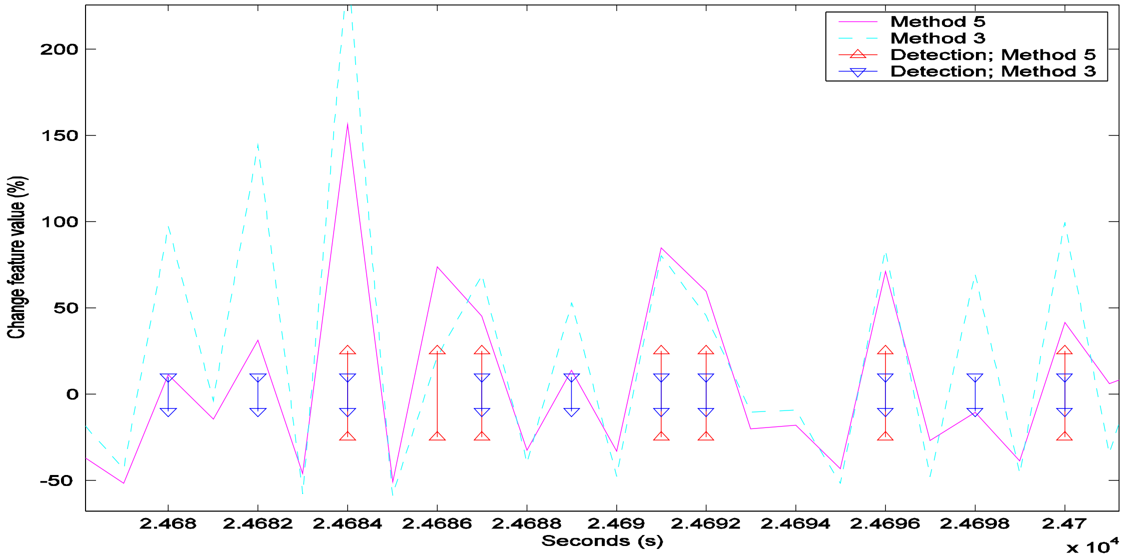
≤

,

≤ ≤


Method 1
Method 2
Method 3
Method 4
Method 5
1400
1200
1000
800
600
400
200
0
Change feature value (%)
2.464
2.466
2.468
2.47
2.472
2.474
2.476
2.478
2.48
Seconds (s)
x 10^4


2000.00 uV
EKG
-2000.00 uV
0.4882 mV
SCSB
-0.4882 mV
5.469 mV
EMFiT
-2.3438 mV
0.6876 mbar
Nasal Pressure
0.0626 mbar
256.25 uV
Snoring Senso
-243.75 uV
Spiking
Spiking
Spiking
Spiking
Spiking
Spiking
Spiking
Spiking
Spiking
Spiking


Method 5
Method 3
Detection; Method 5
Detection; Method 3
Change feature value (%)
200
150
100
50
0
-50
2.468
2.4682
2.4684
2.4686
2.4688
2.469
2.4692
2.4694
2.4696
2.4698
2.47
Seconds (s)
x 10^4

Object-oriented declarative model specification in C++

Risto Lahdelma[*]
[*]University of Turku
Department of Information Technology
Joukahaisenkatu 3, FI-20520 TURKU, Finland
Risto.lahdelma@cs.utu.fi

Abstract

Mathematical optimization models are commonly implemented by specifying them in a high-level declarative modelling language. These languages allow specifying the overall model structure separately from the actual numerical values that are needed to form a specific instance of the model. The existing modelling languages are typically interpreted when the model is generated. In large-scale linear and mixed integer programming the interpretation and generation time may be substantial compared to the time to solve the model. This paper presents MPLC++, which is an object-oriented mathematical modelling language based on C++ classes and objects. Because the model classes are standard C++, they can be compiled and executed efficiently when the model is generated. This approach is particularly suitable in embedded time-critical on-line applications of mathematical optimization models.

Keywords: declarative programming, object-oriented mathematical modelling, linear programming, mixed integer programming, C++

1 Introduction

Mathematical optimization models are commonly implemented by specifying them in a high-level declarative modelling language. Examples of such languages or environments are GAMS /BKM88/, UIMP /ElMi82/, GXMP /Dol86/, AMPL /FGK90/, MPL /Kri91/, PDM /Kri91a/, LPL /Hür93/ and MME /Lah94/. These languages allow specifying the overall model structure separately from the actual numerical values that are needed to form a specific instance of the model. Surveys of modelling languages can be found for example in /GrMu92/, /Kui93/ and /BCLS92/.

Most existing modelling languages require a separate interpreter environment or a pre-processor. Sometimes the modelling language is integrated with a solver, but often the solver is a separate program accepting some intermediate format. Thus, when models are embedded in various design, planning and control applications, the resulting system consists of several programs, which exchange data through large model files. Alternatively it may be possible to link the modelling language interpreter directly into the application. The problem with the above approaches is the overhead caused by

- inter-process communication through files,
- interpreting the modelling language,
- re-generating the internal model format for the solver,
- post-processing and transferring the results back to the application and
- re-initialising the solver in successive optimisation runs with only slightly different parameters.

In large-scale linear and mixed integer programming this overhead may be substantial compared to the time to actually solve the model.

This paper presents an implementation of an object-oriented mathematical modelling language MPLC++ in C++ using classes and instances. Because the model classes are standard C++, they can be compiled and executed efficiently to generate model instances. This approach is particularly suitable in embedded time-critical on-line applications of mathematical optimization models.

2 Object-oriented modelling language in C++

MLC++ is a high-level, hierarchical, object-oriented modelling language, which can be efficiently implemented in C++ using objects and classes. MPLC++ supports modelling with generic, re-usable model components in a way similar to MME (**Mathematical Modelling Environment**) /Lah94/. Model parameters can be inline-coded in C++ or retrieved from auxiliary sources, e.g. external databases. Models can be generated directly into the internal storage format of an integrated optimizer, optimised and the results can be directly retrieved by the application. The basic elements of MPLC++ are implemented as C++ classes according to the class hierarchy shown in Table 1.

Table 1. The basic elements of MPLC++

Class	Abstract syntax
Model	Name Set(Var) Set(Param) Set(Model) Set(Constr)
Constr	Name Cexp <= Linexp <= Cexp
Linexp	Var | Cexp | Linexp (+ | -) Linexp | Linexp (* | /) Cexp
Var	Name Min Max Unit
Real	
Integer	
Binary	
Cexp	Param | Val | Cexp (+ | - | * | /) Cexp
Param	Name Val Unit

2.1 Models

An MPLC++ model consists of a name, a set of decision variables, a set of parameters, a set of constraints between decision variables and parameters and a set of submodels. A model is an implicit relation between decision variables and parameters. New models and submodels are derived from the predefined class **Model**. Decision variables, parameters and submodels may be specified in the private, protected and public section or in the constructor according to the needs for access control. Constraints are defined in the class constructor embedded in executable C++ code. Example:

```
class MyModel : public Model {
    Var x;        // private decision
                  // variables, parameters
                  // submodels
public:
    Var y, z;     // externally accessible
                  // variables, parameters
                  // and submodels

    // constructor
    MyModel(char name[], int n) {
      // local variables, parameters and
      // sub models
     ...
      // definition of constraints
      ...
    }
    // other methods
}
```

Models can be instantiated from model classes with different arguments. Models instantiated from inside another model become components of the surrounding model and a tree-structure of models is formed. The C++ dot notation can be conveniently used for specifying a path to the desired component. If submodels are instantiated from outside any model, they become independent models, forming thus a forest of models.

```
MyModel m1("m1", 2), m2("m2", 5);     // two
independent models m1 and m2
```

Sometimes it is convenient to write a root-model directly into a C++ program without defining a class. The model scope is introduced using the MODELSCOPE-preprocessor macro, which accepts the name of the model as a parameter. The model scope is closed using the ENDMODEL macro.

```
MODELSCOPE(m);         // Name=m
... // definition of decision variables and
    // parameters
... // instantiation of submodels,
    // definition of constraints

  MyModel m1("m1", 2), m2("m2", 5);
// submodels m.m1 and m.m2

ENDMODEL;

m.maximize(m.m1.z+m.m2.z);    // optimise
linear objective
```

2.2 Decision variables

Associated with a decision variable are a symbolic name, a range of allowed values and the unit in which it is expressed. Decision variables are defined directly in C++ programs as instances of the **Var** class or the derived classes **Real**, **Integer**, **Binary**. For example:

```
Var x("x", 0, 100, "kg");     // Name="x",
Min=0, Max=100, Unit="kg"
Real y("DB001"), z;   // Min=0, Max=INF,
Unit= ""
```

Here x becomes a continuous decision variable in the range[0,100], measured in kilograms. The name x occurs twice in the previous definition. The C++ variable x is defined as a reference to a decision variable with run-time name "x". The C++ name x can be used in the succeeding C++ code where constraints and submodels are defined. The run-time name "x" is stored in a symbol table. Without a run-time symbol table it would be impossible to interpret a character string as a variable or to attach meaningful names to a listing of variable values. The C++ name and run-time name may differ as with y in the previous example. If the run-time name is omitted, the system will automatically generate a unique run-time name.

All arguments are optional. The unit defaults to an empty string signifying a dimensionless quantity. Real and Integer variables are by default non-negative, which is common in mathematical programming. Binary variables are by default 0/1 variables.

2.3 Parameters

Parameters are named symbolic constants, which can be defined directly in C++ programs as instances of the Param class or possible subclasses thereof. Example:

```
Param a("a",10,"g");  // Name="a", Val=10,
                      //  Unit="g"
```

When parameters are used instead of ordinary C++ constants or variables, the system is able to keep track of how the model depends on the parameters. This makes it possible to support various parametric analysis techniques. It is also possible to implement spreadsheet-like incremental model re-generation techniques when some parameters are modified.

2.4 Constraints

We consider here only linear constraints. Constraints may have a C++ name and a run-time name. Linear constraints can be double inequalities with constant left and right hand sides or a single inequality or equality between two linear expressions. Unit compatibility is automatically checked in expressions and constraints. Many standard units are built into the system, but new units and conversion rules can also be defined. Incompatible units result in run-time errors. Unit conversions can be used for scaling expressions. For example

```
Constr b("b");          // name “b” for
0 <= 2*y-3*z <= 9;      //  double inequality
1+y >= 2-z;             // unnamed constraint
x == 10*(y+z);          // run-time error,
                        //  unit mismatch
x == a*(y+z);           // ok because
                        //  a = 10g = 0.010kg
x/Unit("g") == 10*(y+z);// x in lhs is
                          //  scaled by 1000
```

3. Extensions

The C++ inheritance mechanism can be used with MPLC++ in different ways. Model classes can be derived from more advanced model classes than the base class Model. Such model components will inherit decision variables, parameters, submodels and constraints from the parent model. Also more specialised variable and parameter classes can be easily derived from the predefined classes.

There is a need for built-in variable and parameter arrays, symbolic index sets and sequences in MPLC++.

4. Conclusions

It is possible to implement a readable, high level modelling language using C++ classes and pre-processor macros. The complete embedding of **MPLC++** into C++ makes it easy to write integrated modelling and optimisation applications. Because all syntactic processing is done by the C++ compiler, and the resulting model can be directly generated in memory into the internal data structures of a solver, this approach should be very suitable for development of embedded applications with high performance requirements.

References

/BCLS92/ Bharadwaj A., Choobineh J., Lo A., Shetty B.: Model Management Systems: A Survey; Annals of Operations Research, vol. 38, December 1992

/BKM88/ Brooke A., Kendrick D., Meeraus A.: GAMS A User's Guide; The Scientific Press, Redwood City, California 1988

/Dol86/ Dolk D.: A Generalized Model Management System for Mathematical Programming; ACM Transactions on Mathematical Software; Vol 12 No 6; June 1986

/ElMi82/ Ellison E.F.D., Mitra G.: UIMP: User Interface for Mathematical Programming; ACM Transactions on Mathematical Software; Vol. 8, No. 3, September 1982

/FGK90/ Fourer R., Gay D.M., Kernighan B.W.: A Modeling Language for Mathematical Programming; Management Science, Vol. 36, No. 5, May 1990

/GrMu92/ Greenberg H.J., Murphy F.H.: A Comparison of Mathematical Programming Modeling Systems; Annals of Operations Research, vol. 38, December 1992

/Hür93/ Hürlimann T.: LPL: A mathematical programming language; OR Spektrum; Band 15, Heft 1, Springer-Verlag 1993

/Kri91/ Kristjansson B.: MPL Modelling System User Manual, Maximal Software Inc., Iceland 1991

/Kri91a/ Krishnan R.: PDM: A knowledge-based tool for model construction; Decision Support Systems; Vol. 7, No. 4, 1991

/Kui93/ Kuip C.A.C: Algebraic Languages for Mathematical Programming; European Journal of Operational Research, Vol. 67, No. 1, May 1993

/Lah94/ Lahdelma R.: An Object-Oriented Mathematical Modelling System; Acta Polytechnica Scandinavica, Mathematics and Computing in Engineering Series, No. 66, Helsinki 1994, 77 p.

Solving and Rating Sudoku Puzzles with Genetic Algorithms

Timo Mantere and Janne Koljonen
Department of Electrical Engineering and Automation
University of Vaasa
FIN-65101 Vaasa
firstname.lastname@uwasa.fi

Abstract

This paper discusses solving and generating Sudoku puzzles with evolutionary algorithms. Sudoku is a Japanese number puzzle game that has become a worldwide phenomenon. As an optimization problem Sudoku belongs to the group of combinatorial problems, but it is also a constraint satisfaction problem. The objective of this paper is to test if genetic algorithm optimization is an efficient method for solving Sudoku puzzles and to generate new puzzles. Another goal is to find out if the puzzles, that are difficult for human solver, are also difficult for the genetic algorithms. In that case it would offer an opportunity to use genetic algorithm solver to test the difficulty levels of new Sudoku puzzles, *i.e.* to use it as a rating machine.

1 Introduction

This paper studies if the Sudoku puzzles can be optimized effectively with genetic algorithms. Genetic algorithm (GA) is an optimization method that mimes the Darwinian evolution that happens in nature.

According to Wikipedia (2006) Sudoku is a Japanese logical game that has recently become hugely popular in Europe and North-America. However, the first puzzle seems to be created in USA 1979, but it circled through Japan and reappeared to west recently. The huge popularity and addictivity of Sudoku have been claimed to be because it is challenging, but have very simple rules (Semeniuk, 2005).

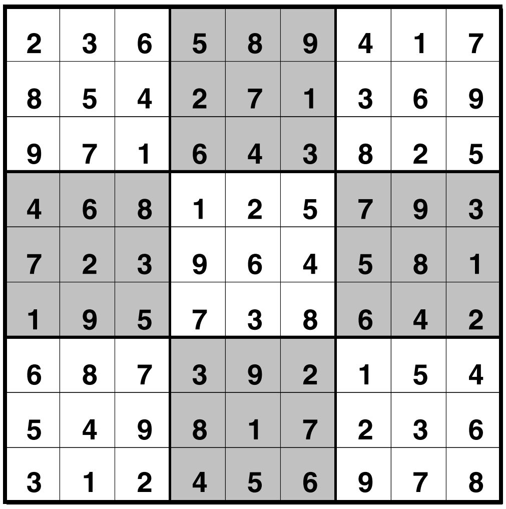
Sudoku puzzle is composed of a 9×9 square that are divided into nine 3x3 sub squares. The solution of Sudoku puzzle is such that each row, column and sub square contains each integer number from [1, 9] once and only once.

In the beginning there are some static numbers (givens) in the puzzle that are given according to the difficulty rating. Figure 1 shows the original situation of the Sudoku puzzle where 30 numbers for 81 possible positions are given. The number of givens does not determine the difficulty of the puzzle (Semeniuk, 2005). Grating puzzles is one of the most difficult things in Sudoku puzzle creation, and there are about 15 to 20 factors that have an effect on difficulty rating (Wikipedia, 2006).

		6		8	9	4		7
				7			6	
	7		6				2	
4		8			5		9	
7								1
	9		7			6		2
	8				2		5	
	4			1				
3		2	4	5		9		

Figure 1. A starting point of the Sudoku puzzle, where 30 locations contains a static number that are given.

Figure 2 shows the solution for the puzzle in fig. 1. Note that each column, row and sub square of the solution contains each integer from 1 to 9 once. The static numbers given in the beginning (fig. 1) are in their original positions.



2	3	6	5	8	9	4	1	7
8	5	4	2	7	1	3	6	9
9	7	1	6	4	3	8	2	5
4	6	8	1	2	5	7	9	3
7	2	3	9	6	4	5	8	1
1	9	5	7	3	8	6	4	2
6	8	7	3	9	2	1	5	4
5	4	9	8	1	7	2	3	6
3	1	2	4	5	6	9	7	8

Figure 2. A solution for the Sudoku puzzle given in figure 1.

The objective of this study is to test if genetic algorithm is an efficient method for solving Sudoku puzzles, and also if it can be used to create generate Sudoku puzzles and test their difficulty levels. If the difficulty rating of the Sudoku puzzles in the newspapers is consistent with their difficulty for GA optimization, the GA solver can also be used as a rating machine for Sudokus.

1.1 Genetic Algorithms

All Genetic algorithms (Holland, 1982) are computer based optimization methods that use the Darwinian evolution (Darwin, 1859) of nature as a model and inspiration. The solution base of our problem is encoded as individuals that are chromosomes consisting of several genes. On the contrary to the nature, in GAs the individual (phenotype) is usually deterministically derived from the chromosome (genotype). These individuals are tested against our problem represented as a fitness function. The better the fitness value an individual gets, the better the chance to be selected to a parent for new individuals. The worst individuals are killed from the population in order to make room for the new generation. Using crossover and mutation operations GA creates new individuals. In crossover genes for a new chromosome are selected from two parents using some preselected practice, e.g. one-point, two-point or uniform crossover. In mutation, random genes of the chromosome are mutated either randomly or using some predefined strategy. The GA strategy is elitist and follows the "survival of the fittest" principles of Darwinian evolution.

1.2 Related work

Sudoku is a combinatorial optimization problem (Lawler *et al*, 1985), where each row, column, and 3×3 sub squares of the problem must have each integer from 1 to 9 once and only once. This means that the sum of the numbers in each column and row of the solution are equal. Therefore, as a problem Sudoku is obviously related to the ancient magic square problem (Latin square), where different size of squares must be filled so, that the sum of each column and row are equal. The magic square problem has been solved by GAs (Alander *et al*, 1999, Ardel, 1994) and also the Evonet Flying circus (Evonet, 1996) has a demonstration of a magic square solver.

Another related problem is a generating threshold matrix for halftoning (Kang, 1999) grayscale images. In the halftoning, the gray values of an image are converted into two values, black or white, by comparing the value of each pixel to the value of the corresponding position at the threshold matrix. The values in the threshold matrix should be evenly distributed. Therefore the sums of the threshold values in each row and column of the threshold matrix should be nearly equal. Furthermore, the demand of homogeneity holds also locally, *i.e.* any fixed size sub area should have a nearly evenly distributed threshold value sum. This guarantees that the resulting image does not contain large clusters of black or white pixels. Threshold matrices have been previously optimized by GAs *e.g.* in (Alander *et al*, 1998 and 1999; Kobayashi and Saito, 1993; Newbern and Bowe, 1997; Wiley, 1998).

There seems to be no scientific papers on Sudoku in the research indices, but there are some white papers on the Internet. In (Gold, 2005) a GA is used to generate new Sudoku puzzles, but the method seems inefficient, since in their example the GA needed 35700 generations to come up with a new puzzle. In our results, we created a new Sudoku, in average, in 1390 generations.

There is also a 'Sudoku Maker' (2006) software available that is said to use genetic algorithm internally. It is claimed that the generated Sudokus are usually very hard to solve. Unfortunately, there are no details, how GA is used and how quickly a new Sudoku puzzle is generated.

Sudoku can also be seen as constrain satisfaction problem, where all the row and column sums must be equal to 45. Constrained optimization problems have been efficiently optimized with evolutionary algorithms e.g. in (Li *et al*, 2005; Mantere, 2005; Runarsson and Yao, 2000).

2 The proposed method

Sudoku is a combinatorial optimization problem, where each row, column and also each nine 3×3 sub square must have a number from {1, 2, …, 9} exactly once. Therefore we need to use a GA that is designated for combinatorial optimization problems. That means that it will not use direct mutations or crossovers that could generate illegal situations: rows, columns, and sub squares would contain some integer from [1, 9] more than once, or some integers would not be present at all. In addition, the genetic operators are not allowed to move the static numbers that are given in the beginning of the problem (givens).

Consequently, we need to represent the Sudoku puzzles in GA program so that the givens will be static and cannot be moved or changed in genetic operations. For this purpose we have an auxiliary array containing the givens. We decided to present Sudoku puzzle solution trials as an integer array of 81 numbers. The array is divided to nine sub blocks (building blocks) of nine numbers corresponding to the 3×3 sub squares from left to right and from top to bottom.

The crossover operation is applied so that it exchanges whole sub blocks of nine numbers between individuals. Thus the crossover point cannot be inside a building block (fig. 2).

The mutations are applied only inside a sub block. We are using three mutation strategies that are commonly used in combinatorial optimization: swap mutation, 3-swap mutation, and insertion mutation (fig. 3).

Each time mutation is applied inside the sub block, the array of givens is referred. If it is illegal to change that position, we randomly reselect the positions and recheck until legal positions are found. In swap mutation, the values in two positions are exchanged. In 3-wap mutation the values of three positions are rotated, either clockwise or counterclockwise. In insertion mutation, one or more numbers are inserted before some number and all the freely changeable numbers are then rotated clockwise or counterclockwise.

To design such a fitness function that would aid a GA search is often difficult in combinatorial problems (Koljonen and Alander, 2004). In this case we decided to use to a simple fitness function that penalizes different constraint violations differently.

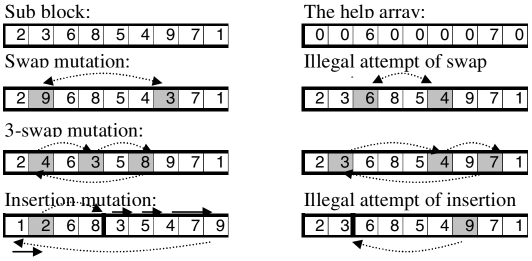


Figure 4. The types of mutation used in Sudoku optimization. Up left is one sub block and up right the static values of that sub block (6 and 7 are givens). The mutation is applied so that we randomly select positions inside the sub block, and then compare the selected positions to the help array if the positions are free to change.

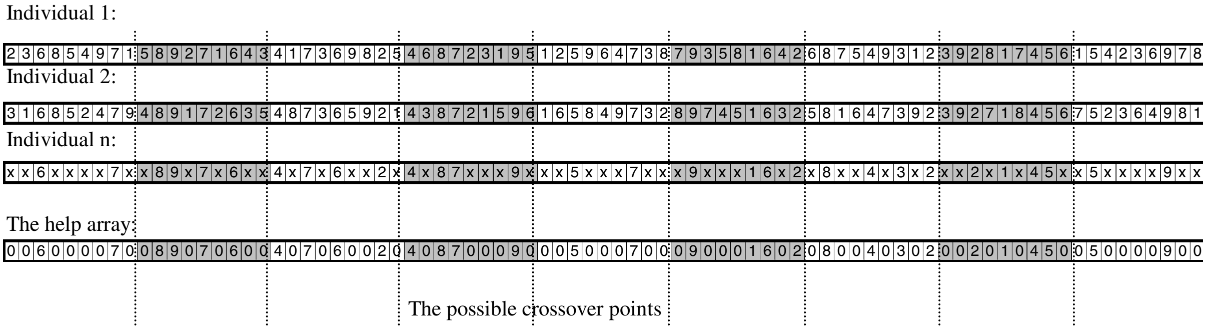


Figure 3. The representation of Sudoku puzzles in GA. One possible solution (individual) is an array of 81 numbers that is divided into nine sub blocks of nine numbers. The crossovers points can only appear between sub blocks (marked as vertical lines). The auxiliary array is used for checking static positions, if there is a number that is not equal to zero that number cannot be changed during the optimization, only the positions that have zero are free to change.

The condition that every 3×3 sub square contains a number from 1 to 9 is guaranteed intrinsically. Penalty functions are used to force the other conditions.

Each row and column of the Sudoku solution must contain each number between [1, 9] once and only once. This can be transformed to a set inequality constraints. The first two equations (1) require that row and column sums should be equal to 45 and the other two equations (2) require that each row and column product should be equal to 9!.

$$\begin{aligned} g_{i1}(x) &= \left|45 - \sum_{j=1}^{9} x_{i,j}\right| \\ g_{j1}(x) &= \left|45 - \sum_{i=1}^{9} x_{i,j}\right| \end{aligned} \tag{1}$$

$$\begin{aligned} g_{i2}(x) &= \left|9! - \prod_{j=1}^{9} x_{i,j}\right| \\ g_{j2}(x) &= \left|9! - \prod_{i=1}^{9} x_{i,j}\right| \end{aligned} \tag{2}$$

Another requirement is derived from the set theory. It is required that the each row x_i and column x_j, must be equal to set A = {1, 2, …, 9}. The functions (3) calculate the number of missing numbers in each row (x_i) and column (x_j) set.

$$\begin{aligned} A &= \{1,2,3,4,5,6,7,8,9\} \\ g_{i3}(x) &= \left|A - x_i\right| \quad , \\ g_{j3}(x) &= \left|A - x_j\right| \end{aligned} \tag{3}$$

where | · | denotes for the cardinality of a set.

In the optimal situation all constraints (1), (2), and (3) should be equal to zero. The overall fitness function is a linear combination of the partial cost functions (1–3):

$$\begin{aligned} f(x) = 10*(\sum_i g_{i1}(x) + \sum_j g_{j1}(x)) + \sum_i \sqrt{g_{i2}(x)} + \\ \sum_j \sqrt{g_{j2}(x)} + 50*(\sum_i g_{i3}(x) + \sum_j g_{j3}(x)) \end{aligned} \tag{4}$$

The fitness function (4) was chosen after a series of tests of different ideas for constraints and parameter values. It penalizes most heavily if some numbers of set A are absent in the row or column set. It also penalizes if sums and products of a row or column are not equal to the optimal situation. This fitness function setting worked satisfactorily, but in the future we need more consideration whether or not this is a proper fitness function for the Sudoku problem.

3 Experimental results

In order to test the proposed method we decided to use an integer coded GA. The size of the GA chromosome is 81 integer numbers, divided into nine sub blocks of nine numbers. The uniform crossovers were applied only between sub blocks, and the swap, 3-swap and insertion mutation with relative probabilities 50:30:20, respectively, inside sub blocks, with an overall mutation probability 0.12. The population size was 100, and elitism 40.

We tested five Sudoku puzzles taken from the newspaper Helsingin Sanomat (2006) marked with their difficulty rating `1-5 stars`. These had 28 to 33 symmetric givens. We also tested four Sudokus taken from newspaper Aamulehti (2006), they were marked with difficulty ratings: `Easy`, `Challenging`, `Difficult`, and `Super difficult`. They contained 23 to 36 nonsymmetrical givens.

We also generated new Sudoku puzzles (no given numbers in the beginning), marked with difficulty rating: `New Sudoku`. We tried to solve each of the ten Sudoku puzzles 100 times. The stopping condition was the optimal solution found, or max. 100000 generation, *i.e.* max. six million trials (fitness function calls).

The results are given in table 1. The columns (left to right) stand for: difficulty rating, the count of how many of the 100 optimization runs found optimal solution, and minimum, maximum, average, median as well as standard deviation calculated only from those test runs that found the solution (showed in count section). The statistics represent the amount of generations required to find the solution.

Table 1 shows that the difficulty ratings of Sudoku's correlate with their GA hardness. Therefore, the more difficult Sudokus for a human solver seem to be also the more difficult for the GA. The GA found the solution for Sudoku's with difficulty rating `1 star` and `Easy` every time. The `Easy` from Aamulehti was the most easiest to solve in general, even easier than to generate a `New Sudoku`.

With other ratings the difficulty increased somewhat monotonically with the rating, and the count of those optimization runs that found a solution decreased. In addition, the average and median GA generations needed to find solution increased.

Table 1. The comparison of how effectively GA finds solutions for the Sudoku puzzles with different difficulty ratings. The rating `New` means no givens (new Sudoku) and numbers 1-5 are Sudoku puzzles with symmetric givens and their difficulty ratings taken from the newspaper Helsingin Sanomat (2006). The Sudokus with ratings `Easy` to `Super difficult` are taken from newspaper Aamulehti (2006) and they had nonsymmetrical givens. *Count* shows how many of the 100 GA optimization runs found the solution and other columns are the descriptive statistics of how many GA generations was needed to find the solution.

Difficulty rating	*Givens*	*Count*	*Min*	*Max*	*Average*	*Median*	*Stdev*
New	0	100	206	3824	1390.4	1089	944.67
1 `star`	33	100	184	23993	2466.6	917	3500.98
2 `stars`	30	69	733	56484	11226.8	7034	11834.68
3 `stars`	28	46	678	94792	22346.4	14827	24846.46
4 `stars`	28	26	381	68253	22611.3	22297	22429.12
5 `stars`	30	23	756	68991	23288.0	17365	22732.25
`Easy`	36	100	101	6035	768.6	417	942.23
`Challenging`	25	30	1771	89070	25333.3	17755	23058.94
`Difficult`	23	4	18999	46814	20534.3	26162	12506.72
`Super difficult`	22	6	3022	47352	14392	6722	17053.33

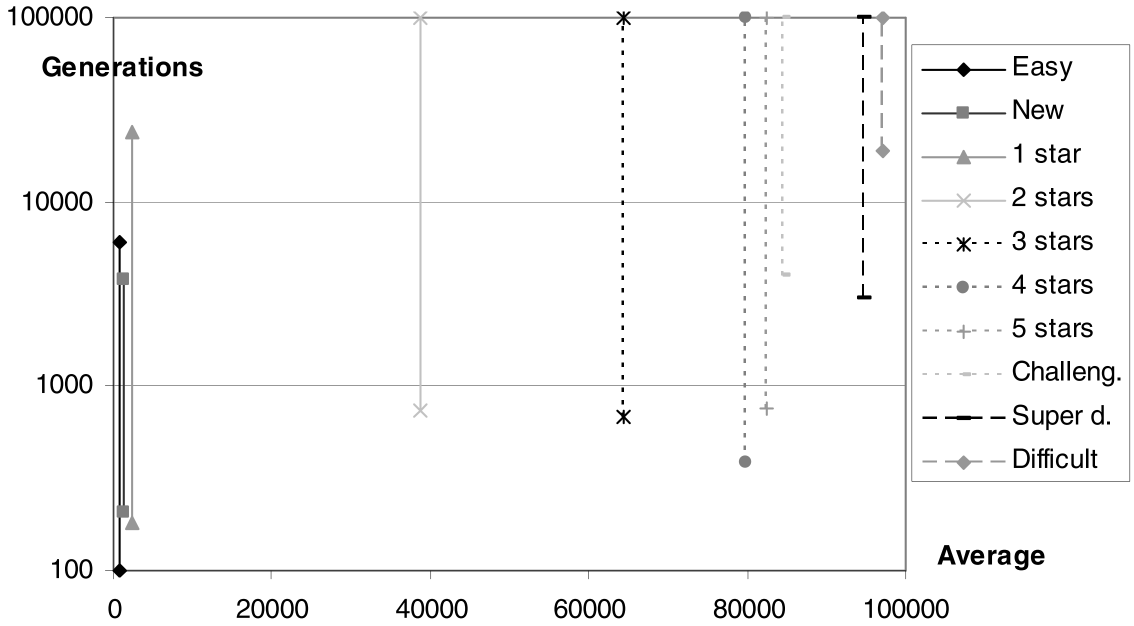


Figure 5. The difficulty order of tested Sudokus. The minimum and maximum generations needed to solve each Sudoku from 100 test runs as a function of average generations needed.

However, the most difficult puzzle (`5 stars`) from Helsingin sanomat was solved 23 times out of 100 test runs, which was almost as often as `4 stars` puzzle (26 times). With the exception of `Easy` the other Sudokus in Aamulehti were found to be more difficult than Sudokus in Helsingin sanomat. The Sudoku with rating `Challenging` was almost as difficult as `4` and `5 stars` Sudokus in Helsingin sanomat.

The puzzles `Difficult` and `Super Difficult` of Aamulehti were much more difficult than any of the puzzles in Helsingin sanomat. The difficulty rating of these two also seemed to be in wrong mutual order (fig. 5), since `Difficult` was harder for GA than `Super Difficult`. However, both were so difficult that GA found the solution only a few times, so the difference is not statistically significant.

Figure 5 shows the difficulty order of the tested Sudokus. The {Easy, New Sudoku, 1 star} seems to form a group of the most easiest Sudokus, while 2 and 3 stars Sudokus lie by themselves between the previous and the group containing {4 stars, 5 stars, Challenging}. The two most difficult Sudokus {Super difficult, Difficult } are in the final group.

The New Sudoku (no givens) is the second easiest for GA in average. This means that GA can be used for generating new puzzles. It takes for GA between [406, 3817] generations ([24400, 229060] trials) to create a new open Sudoku solution that can further be used to create a new puzzle.

If GA is used as puzzle generator, one first searches any possible open solution and then removes some numbers and leave the givens. The difficulty ratings of new Sudoku can also be tested with GA by trying to solve it after the givens are selected.

4 Conclusions and future

In this paper, we studied if Sudoku puzzles can be solved with a combinatorial genetic algorithm. The results show that GA can solve Sudoku puzzles, but not very effectively. There exist more efficient algorithms to solve Sudoku puzzles (Wikipedia, 2006). However, in this study the aim was to test how efficient pure GA is, without problem specific rules, for solving Sudokus. The GA results can of course be enhanced by adding problem related rules.

The other goal was to study if difficulty ratings given for Sudoku puzzles in newspapers are consistent with their difficulty for GA optimization. The answer to that question seems to be positive: Those Sudokus that have a higher difficulty rating proved to be more difficult also for genetic algorithms. This also means that GA can be used for testing the difficulty of a new Sudoku puzzle.

Grading puzzles is said to be one of the most difficult tasks in Sudoku puzzle creation (Semeniak, 2005), so GA can be a helpful tool for that purpose.

The new puzzles are usually generated by finding one possible Sudoku solution and then removing numbers as long as only one unique solution exists. The GA can also be used to select the removed numbers and also testing if there is still only one unique solution. This can be tested e.g. by solving the Sudoku 10 or more times with other GA, and if the result is always identical, we can be relatively sure (but not completely) that a unique solution exists. However, it may be wiser to use more effective algorithms to test the uniqueness of a solution.

The future research may consist of generating a more efficient hybrid GA and algorithms created specifically to solve Sudoku. Another research area would be to study if it is possible to generate a fitness function based on an energy function (Alander, 2004).

It has been said that 17 given number is minimal to come up with a unique solution, but it has not been proved mathematically (Semeniak, 2005). The GA could also be used for minimizing the number of givens and still leading to a unique solution.

Acknowledgements

The research work of the author T.M. was funded by the research grant from the TietoEnator fund of the Finnish Cultural Foundation.

References

Aamulehti. *Sudoku online*. Available via WWW: http://www.aamulehti.fi/sudoku/ (cited 11.1.2006)

Alander, J.T. Potential function approach in search: analyzing a board game. In J. Alander, P. Ala-Siuru, and H. Hyötyniemi (eds.), *Step 2004 – The 11th Finnish Artificial Intelligence Conference*, Heureka, Vantaa, 1-3 September 2004, Vol. 3, Origin of Life and Genetic Algorithms, 2004, 61-75.

Alander, J.T., Mantere T., and Pyylampi, T. Threshold matrix generation for digital halftoning by genetic algorithm optimization. In D. P. Casasent (ed.), *Intelligent Systems and Advanced Manufacturing: Intelligent Robots and Computer Vision XVII: Algorithms, Techniques, and Active Vision*, volume SPIE-3522, Boston, MA, 1-6 November 1998. SPIE, Bellingham, Washington, USA, 1998, 204-212.

Alander, J.T., Mantere T., and Pyylampi, T. Digital halftoning optimization via genetic algorithms for ink jet machine. In B.H.V. Topping (ed.) *Developments in Computational mechanics with high performance computing*, CIVIL-COMP Press, Edinburg, UK, 1999, 211-216.

Ardel, D.H.. TOPE and magic squares, a simple GA approach to combinatorial optimization. In J.R. Koza (ed.) *Genetic Algorithms in Stanford*, Stanford bookstore, Stanford, CA, 1994.

Darwin, C. *The Origin of Species: By Means of Natural Selection or The Preservation of Favoured Races in the Struggle for Life*. Oxford

University Press, London, 1859, A reprint of the 6th edition, 1968.

Evonet Flying Circus. *Magic square solver*. Available via WWW: http://evonet.lri.fr/CIRCUS2/node.php?node=65 (1996, cited 11.9.2006)

Gold, M.. *Using Genetic Algorithms to Come up with Sudoku Puzzles*. Sep 23, 2005. Available via WWW: http://www.c-sharpcorner.com/UploadFile/ mgold/Sudoku09232005003323AM/Sudoku.aspx ?ArticleID=fba36449-ccf3-444f-a435-a812535c45e5 (cited 11.9.2006)

Helsingin Sanomat. *Sudoku*. Available via WWW: http://www2.hs.fi/extrat/sudoku/sudoku.html (cited 11.1.2006)

Holland, J. *Adaptation in Natural and Artificial Systems*. The MIT Press, 1992.

Kang, H.R.. *Digital Color Halftoning*. SPIE Optical Engineering Press, Bellingham, Washington, & IEEE Press, New York, 1999.

Kobayashi, N., and Saito. H. Halftone algorithm using genetic algorithm. In *Proc. of 4th Int. Conference on Signal Processing Applications and Technology*, vol. 1, Newton, MA 28. Sept. – 1. Oct. 1993, DSP Associates, Santa Clara, CA, 1993, 727-731

Koljonen, J. and Alander, J.T. Solving the "urban horse" problem by backtracking and genetic algorithm – a comparison. In J. Alander, P. Ala-Siuru, and H. Hyötyniemi (eds.), *Step 2004 – The 11th Finnish Artificial Intelligence Conference*, Heureka, Vantaa, 1-3 September 2004, Vol. 3, Origin of Life and Genetic Algorithms, 2004, 127-13.

Lawler, E.L., Lentra, J.K., Rinnooy, A.H.G., and Shmoys, D.B. (eds.). *The Traveling Salesman problem – A Guided Tour of Combinatorial Optimization.* John Wiley & Sons, New York, 1985.

Li, H., Jiao, Y.-C., and Wang, Y.: Integrating the simplified interpolation into the genetic algorithm for constrained optimization problems. In Hao *et al.* (eds) *CIS 2005, Part I, Lecture Notes on Artificial Intelligence* 3801, Springer-Verlag, Berlin Heidelberg, 2005, 247-254.

Mantere, T. A min-max genetic algorithm for min-max problems. In Wang, Cheung, Liu (eds.) *Advances in Computational Intelligence and Security - The Workshop of 2005 Int. Conference on Computational Intelligence and Security - CIS 2005*, Xi'an, China, December 15-19, 2005. Xidian university press, Xi'an, China, 2005, pp. 52-57.

Newbern, J., and Bowe Jr., M. Generation of Blue Noise Arrays by Genetic Algorithm. In B.E.Rogowitz and T.N. Pappas (eds.) *Human Vision and Electronic Imaging II*, San Jose, CA, 10.-13. Feb. 1997, Vol SPIE-3016, SPIE - Int. society of Optical Engineering, Bellingham, WA, 1997, 441-450.

Runarsson, T.P., and Yao, X.: Stochastic ranking for constrained evolutionary optimization. *IEEE Transactions on Evolutionary Computation* **4**(3), 2000, pp. 284-294.

Semeniuk, I. Stuck on you. In *NewScientist* **24/31** December, 2005, 45-47.

Sudoku Maker. Available via WWW: http://sourceforge.net/projects/sudokumaker/ (cited 27.1.2006)

Wikipedia. *Sudoku*. Available via WWW: http://en.wikipedia.org/wiki/Sudoku (cited 11.9.2006)

Wiley, K.: Pattern Evolver, an evolutionary algorithm that solves the nonintuitive problem of black and white pixel distribution to produce tiled patterns that appear grey. In *The Handbook of Genetic Algorithms*. CRC Press, 1998.

Using SOM based resampling techniques to boost MLP training performance

Jussi Ahola[*]
Technical Research Centre of Finland[*]
FI-02044 VTT, Finland
Jussi.Ahola@vtt.fi

Mikko Hiirsalmi[†]
Technical Research Centre of Finland[†]
FI-02044 VTT, Finland
Mikko.Hiirsalmi@vtt.fi

Abstract

We describe SOM based modeling work that has been conducted in a research project that aims at developing a flight parameter based fatigue life analysis system for the Finnish Air Force F-18 fighters. The idea of using SOM based clustering as a basis for iteratively selecting poorly modeled test samples to be included into the training set has been investigated. We call these schemes SOM resampling. Our tests indicate that clear gains could be achieved in the test cases when measured in terms of the mean squared error criterion as typical in MLP network training.

1. Introduction

This work constitutes a part of a larger project to develop a flight parameter based data analysis system to estimate flight specific fatigue life expenditure for the Finnish Air Force F-18 fighters. In Figure 1, the general analysis chain of the system is shown. In the analysis chain MLP[1] neural networks have been used to learn regression models to estimate strain in a selected position near each desired critical structural detail. The strain values are adjusted to the desired position through finite element models and fatigue life analysis software is used to estimate the fatigue life expenditure of whole flights. Therefore there is no direct link between the neurally estimated strain values and the fatigue life expenditure estimates. During the operational phase of the system the measurement data is based on currently available instrumentation onboard the aircraft. These data points are preprocessed by filtering and interpolation techniques to produce refined time series data. For the training phase the onboard data has been supplemented with additional strain gauge measurement producing the real strain values in the selected positions. These values are targets for the neural network models. One neural network model has been created for each structural detail. Early work with the system design and the neural network training have been documented in (Salonen, 2005).

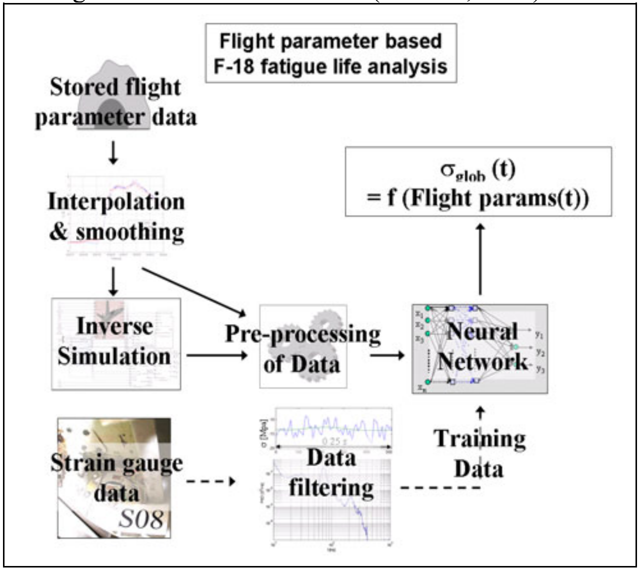


Figure 1: The analysis chain used in proof of concept study to predict strain gauge response (Tikka, Ahola and Hiirsalmi, 2006)

Currently the amount of training data is rather limited covering 25 flights where each flight consists of a temporal sequence of flight manoeuvres. In the coming years hundreds of new flights are going to be carried out with two airplanes equipped with the strain gauge measurements allowing one to gather new data for neural network training during their normal operation. As the

[1] MLP means Multi Layered Perceptron being one form of so called feed forward neural network architectures

amount of the data grows the neural network models are going to be relearned. The input variables have been selected by aeronautical engineers to reflect meaningful dependencies. The selection of training, validation and test data sets is another critical area in neural network modelling. Data selection issues and model complexity have been discussed in (Bishop, 1995) and in (Hastie, Tibshirani and Friedman, 2001). The fundamental idea in the selection of the datasets is that they represent well the phenomena that the neural network is being trained to predict. The following points should be considered:

- The data sets should all come from the same distribution of data, i.e. they should be sampled from the same distribution. Bias between the data sets is harmful for gaining good prediction results.

- There must be enough data samples to be able to learn statistically accurate model parameters. The complexity of the network has an impact on the number of training samples needed for learning. On the other hand, if the complexity of the underlying model is too small the achievable modelling accuracy will be diminished.

- In MLP network training the distribution of the training data samples should be adequately balanced so that there are approximately equal amounts of samples in each value segment of the target variable because typical training session tries to minimize the mean squared error and areas with lower amounts of samples will not be optimized as well. On the other hand, it is also possible to stress important target value ranges by over-representing those samples.

Training data is used by the MLP algorithm to learn the weight and bias parameters. Testing data is used during the training session to evaluate how well the learned neural networks perform on previously unseen data. The best performing network is selected for production usage. Validation data is used to evaluate the performance of the selected network on previously unseen data. It should give a measure of the average behaviour of the network on previously unseen data. The data selection methods used in the project have been better described in (Salonen, 2005) and in (Tikka, Ahola and Hiirsalmi, 2006). One of the problems present is that the overall life cycle performance has not been evaluated with totally unseen test data – in the testing phase the whole time series data has been used to evaluate the performances. This may produce optimistic performance results in cases where extreme strain samples are overly represented in the training data set which generally raises the probability of over-fitting. A better evaluation method could be achieved by leaving a few flights for testing purposes or by averaging the results of several training sessions where a small amount of different flights are left for testing. These tests must wait until we get new training data from further operational flights.

Our role in this project has originally dealt with validation of the modelling results by using SOM[2] modelling in a qualitative sense in order to find input data clusters reflecting typical flight states and looking for systematic errors in the prediction results. Cluster specific prediction error histograms and time series plots which have been coloured with the cluster colours have been used to search for problematic areas. As an extension it was considered worthwhile to investigate whether the learning data set could be adjusted iteratively during the training cycle to automatically improve the balance of the learning data samples. The basic idea has been that if a SOM map unit has a large portion of prediction errors on the test data samples these samples may be poorly represented in the learning data set and further samples should be selected from this area. In some areas the distribution of prediction errors may also be rather flat around the zero level meaning that the data set is inconsistent or that they do not form a compact input parameter cluster.

2. Resampling methods tested

In this work our research hypothesis has been that the modelling efficiency of feed-forward neural networks can be improved iteratively by selecting a small portion of the less well predicted test samples as additional training samples. We have tested a few approaches to accomplish this task and have evaluated their performance with real measurement data.

In order to set a reference line for evaluations we have at first constructed a simple approach where we compute the squared modelling errors (ei2 = (measuredi-modeledi)2) for test data samples and select the desired amount of the worst predicted test samples as new training samples. This is our sampling method 1.

All the other tested methods are based on SOM clustering information (Kohonen, 2001). At first we

[2] Self-Organizing Map (SOM) is an unsupervised learning method for vector quantizing a high-dimensional data set into a lower dimensional output space while at the same time preserving neighbourhood topology

have computed a 2-dimensional SOM map based on nine explanatory variables of the feature vector of the training data samples (the predicted value has been masked off from this training). We have therefore attempted to find the typical input parameter combinations that reflect so called ‘flight states’. A relatively sparse SOM map has been used for this task in order to get a reasonably small set of codebook vectors for further processing. We have then computed the best matching units (BMUs[3]) for all the training samples and using these samples we have computed the average modelling error for each map unit. We have then computed the BMUs for all the test data samples and we have identified the list of test data samples for each map unit. We use these lists for random selection purposes with the next three sampling methods.

In sampling method 2, we scale the sampling likelihoods of each SOM map unit based on the average modelling error of those training data samples that are closest to that map unit. Within each map unit the sampling likelihood is uniform between the data samples. Therefore, it is more likely to select new samples from the test data samples that have been projected to the map units with the highest average modeling error.

In sampling method 3, we extend the previous method by additionally scaling the selection likelihood of the test data samples by their relative modelling error within their BMU map unit. Therefore, it is more likely to select new samples from the test data samples that have been projected to the map units with the highest modeling data error and among these test data samples we prefer the ones that have the largest modeling error.

In sampling method 4, we extend the previous method by additionally scaling the selection likelihood of the test samples by their quantization error[4] within their BMU map unit. Therefore, it is more likely to select new samples from the test data samples that have been projected to the map units with the highest average modeling error and among these test data samples we prefer the ones that have the largest modeling error and at the same time fit less well to their BMU map unit (those that have large quantization errors).

3. Evaluation test results

In the evaluation tests the initial training data set contained 14995 training samples that were selected from a data set of more than 550000 data samples. The samples contained nine explanatory variables and one target variable to be predicted. The variables were already scaled in a proper way in the range of [-0.8, 0.8]. The proper network complexity had been selected while initially training the original networks and in order to get comparable results we used the same network architecture with 9 input variables, 25 hidden nodes in hidden layer 1 and 1 output variable. Therefore, in this case the network complexity was 276 weight parameters to be optimized. The transfer function used in layer1 was tansig and in the output layer linear. The training algorithm was Levenberg-Marquardt backpropagation algorithm with the performance function being the minimization of the MSE error and with the early stopping threshold value being 5 epochs.

During each resampling iteration we taught 5 networks with random initial states and selected the minimum results on a previously unseen test data set as the reference value for each iteration for each sampling scheme. In the first iteration we computed the starting values with the initial training data set and after that we added 1000 randomly sampled test data samples to the training set and recomputed the MLP networks.

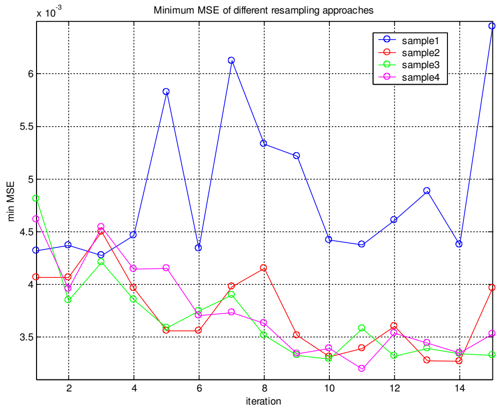


Figure 2: Minimum Mean squared error (MSE) of the tested sampling methods on each iteration step

In Figure 2, we have summarized the performances achieved with the different iterative resampling schemes tested. We have used the MSE error (mean squared error) as the quality criterion as it is the criterion that is being minimized when training our MLP networks. In the figure iteration step 1 marks the initial starting state without the iterative resampling steps. We can see that with the simple sampling scheme, sample1, only moderate gains can be achieved during the first steps but with more steps the results start to oscillate and the accuracy becomes somewhat unstable. The SOM-based schemes (samples 2-4) behave fairly similarly to each other. All of them show clear gains in the modelling accuracy. The achieved accuracies tend to vary between successive iterations but the trend is

[3] BMU means the best-matching unit and it is computed for a data sample as the minimum distance to any of the SOM map unit vectors.

[4] Quantization error means the distance between a data sample and a SOM map unit vector.

towards more accurate results. We can see that sample2 has the biggest oscillations but produces slightly more accurate results than sample3. Sample4 produces fairly consistent, non-oscillatory results and the best accuracy with iteration step 10. This achieved best accuracy produced a 31% gain in accuracy as compared to the starting situation with the initial training data set.

4. Discussion

Our test results indicate that MLP modelling accuracy may be boosted by our SOM based iterative resampling schemes when we measure success with the same performance criterion as when teaching the MLP networks. This indicates that in typical MLP learning our SOM based resampling schemes may be used to boost performance. The techniques are however only preliminary methods and we believe that they could be enhanced by better considering the representativeness of the map units, possibly clustering the map units and by better considering the target variable distribution within each map unit while determining the sampling likelihoods.

In (Tikka, Ahola and Hiirsalmi, 2006) it has however been indicated that no clear gains can be achieved when the performance gains are measured in terms of consumed life time. We believe that this is caused by the fact there is no direct link from the strain values that have been predicted by the MLP neural networks but the effect has been transferred by FEM transfer models and by the life cycle estimation software which inevitably cause noise in the predictions.

SOM based clustering might also be useful in the original data selection phase as an alternative to the K-means clustering previously used in (Tikka, Ahola and Hiirsalmi, 2006).

5. Conclusions

The idea of using SOM based clustering as a basis for iteratively selecting poorly modeled test samples to be included into the training set has been investigated. We call these schemes SOM resampling. Our tests indicate that clear gains could be achieved in the test cases when measured in terms of the mean squared error criterion as typical in MLP network training.

Acknowledgements

This work has been funded by the Finnish Air Force and the Technical Research Centre of Finland. We want to especially thank Jarkko Tikka for setting up research challenges and Risto Laakso for letting us share his SOM modelling work in this domain.

References

Christopher M. Bishop. Neural Networks for Pattern Recognition. Oxford University Press, Oxford, 1995.

T. Hastie, R. Tibshirani and J. Friedman. The elements of statistical learning : data mining, inference and prediction, 2001.

Teuvo Kohonen. Self-Organizing Maps. Springer, 3rd, extended edition, 2001.

Risto Laakso & al., On the Neural Network Applications as Tools to Improve Aircraft Structural Integrity Management Efforts, ICAF 2005 conference 6. – 10.6.2005, Hamburg, Germany.

Tuomo Salonen. An Assessment of the Feasibility of a Parameter Based Aircraft Strain Monitoring Method. M.Sc Thesis at the Technical University of Helsinki, 12.7.2005, www.aeronautics.hut.fi/edu/theses/full_thesis/Salonen_Tuomo_2005.pdf.

Jarkko Tikka, Jussi Ahola, Mikko Hiirsalmi. Parameter based fatigue life analysis, influence of training data selection to analysis performance. American Institute of Aeronautics and Astronautics Modeling and Simulation Technologies Conference, 2006.

Data-Based Modeling of Electroless Nickel Plating

Hans-Christian Pfisterer[⋆]
[⋆]Control Engineering Laboratory
PL 5500
02015 TKK
Finland
h.pfisterer@gmx.de

Heikki Hyötyniemi[†]
[†]Control Engineering Laboratory
PL 5500
02015 TKK
Finland
heikki.hyotyniemi@tkk.fi

Abstract

This work connects the neocybernetic theory of elastic systems to modeling of a complex industrial process. It turns out that using simple mathematics combined with multilinear tools leads to linear models that can be used for monitoring and control. As an application example, electroless nickel plating is studied.

1 Introduction

What has artificial intelligence to do with modeling of chemical processes? It seems that such a very technical application domain would have little in common with the AI approaches. Still, in all complex systems the challenges are similar. One would need to understand *emergence* to escape from the individual low-level behaviors to the level of relevant functionalities.

In the work at hand the industrial process of *electroless nickel plating* is studied. This chemical system is very complex since many chemicals are involved and the process has a lot of variables being observed and controlled in order to reach good plating results. What is more, for commercial reasons not all compounds and reactions have been disclosed. It is very difficult also for a domain expert to understand the behavior of the system.

In this paper, a very simple, data-based model for control purposes of the nickel alloy thickness and its phosphorus content is presented. This model was achieved using basic ideas and mathematical tools based on the theory of *neocybernetics.*

2 Neocybernetics

Since models of industrial systems and observed natural systems get more and more complex and very complicated to understand the traditional way of mathematical modeling with dynamic nonlinear differential equations and constraints leads to an unmanageable mess of functions and equations. Furthermore, one has to be a domain expert to understand the interconnections between variables in order to write the mathematical equations for the description of the behaviors in the first place.

Neocybernetics, introduced by H. Hyötyniemi (2006), offers a new approach to get reasonable system models and helpful system understanding when looking at a system from a different angle. Since industrial systems are monitored extensively the data can be used to find connections among the variables and possibly associate some variables with others. The system structure is hidden in this data and it will emerge when manipulated in an appropriate way. Using the right mathematical tools and a holistic view of the system this modeling machinery is also available for non-experts of the particular domain. Here information about the behavior of the system is retrieved directly from the measurement data.

2.1 Key points

To see a system through neocybernetic eyeglasses some assumptions about the system have to be made. There are some basic principles (see H. Hyötyniemi (2006) and H.-C. Pfisterer (2006)):

- *Emergence:* Some unanticipated functionality can appear after the cumulation of some simple operations.
- *Dynamic Balance:* The emphasis is on systems in some kind of balance rather than on the process itself.
- *Linearity pursuit:* The system behavior is considered to be linear as long as nonlinearity is not absolutely necessary.

- *Environment:* Neoybernetic systems are assumed to be oriented towards their environment, they reflect and mirror their environment.

All these principles and assumptions are reasonable and in many natural and industrial systems they can be fulfilled. The linearity assumption needs a further explanation (see Section 5), especially when modeling chemical systems.

2.2 Elasticity

The system is assumed to be in thermodynamic balance. The changes in the environment are seen as disturbances causing tensions in the system, pushing the system away from the balance. According to the Le Chatelier principle (H. Hyötyniemi, 2006), the system yields, but after the pressure vanishes, the original balance is restored. In a sense, the neocybernetic ideas are a functionalization of the intuitions concerning complex *homeostatic* systems.

2.3 Degrees of freedom

As mentioned above the hidden structure of the system will emerge when the modeling concentrates on the remaining degrees of freedom in behaviors rather than on constraints and restrictions. For simple systems this is not reasonable — for very complex systems where many variables are strongly connected and many constraints have to be considered this approach helps to avoid an unmanageable mess of equations. When concentrating on the non-constrained degrees of freedom, the emphasis is on phenomena that are not explicitly seen in any formulas; one could speak of *emergent models.*

Now the system is kept in dynamic balance and one searches for the structure of covariation that is still left in the variable space. A model based on the degrees of freedom approach is as useful as a traditional one but has the advantage of being simple and understandable. With this “inverse thinking” not all constraints and chemicals and reactions need to be known, as long as it can be assumed that the internal interactions and feedbacks always manage to keep the system in balance. More information about degree of freedom modeling can be found in H. Hyötyniemi (2006) and H.-C. Pfisterer (2006).

3 Multivariate tools

Using the above mentioned key points and utilizing strong mathematical tools a practical modeling machinery can be set up. After applying the modeling tools a black box between known variables and unknown ones to be estimated can be filled with life. For these purposes the known data (column vectors of data samples) is collected in a matrix X and the unknown variables in a target space formed by the matrix Y. Here, it is assumed the k sample vectors of length n and m are stored as rows in the matrices X and Y, respectively.

Principal Component Analysis (PCA)

The information in terms of covariations in the data is captured in the (unscaled) correlation matrix

$$R = X^T X.$$

A lot of hidden information can be revealed by eigenvalues λ_i of this matrix, and by their corresponding eigenvectors θ_i as can be read in H. Hyötyniemi (2001). The eigenvectors are orthogonal and point in the direction of maximum variation in the original dataset. The amount of variety of each particular direction is given by the corresponding eigenvalue. Directions of most variety are assumed to carry most information about the system and it is reasonable to take these into account in the model; hence the model is built on the degrees of freedoms found in the correlation structure of the data.

Large eigenvalues stand for directions of essential information, small eigenvalues stand for directions which most probably contain redundant information or only measurement noise. When the data is projected onto the subspace Z of dimension $N \leq n$ by the mapping matrix $F1$, redundancy and measurement noise in the data can be reduced.

Multilinear Regression (MLR)

To find the connection to the target space Y a regression is applied. In order to achieve good results the base of the starting space should be orthogonal and no redundancy should be there. Using the dataspace Z these prerequisites are fulfilled and the MLR algorithm can find a mapping $F2$ from Z to Y. Hence a combination of both procedures, now called Principal Component Regression (PCR), can explain the behavior in Y by the information already given in X. This is illustrated in Figure 1.

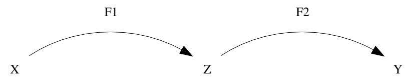


Figure 1: Data spaces and projections with PCR

Partial Least Squares (PLS)

This algorithm includes a view into the target space Y already while building the mapping $F1$. Not only the dataspace X is scanned for the essential information (collected in $Z1$) but also in Y the key information is extracted (to $Z2$) and only then an overall mapping is obtained which now bridges input and output. This steps and transformations are illustrated in Figure 2.

$$\begin{tikzcd} X \arrow[r, bend left, "F1"] & Z1 \arrow[r, bend left, "F2"] & Z2 \arrow[r, bend left, "F3"] & Y \end{tikzcd}$$

Figure 2: Data spaces and projections with PLS

After introducing the industrial process and some specific variables these tools will be applied and their results shown in Section 6.

4 Electroless nickel plating

The process of electroless nickel plating is a step of surface finishing. Printed wiring boards (PWBs) consist of a epoxy laminate base and a copper layer in which electric circuits are etched. Witout protection the copper would oxidize very fast, especially when coming in contact with humidity. In order to enhance the lifetime of the PWB and to improve its mechanical properties the PWB is coated with a gold layer of about $0.01\mu m$. But since the copper would diffuse in the gold and form again an oxidizeable compound the two layers are separated by a nickel layer of around $4\mu m$. Figure 3 shows a cut through a plated PWB and explains the layers in detail; since the gold layer is so thin it can not be seen here.

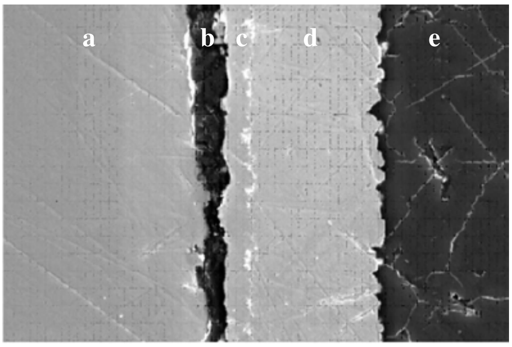


Figure 3: Cut through a plated substrate. Layers: (a) a metallic fastener and (b) a piece of conductive plastic to fix the sample, (c) Ni-P layer, (d) copper layer, (e) the base of PWB (epoxy laminate) (K. Kantola, 2004)

The use of the word “electroless” is misleading; it emphasizes the difference between the method where an external power supply is connected to the substrate which is to be plated and the approach of an active substrate that deposits nickel atoms from ions on its surface. Here nickel is provided in form of ions in an aqueous solution, beside the reducer hypophosphite that provides the necessary electrons for the transformation from nickel ions to real atoms. Due to the used reducer the alloy has a phosphorus content of up to 15 weight percent. The catalytically activated substrate is immersed into the bath and an alloy of nickel atoms and some phosphorus can be built, because nickel itself is also catalytically active itself. There are electric currents in the bath caused by the conversion of ions to atoms, so the process is not really electroless.

The bath consists of many more chemicals which help to keep the bath stable and in a desired status. The composition of these chemicals is either not exactly known or a well kept industry secret which makes the traditional modeling of the bath behavior very difficult or even impossible. Furthermore one has to be a chemical domain expert to understand the connection between the chemicals and their behavior especially when connected to the substrate and its expanding nickel alloy.

The characteristics of the bath are observed continuously and as many variables as possible are measured. Hence online measurements of the current bath nickel concentration, bath temperature and pH value are available and with good controllers kept along desired values to keep the bath constitution constant. The substrate and its characteristics are measured in a laboratory when the plating is finished. Summarizing the dataspaces X and Y (Section 3) basically contain the following information:

X: Nickel concentration, pH value, ammonia concentration, ammonia and nickel sulfate pumping, plating area and temperature.

Y: Alloy thickness, phosphorus content, hypophosphite concentration and orthophosphite concentration.

The original dataset X is further prepared and expanded as explained in the following section. The information about hypophosphite and orthophosphite is not really a plate characteristic and for that reason not further studied here.

5 Linearity

The system is in a state of thermodynamic balance. This state is kept constant by a very strong and accurate control mechanism provided by the bath surrounding. Balance also is one of the neocybernetic key points and so is very important to this approach. The thermodynamic equilibrium can be described by the constant

$$K = \frac{C_{\mathrm{B}_1}^{b_1} \cdots C_{\mathrm{B}_\beta}^{b_\beta}}{C_{\mathrm{A}_1}^{a_1} \cdots C_{\mathrm{A}_\alpha}^{a_\alpha}}.$$

This constant is highly nonlinear dependent on the concentrations of the different chemicals in the solution. But applying a logarithm on both sides and differentiating the expression leads to

$$\begin{aligned} 0 &= b_1 \frac{\Delta C_{\mathrm{B}_1}}{\bar{C}_{\mathrm{B}_1}} + \cdots + b_\beta \frac{\Delta C_{\mathrm{B}_\beta}}{\bar{C}_{\mathrm{B}_\beta}} \\ &\quad - a_1 \frac{\Delta C_{\mathrm{A}_1}}{\bar{C}_{\mathrm{A}_1}} + \cdots - a_\alpha \frac{\Delta C_{\mathrm{A}_\alpha}}{\bar{C}_{\mathrm{A}_\alpha}}, \end{aligned}$$

where $\Delta C_i / \bar{C}_i$ are deviations from nominal values, divided by those nominal values, meaning that it is *relative changes* that are of interest. For more information about this equations see H.-C. Pfisterer (2006) and H. Hyötyniemi (2006).

Even more the whole system can be represented in a linear way when using variables in a proper way. Nonlinearity can by avoided by appropriate preprocessing of the variables (by employing relative changes among the data). What is more, the dynamic nature can be put in the static form when the variables are selected in a clever way. The following points explain how to use information in order to stay in the linear domain even if the real system is nonlinear and can also be found in H. Hyötyniemi (2006):

- *Temperature:* According to the Arrhenius formula, reaction rates k are functions of the temperature, so that $k \propto \exp(c/T)$. When this is substituted, the model remains linear if one augments the data vector and defines an additional variable $\mathbf{z}_T = \Delta T / \bar{T}^2$.

- *Acidity:* The pH value of a solution is defined as $\mathrm{pH} = -\log c_{\mathrm{H}^+}$. Because this is a logarithm of a concentration variable, one can directly include the changes in the pH value among the variables as $\mathbf{z}_{\mathrm{pH}} = \Delta\mathrm{pH}$.

- *Voltage:* In electrochemical reactions one may characterize the "concentration of electrons". It turns out that according to the Butler-Volmer theory the amount of free electrons is exponentially proportional to the voltage. Hence, after taking logarithms, the "electron pressure" can be characterized by $\mathbf{z}_{\mathrm{e}^-} = \Delta U$.

- *Physical phenomena:* It is evident that phenomena that are originally linear like diffusion can directly be integrated in the model, assuming that appropriate variables (deviations from a nominal state) are included among the variables.

In practice, some reactions are not in balance; rather, there can be a constant flux of some chemical exiting from the reacting solution. The above balance expressions can be written also for such "dissipative" reactions, and linear models are again applicable around the equilibrium flux. This means that also the rates of change need to be included among the variables that characterize the dynamic state of the process.

Some of the variables to be included in the regression model are integrals over time — for example, in the coating process, the layer thickness is the integral of the nickel reduction rate. Because of the model linearity, such integrals can be transferred from the model output to the input — meaning that the layer thickness can be modeled as the integrals of the variables are included among the variable values themselves among the data.

Hence apart from the plain data also some features can be included in the dataset X. Since the measurement sample is only taken at the precise time when the plate is taken out of the bath it makes sense to include more information about the whole time the plate is immersed. A weighted integral over the plating time solves this problem, hence the dataset X is extended with integrals of the plain variables which doubles the dimension of X. The mentioned connection between layer thickness and nickel reduction rate is another reason for using integrals in the input structure. Furthermore an old set of data can be included by weighting the old data differently in comparison to the recent information. This smoothing of data again adds featured data to X.

A schematic view and a summary of all the mentioned variables in the input structure can be seen in Figure 4.

6 Modeling the plating process

When using the described setup the first step is PCR (Section 3). Figure 5 shows the eigenvalues λ_i of the correlation matrix of X in descending order. Since the dimension of the used dataspace is $n = 18$ there are as many eigenvalues, each one representing the

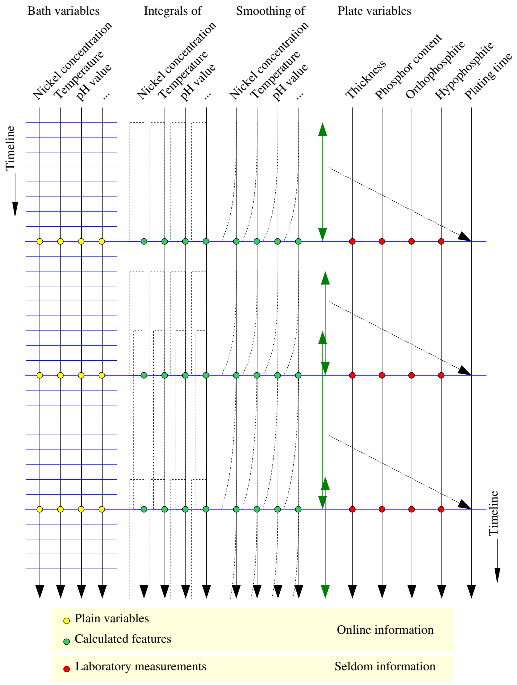


Figure 4: Schematic view of input data set along the timeline of the plating process

amount of information kept in the corresponding direction θ_i. It can be seen clearly that the last directions do not carry any information; that shows the redundancy in the data. Since the noise is assumed to be uncorrelated it is evenly distributed in all directions, hence some directions carry mostly noise and no real information. In the end it is reasonable to use only the first 7 directions for a new dataspace Z from which a final mapping to Y should be obtained. This first $N = 7$ vectors carry already 91% of the full information.

The alloy thickness and its phosphorus content can be accurately estimated with these 7 vectors. Using PLS as an data preparing algorithm the dataspace can even be reduced to a dimension of $N = 2$ and obtain an even better estimate. The results of the model gained by PLS analysis are presented here. For more results and discussions see H.-C. Pfisterer (2006).

Figure 6 shows the process data of the alloy thickness divided in three parts. The first one was used for model estimation along the above described way. The other two parts are validation sets. Blue information is the real data, measured in the laboratory and stored by a data acquisition system. Green data is a model estimation, obtained with a traditional complex electrochemical model by K. Kantola (2004) and used as a comparison to validate the linear model at hand. In red color the data estimation for the alloy thickness by the linear model can be seen and compared to real data and the other model estimation.

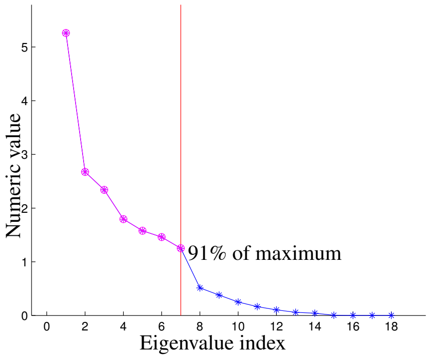


Figure 5: Latent variables θ_i (PCA) and the numerical values of the corresponding eigenvalues (equals importance of information among data)

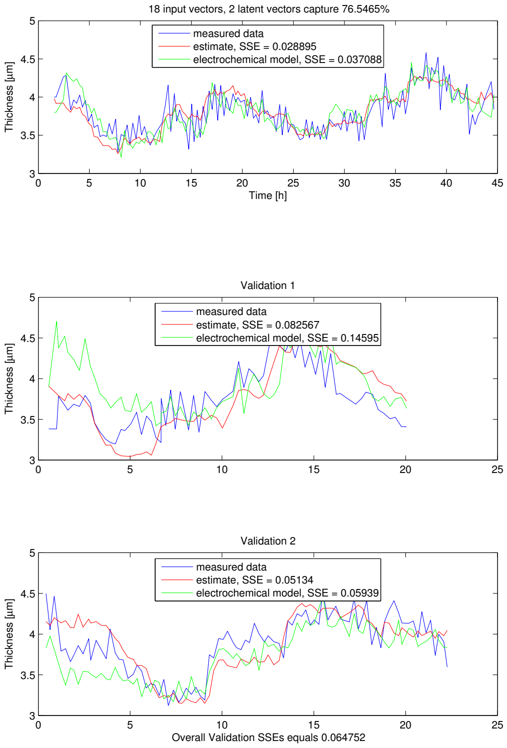


Figure 6: Alloy thickness: measured data and two compared estimates

The same setup of data was used to estimate the phosphorus content of the nickel alloy (Figure 7). In red color again the result of the linear model at

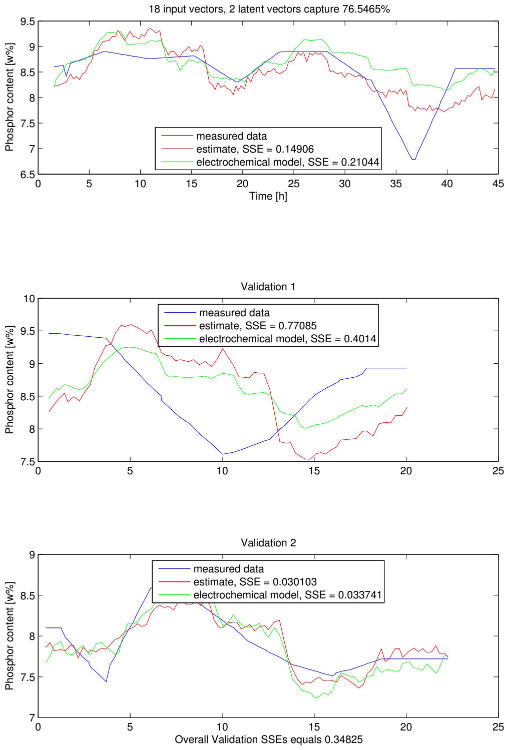


Figure 7: Alloy phosphorus content: measured data and two compared estimates

hand along with real data (blue) and results of another model (green).

The linear model provides a very accurate estimation of the alloy thickness and an accurate estimation of its phosphorus content. This is no surprise in the first parts of either dataset, since they were used to build the model and the model is tailored by the algorithm especially for them. But the correct estimates in the other parts confirm and validate the good performance of the linear model. The power and the correct combination of strong mathematical tools produced a very accurate result which can keep up and even beat complicated and inscrutable models. Since this model is a linear combination of measured data it is also possible to see the importance of each measured variable for the characteristic of each output variable.

It is interesting to see that if the estimate of the linear model is wrong also the estimate of the complex model is wrong. This is a hint for missing information that is not available for the modeling machinery in either case or some mistakes in the measurements, hence no fault of the linear model. Further it should be pointed out that there is an improvement when using PLS instead of PCR and also when including integrals in the input data vector. However, smoothed old data can not improve the results.

The overall result combined with the simplicity of its design leads a way to a good control machinery for nickel plating processes. The model can easily be used and adjusted by process workers and it can provide the necessary information for a controller in form of an observer for the plate characteristics. The information about these characteristics is already hidden in the anyway measured bath variables and it was just necessary to reveal it and make it emerge.

Acknowledgements

Many thanks to Prof. Dr.-Ing. Dr.h.c. Michael Zeitz from the Institute for System Dynamics (ISYS) of the University of Stuttgart who made the paper at hand possible and gave his support and advice.

References

H.-C. Pfisterer. Data-Based Modeling of Electroless Nickel Plating. Master's thesis, Helsinki University of Technology, Control Engineering Laboratory, 2006.

H. Hyötyniemi. Multivariate Regression. Technical report, Helsinki University of Technology, Control Engineering Laboratory, 2001.

H. Hyötyniemi. Neocybernetics in Biological Systems. Technical report, Helsinki University of Technology, Control Engineering Laboratory, 2006.

J. E. Jackson. *A User's Guide to Principal Components*. Wiley-Interscience, 2003.

J. Harju. Characterization of Electroless Nickel Plating Process. Master's thesis, Helsinki University of Technology, Control Engineering Laboratory, 2002.

K. Kantola. Modelling of Electroless Nickel Plating for Control Purposes. Master's thesis, Helsinki University of Technology, Control Engineering Laboratory, 2004.

R. Tenno, K. Kantola, H. Koivo, A. Tenno. Model Identification for Electroless Nickel Plating Through Holes Board Process. Technical report, Helsinki University of Technology, Control Engineering Laboratory, 1996.

Lähdekoodin symbolinen analysointi tekoälyn näkökulmasta

Erkki Laitila

Department of Mathematical Information Technology, University of Jyväskylä
Oy P.O. Box 35, FIN-40014 Jyväskylä, Finland

SwMaster Oy
Sääksmäentie 14, 40520 Jyväskylä

erkki.laitila@swmaster.fi

Tiivistelmä

Tämä artikkeli käsittelee lähdekoodin automaattisen käsittelyn menetelmiä ja tuo esille uutena konstruktiona symbolisen analyysin, joka on tunnettuja menetelmiä korkeatasoisempi, koska siinä tulee esille semioottinen, tulkitseva näkökulma. Esitämme sen tärkeimmät toiminnot ja piirteet tekoälyn näkökulmasta. Artikkelissa kuvataan analysointiprosessi ja sen hyödyntämiseen soveltuva symbolinen tiedonhakumenetelmä, jonka avulla käyttäjä voi saada symboliseen malliin perustuen tarvitsemaansa pragmaattista informaatiota tarkoituksena helpottaa nykyisen Java-kielisen ohjelmaversion ymmärtämistä haluttaessa kehittää uusi versio. Ratkaisu on ohjelmoitu ja koeajettu Visual Prolog-kehittimellä, joka hybridikielenä ja symbolista ohjelmointia tukien tarjoaa erinomaiset menetelmät vastaavan työkalun toteutukseen.

1 Johdanto

Tuskin mikään ohjelmistotoimitus valmistuu jo ensimmäisellä kerralla lopulliseen muotoonsa. Pikemminkin voidaan sanoa, että ohjelmistojen kehittäminen on lähes poikkeuksetta jatkuvaa tieto- ja informaatiovirtaa, siis prosessi, joka sisältää lukuisia vaiheita ja toimituksia. Kaikissa tilanteissa uudelleenkäytettävän koodin määrä tulisi, luotettavuus– ja kustannussyistä, olla mahdollisimman suuri. Lyhimmillään toimitussykli on päivittäistä koostamista sovellettaessa päivän luokkaa. Tietoliikennealalla toimitusten väli saattaa olla kireimmillään vain muutamia viikkoja, missä ajassa kaikki mahdollinen hyöty entisestä versioista olisi pystyttävä siirtämään uuteen versioon. Olemassa olevan koodin hyödyntäminen on siten merkittävä haaste. Sitä kuvataan seuraavassa metodologian näkökulmasta.

1.1 Takaisinmallintaminen

Takaisinmallintaminen (software reverse engineering) tarkoittaa olemassaolevan koodin hyödyntämistä. Erityisesti uudelleenmallintaminen (reengineering) tähtää saadun informaation käyttöön uudessa ohjelmaversiossa.

Selvästi todettu ongelma ohjelmistotyössä on, että ohjelmistoprosessi on luonteeltaan vain yksisuuntainen, sillä sen tuotoksesta, kuten työn laadusta, määrästä, monimutkaisuudesta ja lopullisen koodin käyttäytymisestä on ollut vaikeaa saada palautetta. Koska palaute on puuttunut, ei prosessia ole voitu optimoida samalla tapaa kuin miten teollisen tuotannon prosessit on aikanaan saatu hallintaan. Juuri ohjelmistoprosessien optimoitavuus on todettu korkeimmaksi alan kehityksen tasoksi, mutta vain aniharvat yritykset ovat päässeet korkeimmalle CMMI-tasolle (CMMI). Ilmeistä kuitenkin on, että jos ohjelmistoprosessista saataisiin yhtä hyvä ja ajanmukainen palaute kuin parhaimmissa tuotantoprosesseissa asianmukaisella mittaustekniikalla saadaan, myös ohjelmisto-organisaatiot saisivat merkittävän parannuksen toiminnan laatuunsa, millä olisi positiivisia vaikutuksia luotettavuuteen, läpimenoaikoihin, töiden delegoitavuuteen ja ennenkaikkea mahdollisuuteen saada uutta ydintietämystä jatkokehitykseen.

Tässä artikkelissa kuvattu symbolinen menetelmä tähtää juuri takaisinmallintamispalautteen parantamiseen.

1.2 Semioottinen näkökulma ohjelmistotyöhön

Semiotiikka tarkoittaa merkitysten ja kommunikaation tutkimusta. Se on tämä artikkelin kannalta kiinnostava aihe, koska ohjelmoijahan on aikanaan jättänyt koodiin jälkensä, joiden vaikutuksia ja tarkoituksia ylläpitäjä haluaa tutkia jatkokehityksen kannalta.

Peircen tulkintojen mukaan (1998) semiotiikka jakaantuu kolmeen pääkäsitteeseen, joita ovat ovat merkki (*sign*), kohde eli objekti ja tulkinta (*interpretant*). Merkki viittaa vastaavaan objektiin ja tulkinta yhdistää niitä toisiinsa. Esimerkiksi nähdessään jäniksen jäljet, ihminen tekee tulkinnan, joka yhdistää jäljet oletettuu jänikseen, jolloin jänis on objekti ja jäljet ovat vastaava merkki. Toisin kuin merkit, tulkinnat ovat aina sisäisiä esityksiä. Merkit voidaan jakaa osoittimiin (indice, index), ikoneihin ja symboleihin riippuen siitä minkälaisesta objektista on kysymys. Esimerkiksi kirja on tietämyksen osoitus (tunnusluku, indeksi), samoin savu on osoitus tulesta.

Vieläpä vaikka kausaalista yhteyttä tai mitään attribuutteja olioon ei olisikaan, silti voi löytyä jokin indikaatio (merkki) luomaan yhteyksiä objektiinsa. Symbolin käyttö soveltuu sellaiseen tarkoitukseen, ja juuri tällaista epäsuoraa kytkentää käyttää hyväkseen tässä artikkelissa kuvattu symbolinen menetelmä. Koska kaikki ohjelman sisäiset artifaktat ovat näkymättömiä, tulee analysointiohjelmiston muodostaa riittävän tarkka ja yleinen virtuaalimalli viittauksineen ja symboleineen käyttäjän näkösälle kuvaruudulle. Näytön koko on kuitenkin rajattu ja työkalun on siten mahdotonta tuottaa kerralla riittävää informaatiota yhtenä kokonaisuutena. Siksi analysoinnin tulee olla interaktiivista navigointia, joka mahdollistaa uusia tiedonhakuja eri näkökohdista.

1.2.1 Tietojärjestelmien semiotiikka

Morris (1971) jakaa semiotiikan kolmeen dimensioon, joita ovat syntaksi, semantiikka ja pragmaattinen tarkastelu, jolloin pragmaattinen osuus viittaa Peircen semioottiseen tulkintaan. Seuraavassa kuvataan, kuinka Frisco (1998) on määritellyt nämä kolme dimensiota tietojärjestelmien kehittämisen suhteen.

Frisco luokittelee informaatiojärjestelmän semioottiset tasot seuraavasti. Järjestelmän rakentaminen aloitetaan fyysisen maailman tasolta, joka sisältää valtaisan määrän signaaleja, merkityksiä ja luonnonlakeja. Järjestelmän suunnittelu tehdään empiirisen kokemuksen avulla luoden kokonaisuudesta malleja ja uusia käsityksiä. Tulos ohjelmoidaan, jolloin saadaan syntaksia, käytännössä kiinteä ohjelmisto tiedostoineen, kääntäjineen ja toimintaympäristöineen. Kun syntaksia ja koodia tulkitaan semantiikan kautta, saadaan tietoa koodin rakenteiden merkityksistä, väittämistä ja viittauksista ulkomaailmaan. Silloin kun järjestelmää tutkitaan nimenomaan eri roolien tarpeista lähtien, tavoitetellaan pragmaattista tietoa. Sillä on Friscon mukaan selvä yhteys järjestelmän tuottamaan hyötyyn. Kokonaisvaikutuksen tulee näkyä tietojärjestelmää ympäröivässä sosiaalisessa maailmassa, jonka tarpeisiin tietojärjestelmän tulisi tuottaa lisäarvoa.

Friscon määrittelykonseptia on kritisoitu esimerkiksi siitä, että se pyrkii formalisoimaan lähes epäformaaleja asioita, mutta takaisinmallintamisen kannalta tätä edellä kuvattua ketjua voidaan pitää kelvollisena lähtökohtana, koska takaisinmallintamisessa keskitytään nimenomaan formaalin koodin hyödyntämiseen.

1.3 Koodin analysoinnin teknologiat

Lähdekoodin analysoiminen liittyy ohjelmistojen takaisinmallintamisen (software reverse engineering) alueeseen, jonka tarkoituksena on juuri helpottaa uusien ohjelmaversioiden tuottamista. Koodin analysoinnissa periaatevaihtoehtoina on tutkia koodia käsin perinteisillä editoreilla ja kehitystyökaluilla tai käyttää parhaiten koodimassan käsittelyyn soveltuvia erikoismenetelmiä, joita on varsin vähän.

Kielioppien taustateoriat on tunnettu 50-luvulta asti ja koodin jäsentämisen metodologiat kehitettiin 70-luvulla Unixin mukana. Ylläpidon kannalta haitallista on, että analysointimenetelmät ovat kehittyneet kääntäjäteknologian ja koodin testauksen menetelmien sivutuotteina siten, että suurtakaan huomioita ei ole kiinnitetty juuri analysointituloksen hyödynnettävyyteen koko prosessin kannalta, informaatiojärjestelmän kokonaishyödystä puhumattakaan.

1.3.1 Abstract Syntax Tree (AST)

Vallitseva käytäntö koodin käsittelystä analysointiohjelmistoissa perustuu Abstract Syntax Tree-teknologiaan (AST). Siinä työkaluun tallentuu kieliopin mukaisia abstrakteja rakenteita, jotka muodostavat keskenään tiiviin hierarkian. Vaikka AST on hyödyllinen kääntäjiä suunniteltaessa, se ei kuitenkaan anna mahdollisuuksia syvälliseen tulkintaan, koska kielen semantiikka ei kokonaisuudessaan sisälly kielioppiin. Esimerkiksi muuttujaviittausten ja metodiviittausten logiikka ja oliosidosten toiminta sekä rakenteiden välinen semantiikka puuttuvat AST-määrittelystä.

1.3.2 UML ja round-trip-engineering

Käytännön tasolla oliopohjaisen koodin analysoinnissa on viime aikoina selvästi yleistynyt *round trip engineering* - niminen menetelmä, joka auttaa luokkarakenteiden poimimista koodista. Valitettavasti näin syntyvien UML-notaatioiden tarkkuus ei riitä pitkälle. Lisäksi eri työkalut saattavat tuottaa erilaisia kaavioita samalle lähdekoodille.

Toisaalta UML:n eri esitystavat eivät skaalaudu hyvin riittävän isojen näyttöjen luomiseen, minkä takia käyttäjä joutuu monesti tutkimaan lukuisia erilaisia kaavioita voidakseen tehdä johtopäätöksiä nykyisen koodin toiminnasta. Toistuva erillisten kaavioiden lukeminen on työlästä ja se kuormittaa käyttäjän lyhytkestoista muistia merkittävästi, jonka takia varsinainen ongelma saattaa jäädä ratkaisematta.

Olio-ohjelmoinnin kehittäjäjärjestö OMG keskittyy pääasiallisesti uusien sovellusten kehittämiseen ja sitä kautta uuden koodin luomiseen. Ehkä sen takia heidän mallinsa eivät tarjoa tarpeeksi tarkkoja esityksiä, jotka mahdollistaisivat koodin tarkimpien lauseiden tallentamisen tai suorittamisen, mikä olisi tarkan analyysin perusedellytys. Siten käsitteellä executable UML ei tarkoiteta vastaavan koodin suorittamista vaan lähinnä vastaavan mallin tietojen evaluointia nimenomaisen mallin kannalta, ei koodin kannalta.

1.3.3 Koodin ymmärtämisen peruskäsitteet

Kuten edellä todettiin, takaisinmallintamisen luoma palaute on arvokasta kehittäjäorganisaatiolleen, ja että syntaksi tarjoaa erinomaisen pohjan uudelle automaatiolle, koska käsiteltävä koodin informaatio on luonteeltaan formaalia. Esille tulee kuitenkin kysymys, mikä osa koodista on merkityksellistä ja arvokasta? Vas onko koodi vain tasapaksua massaa, jonka lukeminen siltikin onnistuu parhaiten editorilla?

Ohjelman ymmärtämisen teorioita on kehitetty 80-luvulta lähtien, jolloin Pennigton (1987) kehitti tarkastelun alhaaltapäin alkavan tarkastelun viisikkomallinsa (function, control flow, data flow, state & operation) ja Brooks (1983) kehitti yleiset teoriat tarkastelun hypoteeseille. 90-luvulla von Mayrhauser (1992) loi integroidun metamallin, johon liitettiin osana ylhäältäpäin alkava osittava tarkastelutapa sekä tilannemalli, joka on kumulatiivista ohjelmasta saatua informaatiota toimialansa tietoon yhdistettynä.

Wiedenbeck (2002) on edelleen laajentanut aiempia teorioita 2000-luvulla kattamaan olio-ohjelmien ymmärtämisen tarpeet. On esitetty myös abstraktimpia käsitteitä kuten suunnannäyttäjä (beacon) ja lohko (chunk) kuvaamaan koodin ymmärtämistä esimerkiksi lajittelualgoritmien osalta, mutta niiden merkitys oliomaisen koodin tarkastelussa ei ole enää kovin merkittävä, koska oliot ja metodit rajaavat tehokkaasti toimintoja sisäänsä muodostaen uuden helpottavan abstraktiotason.

Sen sijaan uutena asiana olio-ohjelmointi on tuonut suunnittelumallit ja kerrosarkkitehtuurit uusiksi tarkastelutavoiksi. Niiden jäljittäminen koodista on paljon tutkittu alue ja koodin rakenteen uudelleen suunnittelu (*refactoring*) on siihen soveltuva konkreettinen käyttökohde.

1.4 Tekoälyn ja koodin analysoinnin suhde

Tekoälyyn liittyvät keskeisesti tietämyksen hankinta, palautus ja käsittely, ihmisen ja koneen välinen vuorovaikutus, päättelymekanismit ja älykkäät käyttöliittymät. Lähdekoodin tutkimuksen alueelta voidaan siihen liittyen määritellä, että ohjelman ja sen toiminnan ymmärtäminen on oppimisprosessi, jossa lukija pyrkii kehittämään tietämystään tietojärjestelmän nykyarvon kasvattamiseksi. Ohjelma sisältää paljon sumeaa tietoa, jota ohjelmoijat ovat tallentaneet koodiin omalta kokemustaustaltaan optimoiden. Sen sisältö on usein osittain virheellistä. Tietoa on jopa miljoonia rivejä, joten sitä on mahdotonta esittää näytössä kerrallaan, tarvitaan siis erilaisia haku- ja suodatusmekanismeja.

Parhaille formaalien kielten analysoinnin sovelluksille on ominaista, että koodin tieto saadaan poimittua tarkkaan semantiikan mukaan. Penningtonin viisikkomalli muodostaa tämän tiedon perustan. Se määrittää mikä on kiinnostavaa dataa. Käyttäjä on liikkeellä yleensä joko perehtymistarkoituksessa tai paikantamistarkoituksessa. Molemmissa tapauksissa hän haluaa evaluoida kriittisimpiä koodiosuuksia apukysymysten ja hypoteesien kautta. Yhteinen nimittäjä kaikille näille tarpeille on koodin ymmärtämisen tehostaminen työkalun avulla (program comprehension, PC).

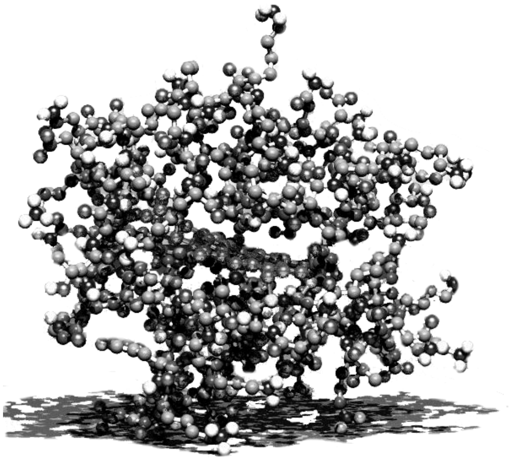
1.5 Tutkimuksen rajaus

Erilaisia ohjelmointikieliä tunnetaan suuri joukko. Painopiste teollisuudessa on siirtynyt oliopohjaisiin kieliin, mutta silti alan tutkimus huomattavan suurelta osin käsittelee proseduraalisia kieliä kuten C, Pascal tai Cobol. Oliopohjaisista pääkielistä C++ on laajin, monimutkaisin ja ehkä jopa tärkein, mutta sen analysointiin on varsin rajatusti menetelmiä. Sen sijaan Java-kieltä ja sen käsittelyä on tutkittu paljon, koska se on kielenä selväpiirteinen.

Tässä artikkelissa käsittelemme etupäässä Javaa, koska sen selväpiirteisen kieliopin ansiosta on mahdollista päästä nopeasti hyvin tarkkaan käsittelytulokseen. Lisäksi edistysaskeleiden osoittaminen paljon tutkitulle Javalle voi olla selvä osoitus uuden tutkimuslähestymisen paremmuudesta perinteisempiin tutkimuslähestymisiin nähden.

Symbolinen analyysi on SwMaster Oyn toimesta käynnistynyt innovaatio, jota Erkki Laitila tutkii väitöstutkimuksessaan Jyväskylän Yliopistossa. Sen tekninen osuus keskittyy koodin symbolien ja niihin liittyvän logiikan kuvaamiseen mahdollisimman hyvin Visual Prolog- kehittimellä, joka on hybridikieli sisältäen erinomaisen oliojärjestelmän mallintamistarkoituksiin ja logiikkaohjelmoinnin perustan koodin logiikan, assosiaatioiden ja riippuuvuksien käsittelyyn.

Symbolinen koodin malli on uusi atomistinen konstruktio, joka mahdollistaa koodin tutkimisen kaikista lähtökohdista, jotka liittyvät Java-koodin semantiikkaan ja simuloidun koodin käyttäytymiseen ja ajovaikutuksiin. Kuva 1 (seuraava sivu) esittää molekyyläärista rakennetta sidoksineen. Se muistuttaa ohjelmakoodia, koska molemmissa on sidoksia. Tärkein asia ohjelmistoja hahmotettaessa on ymmärtää kuinka eri lähestymistavat kuten käyttötapaukset, ohjausvuo ja ohjelmasuunnitelmat liittyvät toisiinsa. Olio-ohjelmointi aiheuttaa omia periaatteellisia ongelmiaan, koska vastaavat kutsut sijoittuvat näkymättömiin eri metodien sisälle koodiin. Syntyviä olioita ei voida myöskään tarkkailla millään analysoinnin perusmenetelmillä. Siten näin havainnollista kuvaa ei tänä päivänä ole saatavissa, mutta mahdollisuuksia on, mikäli tässä tutkimuksessa kuvattu teknologia pääsee aikanaan käytäntöön asti.



Kuva 1. Symbolinen, atomistinen malli.

Voimakkaasti yksinkertaistaen voidaan sanoa, että koodin ymmärtäminen on siinä olevien symbolien ja rakenteiden välisten suhteiden ymmärtämistä. Nämä suhteet muodostavat keskenään symbolivuon, joka tuottaa hyödyllisintä informaatiota silloin kun se on jäsentyneenä ohjelman suoritusjärjestykseen. Siten kuvassa 1 ylinnä voisi olla *main*-metodi, joka haarautuisi alaspäin suorituslogiikan mukaisesti.

2 Analysointimenetelmät

Perinteisiä koodin analysointimenetelmiä ovat staattinen ja dynaaminen analyysi. Staattinen analyysi ei sovellu hyvin oliomaisille kielille kuten Javalle, koska vain osa toiminnoista on tulkittavissa suoraan koodista. Toisaalta dynaaminen analyysi vaatii lähes aina monimutkaisia ajojärjestelyjä.

Staattisella analyysillä voidaan tuottaa riippuvuuskaavioita ja kutsupuita, jotka toimivatkin hyvin proseduraalisille kielille jopa niin, että niiden avulla voidaan saada täsmällinen tieto mitä koodinosia mikäkin proseduuri käyttää ja missä järjestyksessä ohjelmavuo etenee, mikä on arvokasta tietoa mm. vianpaikannustilanteissa. Vaikeita haasteita staattiselle analyysille ovat osoittimet (pointer) ja pysähtymättömyyden analysointi.

Dynaaminen analyysi perustuu pitkälle ohjelman ajamiseen debuggerilla ja sitä kautta koodin toimintojen jäljittämiseen valittujen pysäytysehtojen, testitapausten ja jäljityskomentojen mukaisesti. Näin saadaan talteen jälki, trace, joka voi sisältää erittäin paljon aivan turhaakin tietoa. Jopa 98 % jäljestä voi olla epäkuranttia.

Edellä mainittujen menetelmien rajoitteet tuntien on aiheellista kysyä, voidaanko muodostaa sellainen uusi analysointitapa, jonka painopisteenä olisi koodi-informaation jalostaminen ja muokkaaminen sellaiseen muotoon, että syntyvästä ratkaisusta voitaisiin selektiivisesti kyselyin poimia haluttu informaatio? Voisiko syntynyt menetelmä esimerkiksi toimia kuten tietokanta, jolloin käyttäjä saisi vapaasti valita minkä asioiden välisiä suhteita hän haluaisi tarkastella suunnitellessaan muutoksiaan?

Mietittäessä vastauksia edellisiin kysymyksiin päädytään ideaalisen analyysin määritelmään: Millainen olisi ideaalinen analysointimenetelmä, jos sen voisi vapaasti määritellä puhtaalta pöydältä?

Vertauskohtia voimme tietysti löytää muiden tietojärjestelmien ja teknologioiden parhaista toteutuksista, joita ovat mm. relaatiomalli, informaatioteoria, diagnostiikan perustietous, asiantuntijajärjestelmien perustyypit. Näiden kautta voimme päätyä synteesinomaisesti seuraaviin vaatimuksiin:

- Analyysi voidaan tehdä milloin vain mistä tahansa koodin osasta.
- Analyyysi poimii käsittelyyn kaikki tarvittavat asiat kuten Penningtonin ja Wiedenbeckin määrittelemät tiedot.
- Tuotoksella tulee olla ideaalinen tiedonsiirtosuhde ilman turhaa tietoa.
- Vastaava analysointiprosessi tukee tiedonhankintaa, mahdollistaen uuden tietämyksen hankinnan.
- Vastaus staattiseen kysymykseen saadaan nopeasti. Silti ratkaisu vaativaan ongelmaan saisi kestää kauemminkin, jos vertailukohtana käytetään käsin tehtävää perushakua, mikäli haku vaatii runsaasti syvällistä laskentaa..
- Menetelmän tulee tukea käyttäjän pohdintaa syntaksin ja semantiikan suhteista, mieluummin myös pragmaattiselta kannalta katsottuna.
- Menetelmän tulisi voida tunnistaa arkkitehtuurin kaikki keskeiset piirteet.

Ohjelmakoodi sisältää erilaisia tasoja, kerroksia ja menetelmiä, joten käytännön tutkimuksessa on tehtävä rajauksia. Tässä tutkimuksessa keskitytään

a) ”kovoteoriaan”, joka tuottaa mahdollisimman ytimekkään, mutta ilmaisuvoimaisen mallin
b) ”pehmoteoriaan”, joka määrittelee ihmisen tyypillisen käyttäytymismallin ylläpito-ongelmia ratkottaessa.

Siten abstraktimmat käsitteet, jotka ovat etäällä kieliopin määrittelyistä, jätetään huomioimatta. Esimerkiksi kerrosrakenteen poimimista koodista ei tutkita eikä myöskään suunnittelumallien tunnistamista.

3 Koodin käsitehierarkia

Koodin tutkiminen muistuttaa monin tavoin tutkimusta, koska sen lukeminen vailla tarkoitusta ei ole mielekästä. Aina tarvitaan jokin tavoite, jolloin lukija joutuu miettimään tavoitteen ja koodin välistä suhdetta (von Mayrhauser 1992). Se onnistuu vain olettamuksia eli hypoteesejä tehden. Näin koodin tarkasteluun voidaan soveltaa tieteen peruskäsitteistöjä alla olevaan tapaan.

Ontologia kuvaa koodin olemuksen ja sen keskeisen rakenteen. Epistemologia tunnistaa koodista olennaista tietoa pyrkien määrittelemään sen tietämystä. Metodologia kuvaa tavan, jolla selektiivinen tiedonkeruu ja tietämyksen johtaminen siitä voi tapahtua. Lisäksi tarvitaan vielä menetelmä, joka soveltuu arkielämän tarpeisiin ja erityisesti työkalu, joka helpottaa käyttäjän toimia.

3.1 Peruskäsitteet

Seuraavassa esitetään lyhyesti tutkimuksen keskeiset termit, jotka selostetaan myöhemmin tarkemmin.

3.1.1 Sanasto ja lyhenteet

G	kieliopin symboli
X	koodin syntaksin rakenne kieliopissa G
N	rakenteen X tyyppi (non-terminal)
Y	X:n vaikutus koodissa, semantiikka
Z	X:n merkitys lukijalle, pragmatiikka
B	X:n käyttäytymismalli (input/output-tarkastelu)
S	koodista saatu symboli
O	symbolista S luotu olio
L	logiikka, joka olioon O sisältyy (interpretant)
M	malli, joka koostuu olioista
A	analyysi, joka hyödyntää mallia M
Q	kysely, jolla analyysi A käynnistyy
H	hypoteesi, jonka perusteella kysely Q voidaan tehdä
P	päättelyprosessi, josta käsin luodaan hypoteeseja H
T	ylläpitotehtävä päättelyprosessin P aktivoimiseen
I	pragmaattinen informaatio ylläpitoa tukemaan
K	tietämys (knowledge), jota prosessissa P kehitetään

Symbolinen analyysimenetelmä toimii ideaalisesti, jos se pystyy yhdistämään edellä mainitut käsitteet ideaalisesti.

3.1.2 Määritelmiä

Informaatio on uutta, käyttäjäkohtaista merkittävää tietoa, jonka määrä on minimoitu tarpeeseensa.
Tietämys on sellaista tietoa, jota voidaan käyttää ratkaistaessa muita samantyyppisiä ongelmia.

Tietämyksen hankintaan liittyy siten oppimista ja sisäistämistä. Arvokkain tieto, jota metodologia voi tuottaa, on juuri tietämystä, taitoa ratkaista tulevia ongelmia yhä järjestyneemmin verrattuna tapauskohtaiseen käsittelyyn, jossa ratkaiseminen tapahtuu yhä uudelleen entiseen tapaan. Uuden tietämyksen avulla nykyistä menetelmää voidaan kehittää edelleen, jolloin päästään positiiviseen kierteeseen ja optimoimaan menetelmän parametreja sekä eliminoimaan virhemahdollisuuksia. Walenstein (1998) esittää, että näin tietämys muuttuu säännöiksi ja taidoiksi.

3.1.3 Ohjelmaesimerkki

Seuraavassa esitettyä lyhyttä esimerkkiä käytetään jatkossa havainnollistamaan symbolisen analyysin ideaa. Siinä oleva luokka Server käynnistää muutamaa lausetta käyttäen (4-10) Server-olion.

```
   // Start the server up, listening on an
   // optionally specified port
1   class Server extends Thread {
2     {
3     protected int port;

4a     public static void main(String[]
4b            args) {
5         int port = 0;
6         if (args.length == 1) {
7a             try { port =
7b             Integer.parseInt(args[0]); }
8a             catch (NumberFormatException
8b          e) { port = 0; }
9         }
10        new Server(port);
11    }
12  }
```

Javan koodi on selkeä. Se sisältää vain varattuja sanoja ja joko viittauksia Javan kirjastoon tai käyttäjän antamia symboleita viittauksineen. Varatut sanat rajaavat rakenteita (*nonterminal*), esimerkkeinä luokka Server ja sen sisältämä metodi main.

Javaa tuntematon lukija ei voi päätellä esimerkkikoodin (X) aiheuttamasta toiminnasta (B) paljoakaan, koska hän ei tunne kielen semantiikkaa (Y), riippuvuussääntöjä (N) eikä käsitteitä (G). Se sijaan kokenut Java-ohjelmoija ymmärtää yllä olevan toiminnan täydellisesti (Z) luettuaan jokaisen rivin huolellisesti.

3.2 Koodin ontologia

Formaalien kielten jäsentäminen ei ole ongelmallista, koska kielen määrittely on alkuvaiheissa suunniteltu säännönmukaiseksi. Seuraavassa kuvataan kielten tulkinnan mahdollisuuksia tietosisällöistä lähtien.

3.3 Kieliopin määritelmä

Kieliopin määritelmät ovat peräisin jo puolen vuosisadan takaa (Chomsky 1956). Seuraavassa formaalin kielen kieliopin määritelmä:

$$G = \langle N, \sum, R, S0 \rangle$$

Symboli N edellä tarkoittaa jakokelpoista termiä (non-terminal), jolla on vähintään kaksitasoinen hierarkinen rakenne. ∑ on kielen kaikkien varattujen sanojen luettelo sisältäen välimerkit ja avainsanat (terminal). R on joukko sääntöjä (production rule), jotka kertovat kaikki sallitut rakenteet, joita kukin N voi saada. S0 on kielen käännöksen aloitussymboli, jolla kuitenkaan ei ole analysoinnissa juuri merkitystä.

Tärkeimpiä rakenteita Javan osalta ovat luokan määrittely *ClassDeclaration*, metodi *MethodDeclaration*, lause *Statement*, lauseke *Expression* ja muuttuja *Identifier*.

Ontologian ja kapseloinnin takia luokan määrittely on ensiarvoisen tärkeä. Se on erittäin selkeä. Luokka sisältää määrittelynsä mukaan luokan jäseniä (metodi ja attribuutti), ja sillä on perintämäärittely. Yhdessä nämä ulkoiset määrittelyt muodostavat luokan allekirjoituksen, luokkasopimuksen.

Javan sanastossa erityisesti jokainen viittaus käsitteeseen *Identifier* viittaa joko JDK-kirjastoon tai käyttäjän antamaan symboliin, joka voi olla joko paketin, luokan, metodin, attribuutin tai muuttujan nimi. Kaikki muut kielen rakenteet tulee tunnistaa suhteessa *Identifier*-viittauksiin.

3.3.2 AST:n rajoitukset analysoinnissa

Yleinen tapa käsitellä kielen rakenteita on tallentaa ne AST-muotoon. Siihen liittyy kuitenkin paljonkin ongelmia analysointitarkoituksessa. AST-rakenteilla ei nimittäin ole yksiselitteistä viittaussemantiikkaa, jonka mukaan voitaisiin poimia itsenäisesti evaluoitavaksi yksilöllisiä koodin rakenteita esiin kuten if-lause metodista main. Siten sisempiä AST-rakenteita ei ole mahdollista käsitellä ja evaluoida itsenäisesti vaan vain osana laajempaa kokonaisuutta. Siksi tarvitaan kehittyneempi menetelmä, jota kuvaamme seuraavassa.

3.3.2 Kielioppien saatavuus ja ongelmat

Kielioppeja eri kielille ja myös Javalle on saatavissa internetistä. Ongelmana on kuitenkin kielioppitiedostojen laatu, erilaiset kirjoitusvariaatiot ja niiden taltiointitarkkuus yleisesti, koska osa niistä on pelkästään periaatteellisia ja vain osa riittävän tarkkoja, että ne mahdollistaisivat sellaisenaan parserin rakentamisen.

Esimerkiksi Sunin julkaisema version 3 (Java1.5) kielioppi, joka vastaa uusimman kääntäjän tilannetta ja julkaistaan myös Java 3rd Edition-kirjan mukana, on ongelmallinen. Siinä on useita virheitä ja lisäksi moniosaisten termien semantiikkaa ei ole kuvattu tarkoituksenmukaisesti. Hankalinta on, että Sunin ilmoittama kielioppi ei esitä tarkkaan lausekerakenteita, koska siinä kaikilla laskentatoiminnoilla yhteenlasku, kertolasku, jakolasku ja loogiset vertailut olisi virheellisesti sama prioriteetti ja siten sama laskentajärjestys.

Siksi jotta riittävään tarkkuuteen päästäisiin, tulee Sunin toimittama kielioppi unohtaa kokonaan ja organisoida kielen rakenteet itse uudelleen ja muuntaa ne ristiriidattomaan, hierarkisempaan muotoon.

Prolog-kielelle on kehitetty edistyneitä kieliopin käsittelyn menetelmiä kuten definite clause grammar (DCG), jossa yksittäisen lausekkeen prioriteetti voidaan ilmaista lukuna. Tämä menetelmä soveltuu tulkkaavalle ja ISO-standardin mukaiselle Prologille. Tutkimuksen mukainen Visual Prolog-kehitin sisältää oman menetelmän, jota kutsutaan semanttiseksi kielioppirakenteeksi, minkä on alkujaan kehittänyt Leo Jensen, Prolog Development Center. Alla if-lauseen kuvaus tällä menetelmällä:

```
1 Statement =
2  “if” Condition “then” Statement ->
3    iff(Condition, Statement).
```

Termin nimenä on Statement, jonka yksi mahdollinen muoto olisi edellä mainittu if-lause, jonka syntaksi sisältää hierakisesti alemmalla tasolla sijaitsevat termit Condition ja Statement. Varattuja sanoja siinä ovat lainausmerkeissä olevat sanat if ja then. Syntaksin oikea järjestys ilmenee ”->” -symbolia edeltävästä osuudesta. Sen oikealla puolella oleva rakenne *iff* argumentteineen on muotoa, joka syntyy jäsennyksen tuloksena vastaavan työkalun muistiin. *Iff*-rakenne kuvaa siis semantiikkaa, mistä käytämme tässä artikkelissa nimitystä Y. Näin saatiin ytimekäs kokonaisuus: rivi 1 kertoo määrittelyn nimen (N), rivi kaksi kertoo syntaksin (X) ja rivi kolme semantiikan (Y).

Jos käytännön tilanteessa asianomainen if-lause ei johda halutulla tavalla sen sisältämän osuuden käynnistymiseen, vika voi olla vain sen Condition-osassa. Tällöin tämä ehto olisi vianpaikantajalle pragmaattista tietoa (Z).

3.3 Koodin mallien epistemologia

Epistemologia pyrkii erottamaan todellisen ja riittävän tietämyksen paikkansa pitämättömästä tietämyksestä. Se tutkii mm. tiedon luonnetta, tietojen välisiä yhteyksiä. OMG ei erikseen puhu epistemologiasta tutkiessaan metodologioita ohjelmistokehityksen näkökulmasta, sehän ei edusta tiedettä vaan käytännön päämääriä. Kuitenkin koodin ja mallien synkronointi on OMG:n päätavoitteita. OMG:n painopisteenä on mallintamiskokonaisuus nimeltään Model Driven Architecture, MDA (MDA 2006), jossa keskeisimpänä tietosisältönä on MOF-määrittely (MOF 2006). MOF:ssa koodintarkan informaation käsittely osoittautuu mahdottomaksi, koska Javan lause-tason informaatio puuttuu siitä (tosin siinä on OCL-rajoitekieli).

MOF tarjoaa koodista poimitulle informaatiolle seuraavat mallit ja tasot. Taso M0 eli systeemitaso kuvaa koodia sellaisenaan. Taso M1 eli luokkataso viittaa koodista poimittuun luokkamäärittelyjen rajapintaan. Taso M2 viittaa metamalliin, jolla määritellään luokkien yleiset rakenteet. Ylinnä on taso M3 metametamalli, joka määrittelee yleisen transformaatiomallin lähinnä muunnostarkoituksiin.

Koska MOF ei tue lausetarkkaa käsittelyä, sen tarjoama ymmärtämistuki jää suppeaksi. Niinpä alimmalle tarkastelutasolle tarjotaan vaihtoehtona AST-rakenteiden avulla muodostettua mallia, mutta siinähän on muutamia perusongelmia, joita on kuvattu tämän artikkelin alkuosassa.

Seuraavassa jaksossa koodista poimittavissa olevaa tietoa käsitellään erillisten teorioiden valossa tarkoituksena muodostaa looginen ketju sellaisen eheän mallin luomiseksi, joka sisältäisi sekä MOF:n kaltaisen metariippuvuusinformaation että koodin tarkan lauserakenneinformaation, jotta tulosta voitaisiin käyttää yhtenäiseen tarkasteluun.

3.3.1 T1 Koodin jäsentämisen teoria

Koodista saadaan jäsennyksen tuloksena rakenteet (N), jotka parserin jäljiltä ovat yhdessä hierarkisessa rakenteessa, jonka juurena on aloitussymboli (S0), esimerkiksi *TranslationUnit*, joka jakaantuu paketin ja luokan määrittelyihin.

Prologilla jäsennetty parsintapuu (hierarkinen kokonaisuus) on predikaattilogiikan (1. kertaluvun logiikka) mukaisesti täydellisesti aksiomatisoitavissa. Parsinta on täydellinen ja todistettavissa, jos kukin rakenne säilyttää alkuperäisen muodon ja riittävän tarkan informaation (koherenttius ja konsistenssius).

3.3.2 T2 Abstraktiotason nosto, symbolinen kieli

Laskentasääntöineen ja sisäkkäisine lauseineen Javasta saadut rakenteet ovat tarpeettoman monimutkaisia sellaisenaan analysoitaviksi. Kun rakenne on kerran jäsennetty oikein, sitä voidaan merkittävästi (noin 90 %) yksinkertaistaa pudottamalla sisäkkäisiä itsestään selviä välitasoja ilman, että uusi sisältö vääristyy tai muuttuu epäloogiseksi.

Jatkoanalysoinnin kannalta tärkeää on ryhmitellä rakenteet mahdollisimman hyvin. Analysointitarpeita ovat mm. seuraavien rakenteiden tunnistaminen, suluissa jatkossa käytetty lyhenne: määrittelylauseet (*def*), konstruktorikomennot (*creator*), dataa muuttavat rakenteet (*set*), dataan viittaavat rakenteet (*ref*), metodiviittaukset (*invoke*, *get*), operaatiot (*op*), silmukkalauseet (*loop*), ehdolliset lauseet (*path*, *condition*), vakiot (*value*) ja muut lauseet. Näin saadaan 11 lauseryhmää.

Tällä ryhmittelyllä päästään seuraavaan tavoitteeseen. Eri analyysit voidaan määritellä abstraktien rakenteiden välisinä yhteyksinä, esimerkiksi data flow – analyysi on saman datan set- ja ref-rakenteiden välinen keruutoiminto. Ohjausvuo on valitusta koodinosasta koottu get- ja creator-lauseiden kokonaisuus, joka käy läpi muiden lauseiden tarvittavat alarakenteet.

Ryhmittelylogiikasta voidaan muodostaa uusi symbolinen kieli, jolle tutkimuksessa on annettu nimi Symbolic. Kuten jokaisella formaalilla kielellä, myös Symbolicilla voi olla oma syntaksi ja semantiikka. Näin ollen sille voidaan luoda jäsennin ja koodigeneraattori, joista on hyötyä muunnettaessa abstrahoituja rakenteita lukijan paremmin ymmärtämään muotoon tai tarvittaessa luoda sujuva testausympäristö analysointikokoonpanolle alkuarvojen syöttömahdollisuuksineen.

Jos Java-kielen keskeinen lause on nimeltään *Statement*, voisii sitä vastaava korkeamman tason rakenne Symbolic-kielessä olla *Clause*. Näin saadaan translaatiotarve *Statement2Clause*, jota kuvataankin seuraavassa.

3.3.3 T3 Translaatio

Formaalien kielten translaatioon on kehitetty erillisiä monimutkaisia menetelmiä, jotka perustuvat useinmin ulkoiseen translaatiomekanismiin (*translation engine*). Mielenkiintoista on, että translaatio voidaan tehdä myös ”suoraan” ilman erillistä sääntölogiikkaa rekursiivisen sijoittamisen avulla. Prolog tukee tällaista vaihtoehtoa tunnistaessaan automaattisesti kielen rakenteet. Tällöin aivan tavallinen sijoituslause käy translaatioksi.

Seuraava rekursiivinen *xslate*-komento muuttaa Java-kielen if-lauseen Symbolisen kielen path-lauseeksi:

```
xlate(iff(Condition,Statement)) =
  path(xlate(Condition),xlate(Statement)).
```

Olennaista ylläolevassa komennossa on se, että vasemman puolen suluissa on muunnettava Java-termi ja yhtäsuuruus-merkin jälkeen Symbolic-kieleen syntyvä rakenne. Koska jokaisen alarakenteen yhteydessä kutsutaan alas laskeutuen uudelleen translaatiokomentoa, kaikki alarakenteet muuntuvat automaattisesti uuteen muotoon. Translaation suorittamiseksi jokaiselle erilaiselle Javan termille tarvitaan yleensä vain yksi edellämainitun kaltainen sijoituslause. On vain muutama poikkeus.

Siirto Javasta symboliseen kieleen (*Statement2Clause*) voidaan validoida analyyttisesti translaatiolausekkeiden kautta sekä testitulosteilla. Lisäksi Visual Prolog sisältää tyypintarkistustoiminnon, joka tarvittaessa varmistaa, että kukin alalauseke on kattavasti määritelty. Se poistaa oikein käytettynä puutteet muunnoksen kattavuudessa.

3.3.4 T4 Semantiikan talteenotto

Ohjelmointikielten semantiikan ilmaisemiseen on useita esitystapoja eli notaatioita. Merkittävimpiä ovat toiminnallinen semantiikka (operational semantics). Eräs sen muoto, luonnollinen semantiikka (natural semantics), on Prologin kannalta merkittävä, koska se on Prolog-yhteensopiva. Siten jokainen luonnollisen semantiikan lause voidaan periaatteessa ohjelmoida Prolog-lauseiksi. Myös Prolog-lauseita voidaan tietyin edellytyksin muuntaa luonnollisen semantiikan esitystapaan, jolloin sitä voidaan käyttää kuvaamaan Prolog-koodin käyttäytymistä alkuperäistä korkeammalla tasolla.

Luonnollinen semantiikka tarjoaa seuraavia etuja perinteisempiin semantiikkoihin nähden:

- Kaikki sen oliot ovat äärellisiä termejä, joten rakenteen käsittelyyn ei tarvita monimutkaista tyyppiteoriaa.
- Koska jokaiselle rakenteelle voidaan määritellä useita päättelysääntöjä, sillä voidaan mallintaa myös epädeterminististä hakua, joka on Prologin parhaita ominaisuuksia.
- Epädeterminisitinen määrittely mahdollistaa polymorphististen rakenteiden mallintamisen.

Luonnolisen kielen semantiikan yleinen lausemuoto on seuraava:

$$\frac{H_1 \vdash T_1 : R_1 \ \ .\,.\ \ H_n \vdash T_n : R_n}{H \vdash T : R} \quad \text{if } \langle\text{cond}\rangle$$

Kuva 2. Luonnollisen semantiikan periaate.

Kuvassa 2 alarivillä on todistettava lauseke, joka sisältää hypoteesin H, johon liittyy parametrikokonaisuus T ja tavoitetulos R. Se ratkeaa, jos ylemmällä rivillä oleva hypoteesi H1 argumentein T1 ratkeaa tuottaen tuloksen R1, joka edelleen mahdollistaa muiden hypoteesien H2 … Hn toteutumisen. Lopullinen tulosjoukko R on kombinaatio ehdollisten hypoteesien tuottamasta tulosjoukosta. Tämä lauseke toteutuu vain jos ehto *cond* toteutuu myös.

Prolog sallii predikaatteineen tällaisen lauseen kirjoittamisen. Visual Prolog erityisesti sisältää oliojärjestelmän, joka myös soveltuu sekä ylläolevien hypoteesien ohjelmointiin että koodin muuttamiseen vastaavanlaisiksi hypoteesimäärittelyiksi. Alla pieni esimerkki, jossa hypoteesit on muutettu predikaateiksi:

```
h(T, Cond) = R:-
    ifTrue(Cond),
    R1 = h1(T1),    … Rn=hn(Tn),
    Result = [R1,R2,… Rn].
```

Predikaatti ifTrue tarkistaa aluksi onko ehto Cond voimassa. Jos on, siirrytään h-predikaattien evaluointiin.

Seuraavassa esimerkissä hypoteesit ovat oliota (tässä tutkimuksessa symbolisen mallin elementtejä):

```
evaluate(T, Cond) = R:-
  ifTrue(Cond),
  R1 = H1:evaluate(T1),
  …
  Rn = Hn:evaluate(Tn),
  R = [R1, … Rn].
```

Esimerkin selostus: Visual Prologissa suorat dataviittaukset voivat kohdistua vain ko. olion sisälle. Se estää haitalliset sivuvaikutukset. Jokaisella oliolla tulee olla metodi *evaluate*, joka laskee oman tuloksensa (R_i) rekursiivisesti.

Ohjelmointikielten semantiikassa keskeisiä asioita ovat seuraavat asiat:

- Dataviittausten huomiointi
- Metodikutsujen huomiointi parametreineen
- Polymorphististen metodien huomiointi
- Perintähierarkian huomiointi

Kirjallisuutta semantiikan rajoitteista ja erityisesti Java-koodin semantiikasta on runsaasti. Javan spesifikaatio sisältää täsmälliset määrittelyt kielensä piirteistä edellä mainitut asiat huomioiden.

Symboliseen malliin nämä yllämainitut asiat saadaan toteutettua siten, että alkuperäinen Javan rakenne ja viittaus siihen korvataan joko mallia luotaessa tai mallia läpikäytäessä alkuperäistä laajemmalla rakenteella (tyypillisesti oma haku- metodi *lookUp*), joka suorittaa täsmällisemmän ja yleisemmän identifioinnin rakenteelle.

Esimerkiksi symboli *port* saattaa olla sekä metodin argumenttina, muuttujana ja myös luokan attribuuttina. Siten eri metodeissa sama muuttujanimi viittaa eri asiaan. Tätä kutsutaan alias-ilmiöksi. Sen vuoksi metodi-kutsuissa tuloargumentit ja parametrit on saatava vastaamaan toisiaan.

Semantiikkateorian T4 tarkoituksena on varmistaa, että mallinläpikäyntialgoritmi voi toimia eri tilanteissa samaan tapaan kuin alkuperäinen Java-koodi tunnisten vastaavat rakenteet. Se tulkitsee siten koodin käyttäytymistä ohjelmoijan eduksi. Ohjelmoijan kiusana on ns. jojo-ilmiö, jossa kutsuttua oliorakennetta joudutaan etsimään laajalti olion perintähierarkiasta. Semantiikkateorian tuloksena saadaan kerättyä UML:n sekvenssikaavion mukaiset tiedot, mutta tarkempana sisältäen kaikki lausekkeet ja eri suoritusvariaatiot.

3.3.5 T5 Symbolinen malli ja mallin kutoja

Erilaisia malleja ja mallienluontimekanismeja (model weaver) on kehitetty runsaasti mm. OMG:n toimesta (Bezevin 2005). OMG on verrannut erilaisten malliratkaisujen ominaisuuksia seuraavasti. Tärkeitä vertailtavia piirteitä ovat modulaarisuus, muunnettavuus, jäljitettävyys, formalisoitavuus, koodin suoritettavuus, aspektien poiminnan mahdollisuus ja ratkaisun erikoistamisen mahdollisuudet. MOF-mallien keskeinen heikkous on assosiaatioiden käsittelyssä. Koska oliomallit voivat itsessään käyttää vain olioita, joiden heikkoutena on suljettu sisäinen rakenne, MOF-malleissa joudutaan jokainen assosiaatio pilkkomaan useiksi olioiksi (kuten association, associationEnd, link ja linkEnd). Pilkkomisen takia mallien läpikäynti vaatii paljon ohjelmointia ja monimutkaisia algoritmeja.

Takaisinmallintamisen kannalta tärkeimpiä mallien piirteitä ovat modulaarisuus, muunnettavuus, jäljitettävyys, formalisoitavuus ja suoritettavuus. OMG toteaa, että MDA:lla ei voida tuottaa suoritettavaa mallia, vaikkakin ”executable uml” on paljon käytetty käsite. Sen sijaan äskeisten teorioiden T1-T4 mukaan kehitetty malli on simuloitavissa eli mallinnettavissa niin pitkälle, että myös koodin sivuvaikutukset voidaan analysoida. Samalla simuloinnista saadaan tuloksena jälki (trace), joka vastaa dynaamisen analyysin tulosjoukkoa. Symbolinen malli on formalisoitavissa tarkkaan ja muutettavissa myös MDA-malleiksi ns. xmi-esitystavan avulla (XMI 2005).

Symbolisen mallin keskeisin ja lähes ainoa rakenne on symbolinen elementti (SE), jolla on optimoitu perintähierarkia. Se periytyy Symbolic-nimisestä luokasta, joka sisältää teorian T2 mukaiset määritelmät, parserin ja koodigeneraattorin. Eri tyyppiset elementit, jotka kuvattiin teorian T2 yhteydessä, toteutetaan peruselementin SE erikoistuksina. Kunkin elementin muistiin tallennetaan siihen kuuluva koodi, joka on luontivaiheessa muuntamalla teorian T3 mukaiset T2:n rakenteet omiksi elementeikseen ja prosessoimalla saatu koodi vielä teorian T4 mukaisesti. Näin saadaan aikaan atomistinen malli.

3.3.6 T6 Selektiivinen kysely

Tunnetuin koodin tarkastelun menetelmä on nimeltään viipalointi (slicing). Se tuottaa tietystä koodin osasta siihen suoranaisesti liittyvät muut ohjelmalauseet suoritusjärjestyksessä joko taakse- tai eteenpäin.
Viipalointiin liittyy kuitenkin paljon rajoituksia ja se soveltuu vain koodin matalan tason tarkasteluun. Siksi olemme valinneet paremman perusmenetelmän, joka on nimeltään leikkely eli chopping.

Chopping on tärkeä analysointikeino, joka tarkoittaa haluttujen kohteiden saavutettavuuden tutkimista. Se käy erinomaisesti syy-seuraussuhteiden käsittelyyn, joka on olennaista vianpaikannustehtävissä. Chopping pyrkii analysoimaan kuvan 3 mukaan perättäisten funktiokutsujen joukkoja, jotka alkavat tietystä Start-kohdasta ja etenevät aikajärjestyksessä Target-kohtaan saakka:

```
Start = fk •. fk-1 •. ... •.f  • Target.
```

Chopping-yhtälö voidaan ilmaista monella eri tavalla symbolista notaatiota käyttäen. Ilmaisu lopusta alkuun on muotoa:

```
Target = fk •. fk-1 •. ... •.f (Start).
```

Sisäkkäisessä muodossa se voidaan kirjoittaa:

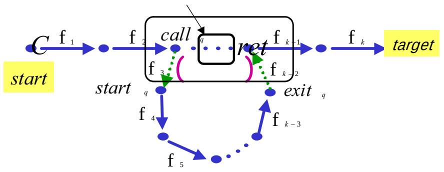
```
Target = f (f (f ( ... f( Start ) ) ) )
```

Tai peräkkäisessä muodossa listana:

```
Target = [f(XK), f(XK-1).. f(“Start”)].
```

Chopping-analyysin ohjelmointi Prologilla on selväpiirteistä. Seuraava koodi sisältää rekursiivisen säännön, joka kerää informaatiota seuraavilta perättäisiltä solmuilta, jotka on määritelty *f*-rakenteina sisältäen kunkin funktion nimen ja vastaavat argumentit. Tulokset saadaan kumulatiivisesti kerättyä predikaatin toiseen argumenttiin *Sequence*-muuttujan välityksellä:

```
chopping (Target, [f(This)|Sequence]):-
     f(NextId,Arguments),
     NextId:chopping(Target,Sequence).
```



Kuva 3. Chopping ohjausvuon tulkinnassa.

3.3.7 T7 Tyhjentävä haku ongelman ratkaisijana

Perinteinen koodin analysointi tarkastelee tyypillisesti ohjelman lauseiden välisiä suhteita sellaisenaan [Slicing, Reps] ilman korkeamman abstraktion malleja. Tarkka käsittely johtaa usein laskennallisesti liian vaativiin algoritmeihin, jos aluetta ei voida rajata tarpeeksi.
Äsken kuvaamamme selektiivinen kysely, chopping, sen sijaan tuottaa suppean tietoaineiston, joukon mallin elementtejä, jota voidaan käsitellä monin eri tavoin tyhjentävästi mm. tarkastelemalla eri suoritus-polkuvaihtoehtoja, datan kulkua ja sivuvaikutuksia ja mahdollisia aktivointeja eri tilanteissa.

Yhden metodin osalta kaikki Penningtonin tietotarpeet voidaan aina poikkeuksetta poimia täydellisesti siten, että tuntemattomat parametrit käsitellään viittauksina. Mikä tahansa suorituspolku voidaan myös evaluoida yhtenä kokonaisuutena, jolloin voidaan tutkia pienimpiä ilmiöitä koodista.

Ohjelmasilmukoiden käsittely on silti ongelmallista, jos suorituskertoja tulee lukemattomia. Siinä tapauksessa toistokertoja on voitava rajoittaa. Samoin ulkoisten rajapintojen tulkitseminen ehtolausekkeissa vaatii alkujärjestelyjä ennen testausta.

3.3.8 T8 Javan täydellinen relaatiomalli

Javaa ja sen koodista kehitettyä symbolista mallia tarkastellaan järjestelmällisimmin sitä varten suunnitellun, yhtenäisen relaatiorajapinnan kautta. Näin voidaan tunnistaa tärkeimmät käsitteet (entity) ja niiden väliset suhteet (relation), mikä tuottaa kelvollista tietoa vianpaikannukseen. Relaatiomalli sisältää seuraavat perusmääritykset, joiden välisiä relaatioita on mahdollista tutkia:

- Luokan nimi, class
- Luokan jäsenet
- Luokan super-luokka
- Metodin sisä ltämät elementit
- Metodin sisältämät staattiset kutsut
- Metodin suorittamat dynaamiset kutsut
- Kuhunkin elementtiin liittyvät navigointitiedot.
- Kuhunkin elementtiin liittyvät sivuvaikutukset
- Jäljityshistoria

Kun koodin yksittäiset rakenteet saadaan näin yksittäisillä kyselyillä hallintaan, tarkastelua voidaan laajentaa käyttämällä kyselyjen välillä loogisia operaatioita ja alakyselyjä.

3.3.9 T9 Selitysmalli

Tietämysteknisesti tarkasteltuna koodin informaatio on mielenkiintoista. Halutessamme kysyä syy-seuraussuhteita, koodista saadaan läpikäyntialgoritmilla loogiset ehdot ja rakenteet, jotka vaikuttavat kyseiseen tarkasteluväliin. Saatujen symbolisten rakenteiden joukko voidaan muuntaa luonnolliselle kielelle, esimerkiksi englanniksi, jolloin saadaan selväkielinen perusteluketju tarvittavine parametreineen.

Yhteistä kaikelle käsittelylle on se, että saadun tulosinformaation muoto on aina sama kyselystä riippumatta. Se helpottaa ohjelmointia ja toteutusta ja tulosten integrointia. Tulosinformaatiolle voidaan valita erilaisia tulosteita ja tarkastelutapoja kuten what, why tai how, jolloin näytölle syntyvä aineisto saadaan vastaamaan parhaiten käyttäjän tekemää kyselyä.

Tulostuksessa voidaan toki käyttää myös luonnollista semantiikkaa ja sen esitysasua, mutta se on ohjelmoijille vieraampi.

3.3.10 T10 Pragmaattisen tulkinnan teoria

Ylläpito sisältää paljon erilaisia tehtäviä ja alatehtäviä, joissa koodin tarkastelu on tarpeen. Aiheen laajuuden johdosta sitä käsitellään tässä kohdassa vain muutamin esimerkein Penningtonin tietotarvemäärittelyn pohjalta:

- Jokainen työkalun tuotos, joka johdattaa käyttäjää vianpaikannustehtävässä ja supistaa alkuperäistä informaatiojoukkoa, on pragmaattista informaatiota.
- Jokainen perehtymistilanteessa käyttäjälle tuotettu vastaus kysymyksiin what, why ja how, on pragmaattista informaatiota.
- Jokainen syy-seuraussuhdetta erittelevä haku, joka on kohdistettu vian määritykseen, tuottaa pragmaattista vian rajauksen informaatiota.
- Jokainen metodin suorituspolkujen hahmottamiseen liittyvä kokonaisuus palvelee vianhaun tarpeita tutkittaessa ohjelman pysähtyvyysongelmia.
- Jokainen kriittisen kuuntelijaolion tiloihin liittyvä kysely tuottaa pragmaattista informaatiota.

Teknologiamielessä pragmaattisen informaation tarkastelu vaatii käyttäjän tarpeiden tunnistamista ja metodologian sovittamista niihin. Saatu tietämys on kuitenkin arvokasta, sillä se palvelee kiireistä käyttäjää hänen jokapäiväisessä työssään sellaisenaan.

4 Metodologia

Edellä kuvattu ontologia ja tietämyksen määrittely eri teoria-alueineen toimivat lähtötietoina metodologian tarkastelulle.

4.1 Metodologian toteutus

Seuraavassa esityksessä Peircen semioottista käsitteistöä, jonka muodostavat merkki (*sign*), kohde (*object*) ja logiikalla toteutettava tulkinta (*interpretant, logic L*), lähestytään symbolista alkaen. Yhdistämällä nämä käsitteet saadaan hahmoteltua kokonaisuus SOL. Malli (M) määritellään yksinkertaisesti säiliöksi, joka sisältää malliin luodut oliot. Mallin analysointi (A) tapahtuu kyselyiden (Q, Query) avulla oliorajapinnan kautta.

4.1.1 Symbolin toteutus (S)

Kieliopista saadaan suoraan kaikki symbolit, jotka viittaavat lähdekoodin muuttujiin. Niitä ovat luokan nimi, metodi, attribuutti ja muuttujanimet. Koodin tarkasteluun tämä metoditarkkuus ei vielä riitä, koska kun saatua symbolista mallia halutaan tutkia esimerkiksi vianpaikannuksen tarkoituksessa, tarvitaan suorituspolun tarkkuus ja sen toteuttamiseen suorituspolkuelementti.

Jos erityisesti käyttäytymismalli ja elementtien vaikutukset kiinnostavat, tarvitaan lisäksi sivuvaikutuselementti, joka tallentaa kunkin elementin suorittamat ulkoiset muutokset.

Symboli tarkoittaa siis symbolisessa mallissa käsitettä, joka voi olla

- Käyttäjän antama alkuperäinen symboli
- Suorituspolkuun viittava symboli
- Sivuvaikutukseen viittaava symboli

Symboli on käytännössä vastaavaan työkaluun luotu viittaus vastaavaan olioon, joka voidaan tulostaa ja visualisoida eri tavoin sen sisällön ja tulkinnan mukaan. Kun käyttäjä napauttaa symbolin kuvaketta näytöllä hän pääsee kaikkiin sitä vastaavan olion ominaisuuksiin käsiksi, jolloin saadaan aikaan semanttinen käyttöliittymä. Symboli on siis esittämisen, kommunikaation ja ohjauksen väline.

4.1.2 Objektin toteutus (O)

Määrittelynsä mukaan oliojärjestelmän olio ja luokka sisältävät abstrahointia, yleistämistä ja erikoistamista tukevia piirteitä. Ne kaikki ovat hyödyllisiä lähdekoodin analysoinnissa ja soveltuvat Peiren semiotiikkaan seuraavasti:

- Abstrahoinnilla voimme luoda käsiterakenteita, jotka ovat kaikkien analyysien kannalta yhteisiä ja kehittää näin mahdollisimman teoreettista yhteistä mallia.
- Yleistämisellä voimme määritellä yhdenmukaisen olion työkalun luokkana. Tässä sitä kutsutaan symboliseksi elementiksi, *symbolicElement*. Se pystyy mallintamaan kaikkia ohjelman rakenteita, koska kaikilla rakenteilla on merkittävä määrä yhteisiä piirteitä.
- Voimme luoda riittävän tarkkoja erikoistettuja olioita symbolicn alaluokkina siten, että ne jakaantuvat teorian T2 ryhmittelyn mukaan.
- Voimme periyttää kaikki oliot superluokasta *symbolic* , joka sisältää symbolisen kielen määrittelyt. Näin kaikkien olioiden välille saadaan aikaan symbolinen kieli ja sen täydellinen tuki.

Ohjelmistokehitystasolla keskeisin olio, symbolinen elementti, toteuttaa kaikki tarvittavat piirteet rajapintojen ansiosta. Se periytyy *symbolic*-luokasta kuten edellä mainittiin ja sitä käyttävät super-luokkana kaikki symbolisen kielen tyyppirakenteet.

Puhdas oliomalli ei kuitenkaan sellaisenaan riitä olioiden esittämiseen, koska kapselointi rajoittaisi tiedonsaantia olioiden välillä ja tekisi mallin assosiaatioiden läpikäyntialgortmien rakentamisen hyvin vaikeaksi, koettaisiin samat ongelmat kuin xmi-malleissa. Siksi tarvitaan täydentävä ohjelmointiparadigma, logiikka, jota kuvataan seuraavassa.

4.1.3 Tulkinnan toteutus, logiikka (L)

Olion sisäiset tiedon tallennukset suoritetaan logiikan kautta predikaatteina ja faktoina. Oliot muodostavat keskenään navigointia tukevan kaksisuuntaisen verkoston, joka saadaan aikaan *child*- ja *parent*- sekä *next*- ja *previous*-faktoilla. Elementin sisältämä koodi tallentuu contents-nimiseen faktaan, joka on käytännössä symbolisen kielen rakenne, *Clause*.

Mallin läpikäyntialgoritmit käyttävät hyväkseen joko navigointifaktoja tai contents-faktan sisältöä.

Prologin päättelykone tarjoaa tukea läpikäynnin ohjelmoimiseen. Näin saadaan aikaan traverse-toiminto.

Mallin suorittamiseen tarvitaan run-toiminto, joka palauttaa oletuksena symbolisen kielen lauseita.

4.1.4 Mallin toteutus (M)

Yleistä symbolista mallia voidaan luonnehtia seuraavalla määrittelyllä:

```
model(L, B, T, F),
```

jossa L (layers) on tarvittavien mallin kerrosten määrä, B (bases) on tarvittavien erilaisten superluokkien määrä, T (types) on ratkaisun erilaisten tietotyyppien määrä ja F (features) on sellaisten funktioiden (piirteiden) määrä, jotka voidaan toteuttaa symbolisella elementillä.

Ylläolevasta tarkastelusta voimme päätellä seuraavaa:

- L on 1, koska tarvitaan vain 1 kerros, joka sisältää kaiken informaation.
- B on 1, koska kaikki oliot periytyvät symbolisesta elementistä, joka edelleen periytyy symbolisen kielen määrittelystä.
- T on 11, koska symbolisen kielen rakenteeseen liittyy 11 perustyyppiä.
- Mallilla toteuttavien piirteiden määrä, F, on periaatteessa ääretön, koska symboliselle mallille voidaan ohjelmoida lukemattomia erilaisia käsittelijöitä eri tarkoituksiin, mm. vastaamaan kyselyihin.

Edelliseen verrattuna MDA-mallit ovat huomattavasti monimutkaisempia, mutta silti heikompia ominaisuuksiltaan ja laajennettavuudeltaan.

Tyypillisimmät mallin toiminnot ovat seuraavat:

- Build rakentaa mallin ja sen oliot
- Traverse käy mallin sisältöä läpi tietyin oletuksin
- Chop hakee kahden metodin väliin kutsuhierarkiassa sijoittuvan metodien joukon.
- Run ajaa (simuloi) koodia halutusta kohdasta lähtien.
- Reset poistaa oliot muistista.

4.1. 5 Analyysin toteutus (A)

Edellä mainittiin mallin ja oliorajapinnan keskeisimpiä piirteitä. Ohjelmistoteknisesti analyysi on kuuntelija (observer), joka mallin käsittelyn käynnistyttyä jää odottamaan dataa. Analyysin parametrit määrittelevät mistä piirteestä kulloinkin on kysymys. Tulokset saadaan symbolisen kielen muodossa.

4.1.6 Kysely eli Query (Q)

Kuten edellä kerrottiin, tarkastelun tarpeilla on kysymysten kautta yhteys hypoteeseihin, joista edelleen on yhteys kyselyihin. Vastaus kysymyksiin what, why ja how saadaan joukkona symbolisen kielen rakenteita, jotka voidaan edelleen muuntaa kaavioiksi, näytöksi, tekstiksi tai xmi-muotoon siirrettäväksi muihin kehitysvälineisiin.

Tyypillinen kysely liittyy ongelman rajaukseen ja antaa tuloksena joukon elementtejä, joita epäillään. Näitä kutsumme ehdokkaiksi, vikaehdokkaiksi. Kun käyttäjä saa näkyville listan vikaehdokkaista, hän voi valita sopivan etenemistavan navigointiin ongelman ratkaisun täsmentämiseksi.

4.2 Yhteenveto metodologiasta

Symbolisen mallin rakentamiseen ja hyödyntämiseen tarvitaan seuraava teoriakokonaisuus:
T1 Java - koodin lukemiseen parsintapuuksi
T2 Uusi abstraktiotaso parsintapuulle
T3 Translaatio symboliseen kieleen
T4 Javan semantiikan siirto malliin
T5 Mallin kutoja luomaan yhtenäisen aineiston
T6 Suppea kyselyjärjestelmä alkurajaukseen
T7 Laajat kyselyt suorituspolkujen analysointiin
T8 Kohdekielen ja mallin relaatiomalli
T9 Selittämisen teoriat muuntamaan tulokset logiikaksi
T10 Teoriat pragmaattisen informaation tuottamiseen

5 Ylläpitotehtävän suorittaminen

Tutkimusrajauksessa tulotietona toimii ylläpitoon luotu tehtävä, jota kutsutaan muutosvaatimukseksi (Change Request, CR). Tällöin kyseessä on ongelma tai parannusehdotus tai perehtymistarve.

Seuraava proseduuri pyrkii luomaan toimivan yhteyden ylläpitotehtävän ja symbolisen mallin kyselyn välille.

5.1 Ylläpitometodi

Ylläpitotehtävän kulkua kuvataan seuraavassa lyhyen kirjainyhdistelmän TPHQ avulla.

5.1.1 Task (T)

Ylläpitotehtävän käsittelyä hajautetussa ohjelmistotyössä on kuvannut Walenstein (2002) RODS-käsitteistöllään, joka esittää kuinka tehtävän suoritus pyritään minimoimaan, kuinka suoritusalgoritmi halutaan optimoida, kuinka tehtävän eri osat jaetaan useille tekijöille ja kuinka erikoistuneita asiantuntijoita pyritään käyttämään tehtävän osien suorittamiseen. Jos tehtävä ei ole triviaali, se alkuvalmistelujen jälkeen muuttuu prosessiksi, josta kerrotaan seuraavaksi.

5.1.2 Prosessi (P)

Prosessin tarkoituksena on ratkaista monivaiheinen ongelma, joka vaatii useita erillisiä toimenpiteitä. Koodin tarkastelun välivaiheita on tutkinut mm. von Mayrhauser (1992). Ihminen ajattelee tällaisessa tilanteessa kysymysten ja vastausten kautta. Letovsky (1983) on kuvannut ylläpitoon liittyviä kysymyksiä luokittelulla : what, why ja how. Käyttäjä haluaa selvittää mitä jokin koodin osa tekee (what), miksi se suorittaa tiettyjä toimintoja (why) ja kuinka se toimii (how). Koska nämä tiedot eivät selviä lukematta koodia, hän joutuu arvamaan ja tekemään olettamuksia. Näin hän tulee luoneeksi hypoteeseja.

5.1.3 Hypoteesi (H)

Hypoteeseilla käyttäjä todistaa toimiiko esimerkiksi Server-luokka niin kuin hän oletti. Jos se toimii, hän voi kehittää hypoteesin pyrkien todistamaan sen oikeaksi tai vääräksi. Lopullinen päätelmä voidaan tehdä kun hypoteeseista on saatu riittäviä tuloksia. Vaikka tällainen hypoteesien käsittelyn prosessi kuulostaa vaivallolliselta, se on monessa tilanteessa esimerkiksi vianhaussa ainoa vaihtoehto. Jos käyttäjällä on käytössään vain editori, työ on hyvin hankalaa, mutta kehittyneet takaisin-mallintamisen työkalut voivat tässä häntä auttaa.

Peirce esittää ongelman lähestymiseen kolme erillistä tapaa: abduktion, induktion ja deduktion. Seuraavassa muutama esimerkki niistä:

- Abduktion käyttö ohjausvuon käsittelyssä: Kaikki ohjelman sekvenssit, jotka alustavat olion *Server*, ovat palvelimeen liittyviä olioita. Esimerkiksi koska metodi *main* alustaa serverin, metodi *main* on kriittinen palvelimen käynnistymisen kannalta.
- Induktio: Muuttuja *port* saa komponentin sisällä serverin aloituslogiikassa arvoja 80,81 ja 82, jotka kaikki ovat oikeita. Toinen vaihtoehto on se, että komponenttia kutsutaan ulkoa, jolloin portille ei voida asettaa käynnistyarvoa, mikä on riski. Siten on varmistettava nämä haarat ja tarkistuksena lisättävä oletusarvoksi 80. Näin voidaan yleistäen (induktiivisesti) päätellä, että kaikki tapaukset ovat kunnossa.
- Deduktio: Olion *Server* käynnistymiseen vaikuttavat komentoriviltä saatu arvo ja if-lausekkeessa saatu arvo. Käynnistyminen estyy, jos poikkeustilanne laukeaa main-metodissa. Nämä lauseet muodostavat

täydellisen logiikan (and) serverin käynnistymiseen. Jos olio ei käynnisty oikein, vian täytyy olla jossakin näistä kohdista.

Letovskyn kysymyksillä on seuraava yhteys koodin sisältöön:

- What – kysymys viittaa elementin tarkoitukseen ja toiminnallisuuteen (function).
- Why – kysymys viittaa syy-seuraussuhteeseen, joka on vianpaikannuksessa hyvin hyödyllinen piirre.
- How-kysymys viittaa elementin sisäiseen logiikkaan, sen suorituspolkuihin.

5.1.4 Kysely, Query (Q)

Kun kysymys on tunnistettu, se voidaan muuntaa kyselyksi. Jos kysymykseen liittyy kaksi rajausta, se voi olla esimerkiksi muotoa:

```
Query = why(Start,Target,Options)
```

Jos rajauksia on vain yksi, se on muotoa

```
Query = what(Object)
```

tai

```
Query = how(Object)
```

5.1.5 Systemaattinen vai opportunistinen oppiminen

Käyttäjä voi tehdä kyselyjä systemaattisesti tai opportunistisesti optimalla hakujen määrää.

Systemaattinen haku vie usein liian paljon aikaa, joten käyttäjän ammattitaito tulee esille, kun hänen pyrkii selviytymään ratkaisuun mahdollisimman joustavasti, mutta kuitenkin riittävän kattavasti.

Työkalu voi tukea tiedonhakua mm. tarjoamalla joustavan navigoinnin ja minimoimalla kyselyyn kohdistuvan informaation. Näistä piirteistä kerrotaan seuraavassa.

6 Työkalu JavaMaster

Edellä selostettiin karkealla tasolla kohdealueen metodologia ja työskentelymetodi. Seuraavaksi käydään läpi työkalu nimeltä JavaMaster ja sen peruspiirteitä.

Työkalun sisältämän Visual Prolog- kielisen lähdekoodin koko on noin 40.000 riviä. Se sisältää noin 450 luokkaa ja toimii Windows-ympäristössä.

6.1 Työkalun arkkitehtuuri

Takaisinmallintamistyökalun arkkitehtuuriksi muodostuu varsin luontevasti Model View Control (MVC) –rakenne, koska lähdekoodista saatu malli muodostaa tärkeimmän osan (Model). Käyttäjä ohjaa työkalua Control-osan kautta saaden aikaan näkymiä (View).

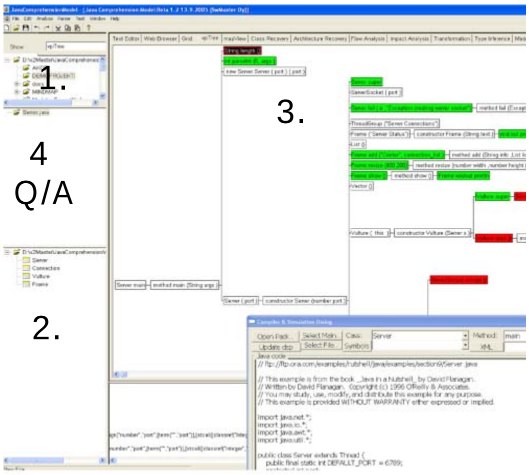
Navigointi tapahtuu näyttöjen kautta tai työkalun päämenuista.

6.2 Työkalun piirteet

JavaMasterin pääpiirteitä ovat: 1) lähdekoodin lataus muistiin ja malliksi, 2) ohjelmamallin tietojen selaus eri välilehtiä hyödyntäen, ja 3) kyselyjen suorittaminen tarkastelun hypoteesien mukaisesti.

Työkalu tarjoaa lukuisia erilaisia tarkastelutapoja. Latausvaiheessa se luo koodista työlistan, joka palvelee lähinnä koodiin perehtyjää, joka sen mukaisesti voi käydä läpi vaativimman aineiston, jota hän voi katsella puiden, taulukkonäyttöjen, riippuvuuskaavioiden, uml-kaavioiden ja xml-dokumenttien muodossa.

Kuva 4 esittää JavaMasterin käyttötilannetta. Koodi ladataan siinä tiedostopuun (kohta 1.) avulla muistiin. Lataus tuottaa puuhun (2.) ohjelmamallin mukaisen hierarkian. Sitten käyttäjä napauttaa Worklist-painiketta, jolloin työkalu näyttää vasteenaan sarjan navigointikysymyksiä (4. Q/A). Navigointitilannetta vastaava informaatio muodostuu eri näyttöihin (3.).


JavaComprehensionModel - [Java Comprehension Model Beta 1.2 13.9.2005 (SwMaster Oy)]
File Edit Analyze Parser Test Window Help
Show
1.
3.
4
Q/A
2.
Text Editor | Web Browser | Grid | Tree | mapView | Class Recovery | Architecture Recovery | Flow Analysis | Impact Analysis | Transformation | Type Inference
Server
Connection
Vulture
Frame
Server.java
Compiler & Simulation Dialog
Open Pack
Select Main
Update dsp
Select File
Class:
Server
Symbols
Method:
main
XML
Java code

```
// ftp://ftp.ora.com/examples/nutshell/java/examples/section9/Server.java

// This example is from the book _Java in a Nutshell_ by David Flanagan.
// Written by David Flanagan.  Copyright (c) 1996 O'Reilly & Associates.
// You may study, use, modify, and distribute this example for any purpose.
// This example is provided WITHOUT WARRANTY either expressed or implied.

import java.net.*;
import java.io.*;
import java.awt.*;
import java.util.*;

public class Server extends Thread {
    public final static int DEFAULT_PORT = 6789;
```

Kuva 4. JavaMasterin käyttöliittymä.

6.3 Työkalun evaluointi

JavaMasterin evaluoimiseksi on useita mahdollisuuksia:

- Koodi ja algoritmit voidaan verifioida analyyttisesti lause lauseelta.
- Työkalun tulokset voidaan evaluoida kokeellisesti tutkien mm. kutsupuiden kattavuus.
- Koodin sisäisiä rakenteita voidaan verrata AST-työkaluihin, joita löytyy mm. Eclipse-työkalusta.
- Käyttötapauksina voidaan tutkia tyypillisä ongelmanpaikannustilanteita, esimerkiksi tilanteita, joissa tietokantapalvelin toimii väärin. Tälloin selvitetään miten kuvattu ongelmatilanne on purettavissa hypoteeseiksi ja kuinka niitä vastaavat kyselyt on suoritettavissa työkalun avulla.

Työkalun evaluointiprosessi on vielä kesken, mutta sen absoluuttista kapasiteettia on jo evaluoitu syöttämällä työkalun symboliseen malliin miljoonan elementin aineistoja, jotka määrältään vastaavat Microsoftin Word97-ohjelmiston AST-solmujen määrää. Tällöin on todettu, että myös suurten ohjelmistojen analysointi on symbolisella mallilla mahdollista, vaikkakin menetelmän suurimmat edut löytynevät tiettyjen rajattujen ongelmalliseten koodialueiden tarkastelusta monin eri tavoin.

7 Yhteenveto

Esitetty symbolinen malli kuvaa sellaisenaan eräänlaisen virtuaalitoteutuksen, joka abstrahoiduilla rakenteillaan muistuttaa alkuperäistä koodia. Siten se poikkeaa merkittävästi konkreettisista työkaluista, jotka pyrkivät tarjoamaan käyttäjälle aina täydellisen ratkaisun. Abstrahoinnin tueksi Walenstein (2002) on esittänyt, että täydellisen ratkaisun tarjoaminen johtaa käyttäjän ylikuormittumiseen, siksi abstrahointi olisi suositeltavaa. Mutta mikä olisi abstraktin toteutuksen esikuva, onko sellaista esitetty aiemmin?

Kovin monenlaisia erilaisia teorioita abstrakteista simuloitavista koneista ei ole tehty, jos virtaalikoneita ei tässä huomioida. Ehkä tunnetuin niistä on Turingin kone (Kokkarinen 2003), joka käy läpi tulotietona olevaa nauhaa ja ohjausyksiköllään pyrkii reagoimaan nauhan tietoon määrittelyn mukaisesti.

Symbolinen analyysi muistuttaa monin tavoin Turingin konetta ja sen ohjausyksikköä. Symbolinen mallihan sisältää tilakonemaisen ohjausyksikön poimittuaan alkuperäisestä koodista logiikkapolut ja lauseiden semantiikan. Mutta toisin kuin Turingin kone, symbolinen malli sisältää myös formaalin tietoaineiston, joka jäljittelee alkuperäistä koodia.

Suorituksen aikana symbolinen malli tallentaa välitulokset ja lopputulokset perusolioihinsa, joten niiden sisältö palautuu mallin kyselyn kautta käyttäjälle. Siten saatu informaatio kuormittaa ”ohjausnauhaa” eli ihmisen ja koneen välistä kommunikointiväylää mahdollisimman vähän. Sehän oli abstrahoinnin tavoitekin.

Toinen mielenkiintoinen asia on tarkastella, kuinka hyvin saatu konstruktio vastaa semioottista käsitteistöä. Symbolisen mallin pääkäsitteistä voi havaita, että esitetty metodologia sopii suhteellisen hyvin yhteen Peircen käsitteistöön, koska molempiin liittyy kolme pääsuuretta: merkki (sign /symboli), kohde (object) ja tulkinta (interpretator) - sillä tarkennuksella, että tulkintaa symbolisessa mallissa vastaa logiikka, joka on toteutettu mallin elementin sisältämänä predikaattina.

Edellä kuvattu kolmijako muodostui metodologiaan tuntematta Peircen teoriaa vielä suunnitteluvaiheessa, mikä on mielenkiintoinen havainto. Siitä voidaan vetää johtopäätös, että konstruktio pääsi kehittymään oikealla tavalla oikeaan suuntaan jo alusta alkaen.

Kiitokset

SwMaster Oy on mahdollistanut alustavan tutkimuksen ja työkalun kehittämisen vuodesta 2000 alkaen. Sen lisäksi vuoden 2005 alusta Jyväskylän yliopiston COMAS-ohjelma on rahoittanut Erkki Laitilan väitöskirjahankkeeseen liittyvää tutkimusta.

References

Paul Anderson, Thomas Reps, and Tim Teitelbaum. "Design and Implementation of a Fine-Grained Software Inspection Tool", IEEE TSE 29 (8), 2003, pp. 721-733.

Richard Brooks. Towards a Theory of Understanding Computer Programs, IJMS 18 (6), 1983.

Jean-Marie Burkhardt, Francoise Detienne and Susan Wiedenbeck. "Object-Oriented Program Comprehension: Effect of Expertise, Task and Phase", Empirical Softw. Eng. 7 (2), 2002, pp. 115-156.

Gerardo Canfora, A. Cimitile, and U. de Carlini. "A Logic-Based Approach to Reverse Engineering Tools Production", IEEE TSE 18 (12), 1992, pp. 1053-1064.

Thomas Cheatham, Glen Holloway and Judy Townley. "Symbolic Evaluation and the Analysis of Programs", IEEE Trans. Softw. Eng. 5 (4), 1979, pp. 402-417.

Noam Chomsky. Three models for the description of language. IRE Transactions on Information Theory, IT-2:3:113–124, 1956.

CMMI. Capability Maturity Model Integration (CMMI), http://www.sei.cmu.edu/cmmi.

Eclipse. Eclipse Research Community: Eclipse, Open Source Project. http://www.eclipse.org [2003].

David Flanagan. Java Examples in a Nutshell: Example 7-5, ftp://ftp.ora.com/examples/ nutshell/java/examples, O'Reil-ly, 2000.

Frisco. A Framework of information system concepts. http://www.mathematik.uni-marburg.de/~hesse/papers/fri-full.pdf, 1998.

Susan Horwitz, Thomas Reps, and David Binkley. Interprocedural slicing using dependence graphs. ACM Trans. Program. Lang. Syst., 12(1):26–60, 1990.

Ilkka Kokkarinen. Tekoäly, laskettavuus ja logiikka, 2003.

Erkki Laitila. Visual Prolog: Teollisuuden Sovellukset, Docendo/Teknolit, Jyväskylä, Finland, 1996, 228 p.

Erkki Laitila. Method for Developing a Translator and a Corresponding System. Patent: W02093371, http://www. espacenet.com, 2001.

Erkki Laitila. "Program Comprehension Theories and Prolog Based Methodologies", Visual Prolog Applic. & Lang. Conf. (VIP-ALC 2006), Prolog Development Center, 2006.

Stan Letovsky. "Cognitive Process in Program Comprehension", Empirical Studies of Programmers: First Workshop, Ablex, 1986, pp. 80-98.

Esko Marjomaa. Theories of Representation, http://cs.joensuu.fi/^marjomaa/tr/tr/doc

Anne-Liese von Mayrhauser and Ann Marie Vans. "Hypothesis-Driven Understanding Processes During Corrective Maintenance of Large Scale Software", Proc. Int. Conf. Softw. Maint. (ICSM 1997), pp. 12-20.

Charles W. Morris. Signs, Language and Behaviour, Prentice-Hall, New York, 1946.

Charles W. Morris. *Writings on the general theory of signs*. The Hague: Mouton (1971).

OMG. UML, Model Driven Architectures, http://omg.org [2005].

Charles S. Peirce. Kotisivut, C.S. Peirce, http://peirce.org, 2006.

Nancy Pennington. "Stimulus Structures and Mental Representations in Expert Comprehension of Computer Programs", Cognitive Psychology 19 (3), 1987, pp. 295-341.

Prolog Development Center. Visual Prolog 6.3, http://www.visual-prolog.com, 2005.

Prolog Development Center. Visual Prolog Applic. & Lang. Conf. (VIP-ALC 2006), http://www.visual-prolog. com/conference2006, 2006.

Thomas Reps. "Program Analysis via Graph Reachability", ACM SIGPLAN 2000 Conference on Programming Language Design and Implementation (PLDI'2000).

Tamar Richner and Stand Ducasse. "Recovering High-Level Views of Object-Oriented Applications from Static and Dynamic Information", Int. Conf. Softw. Maint. (ICSM 1999), IEEE, pp. 13-22.

Spice. Software Process Improvement and Capability Determination. http://www.sqi.gu.edu.au/spice.

Sun. The Grammar of Java Programming Language, http://java.sun.com/docs/books/jls/secondedition/html/syntax.doc.html, 2003.

Turing Machines. http://plato.stanford.edu/entries/turing-machine.

Andrew Walenstein. Cognitive Support in Software Support in Software Engineering tools: A Distributed Cognition Framework, ftp://fas.sfu.ca/pub/cs/theses/2002/AWalensteinPhD.pdf.

Norman Wilde,and Ross Huitt. "Maintenance Support for Object-Oriented Programs", IEEE Trans. Software Eng. 18 (12), 1992, pp. 1038-1044.

Stefano Zacchiroli. Understanding Abstract Syntax Tree, http://www.connettivo.net/article.php3?id_article=49, 2003.

Scrap charge optimization using fuzzy chance constrained linear programming

Risto Lahdelma[*]
[*]University of Turku
Department of Information Technology
Joukahaisenkatu 3, FI-20520 TURKU, Finland
Risto.lahdelma@cs.utu.fi

Aiying Rong[†]
[†]University of Turku
Department of Information Technology
P Joukahaisenkatu 3, FI-20520 TURKU, Finland
aiyron@utu.fi

Abstract

We consider the problem of determining the optimal mix of different kinds the scrap in steel production. The uncertainty of the chemical composition of different kinds of scrap induces a considerable risk for the scrap mix failing to satisfy the composition requirements for the final product. We formulate the scrap charge optimization problem as a fuzzy chance constrained linear programming problem. We adopt a strengthened version of soft constraints to interpret the fuzzy constraints based on the application context and form a crisp model with consistent and compact constraints for solution. The simulation results based on the realistic steel production data show that the failure risk can be hedged by proper combination of aspiration levels and confidence factors for representing the fuzzy number. There is a tradeoff between failure risk and material cost. The presented approach applies also for other scrap-based production processes, such as aluminum and copper production.

Keywords: Fuzzy sets, linear programming, chance constraint, scrap charge optimization, steel production.

1 Introduction

A general trend during the past decades is that scrap-based steelmaking has increased its share, reaching around 40% of global crude steel production in 2001 (Rautaruukki 2001). Steel is also the world's most important recycled material. The use of the steel scrap as a raw material for steelmaking results in a saving of 600 million tonnes of iron ore and 200 million tonnes of coke each year (EUROFER 2001). With the growing concern on environmental issue, the popularity of using scrap could further increase because scrap-based steelmaking emits significantly less CO_2 as compared with integrated steelmaking using metallurgical coke as reductant for iron-making. Undoubtedly, the use of the scrap offers the opportunity to produce high quality products most economically. In the meantime, it also poses challenge in charge optimization caused by the uncertainty of chemical composition of scrap. The uncertainty mainly comes from two sources.

First, the constituents in the scrap and steel product are diverse. With the development of steel products for higher grades and higher performance, use of steel materials in combination with nonferrous metals or non-metallic materials has increased. Depending on the melting process and the requirements of the particular product, some of the constituents in scrap are considered impurities while others are valuable additives.

Second, the diverse scrap materials are generally divided into scrap types for handling and the materials included in each scrap type are heterogeneous because the classification varies based on different criteria. That means there are large deviations in material properties inside class, sometimes larger than between classes. Therefore, it is difficult to come up with accurate chemical composition (element concentration) analysis for the scrap.

Scrap-based steelmaking process starts with the charge of the predetermined scrap mix in an electric arc furnace (EAF). One furnace of refined steel (with required chemical composition) is called a *heat*. When the scrap mix is melted, there is a considerable risk for the outcome failing to satisfy the composition requirements for the product because of the uncertainty in the chemical composition for the scrap. The objective of the scrap charge optimization is to select the most economical scrap mix for each produced steel grade while minimizing the failure risk caused by uncertainties in raw material consistence.

There are three kinds of charge optimization models based on different planning horizon: single-heat model (AFS 1986) for short-term planning (e.g. daily operations), multi-heat model (Kim & Lewis 1987) for medium-term planning (e.g. weekly or monthly) and product recipe model (Lahdelma et al. 1999) for long-term planning (annually). Any charge plan is implemented on single-heat basis and on-line analysis is applied to identify whether the outcome match the predicted characteristics and to correct the possible bias in future predictions (Wilson et al. 2001). However, if the charge planning is designed using the single-heat model, this may result in non-optimal use of raw materials. Minimizing raw material costs in one heat can eat up the cost-efficient raw materials and therefore increase the costs in future heats. The multi-heat model can allocate the available raw materials optimally among the different heats based on the current raw material stock and predicted deliveries. For long-term planning, it is more convenient to use product recipe model which can viewed as a model where all heats of the same product are grouped. Therefore, the product recipe model is much smaller than the multi-heat model. The long-term scrap charge plan is designed based on forecast customer orders and possible available raw materials. The product recipe model can also be applicable to medium-term planning.

In terms of handling uncertainty, there are several methods of constraining failure risks. Bliss (1997) added safety margins to the product standard by using a tighter product standard in optimization. Lahdelma et al. (1999) added safety margins to the chemical composition for scrap in optimization. The approach of adding safety margin can be viewed as the extension of the deterministic model to accommodate the uncertainty. Turunen (2000) constrained the failure risk based on stochastic chance constraints to guarantee that the failure rate is less than a predetermined level (or to allow small violations in some constraints). However, stochastic chance constrained programming models are difficult to solve since the deterministic version of the model is non-linear (Kall & Wallace 1994, Watanabe & Ellis 1994). The solution can be even more complicated if the stochastic model includes simultaneously chance constraints and ordinary constraints with stochastic parameters.

In this paper, we formulate the scrap charge optimization problem based on the product recipe model. We represent the uncertainty for the scrap composition and the composition specification for product standard based on fuzzy set theory (possibility theory, Zadeh 1978). Then we constrain the failure risk based on a possibility measure. Consequently, the scrap charge optimization problem is modeled as a fuzzy linear chance constrained programming problem. This approach can be viewed as explicitly integrating methods of adding safety margin for product standard and for scrap composition and allowing small violations (chance constraints) in some constraints under a single framework. Since the constraints in the scrap charge problem mainly address the specification of the product, the crisp equivalence of the fuzzy constraints should be less relaxed than that purely based on the concept of soft constraints. For general discussions on the interpretation of the ordinary fuzzy constraints based on the concept of soft (flexible, approximate) constraints, we refer to Canz (1996), Dubois & Prade (1980), Slowinski (1986), Werners (1987), Zimmermann (1978). For interpretation of general chance constraints directly based on the possibility measure, we refer to Liu & Iwamura (1998). Here we interpret chance constraints based on the likelihood measure and the resulting crisp constraints are much stricter than those based directly on the possibility measure. The property of the likelihood measure introduced in this paper is similar to that of the likelihood profile discussed in the fuzzy machine scheduling context (Wang et al. 2002). There are two ways to interpret ordinary fuzzy constraints: soft constraints and tolerance constraints. The strict tolerance constraint (Dubois & Prade 1980) means that that fuzziness of the right-hand side of the constraint is interpreted as a maximum tolerance for that of the left-hand side of the constraint. However, the system with the strict tolerance constraints may be inconsistent. Therefore, we interpret the ordinary fuzzy constraints based on the framework of relaxed tolerance constraints where we try to eliminate the possible conflicts of the strict tolerance constraints by dropping the relatively less critical constraints or by weakening some constraints. Finally a crisp model with consistent and compact constraints is formed. The resultant crisp model is a standard linear programming (LP) model, which can be solved by standard LP software.

The paper organizes as follows. In Section 2, we review a generic fuzzy linear programming model and describe the relationship between the fuzzy number and the stochastic parameter with given distribution. In Section 3, we describe the scrap-based steelmaking process first, and then we formulate the scrap charge optimization problem as a fuzzy chance-constrained linear programming model. In Section 4, we transform the fuzzy model into a crisp model with compact and consistent constraints based on the application context. In Section 5, we report the simulation results.

2 Fuzzy linear programming model and fuzzy numbers

We define fuzzy linear programming (FLP) as the extension of the classical linear programming (LP) in operations research (Taha 1992) in the presence of uncertainty in the optimization process where some or all of the coefficients are represented as fuzzy quantities (numbers) based on fuzzy set theory (possibility theory). A non-fuzzy quantity can be viewed as a special case of fuzzy quantity as discussed later. A generic FLP model is given below.

$$\min \sum_{j=1}^{n} \tilde{c}_j x_j \qquad (1)$$

s.t.

$$Pos\left\{\sum_{j=1}^{n} \tilde{a}_{ij} x_j \le \tilde{b}_i\right\} \ge \alpha_i, \quad i = 1,\ldots,m_1, \qquad (2)$$

$$\sum_{j=1}^{n} \tilde{d}_{ij} x_j \le \tilde{e}_i, \quad i = m_1+1,\ldots,m, \qquad (3)$$

$$x_j \geq 0,\ j = 1,..,n, \qquad (4)$$

where x_j are decision variables, $\tilde{c}_j$ are cost coefficients, $\tilde{a}_{ij}, \tilde{d}_{ij}$ are technical coefficients, and $\tilde{b}_i, \tilde{e}_i$ are right-hand side coefficients. Some or all of these coefficients can be fuzzy numbers. Formula (1) is the objective function to be optimized. Constraints (2) are called chance-constraints. *Pos*{.} denotes the possibility of events {.}. The predetermined aspiration level α_i is usually interpreted as constraint reliability. The constraints (2) mean that the possibility for violating $\sum_{j=1}^{n} \tilde{a}_{ij} x_j \leq \tilde{b}_i$ should be less than 1-α_i. Constraints (3) are ordinary linear constraints.

Fuzzy numbers play a central role in the fuzzy programming model. First, we review the concept of fuzzy numbers briefly. Then we represent the statistical uncertainty based on fuzzy set theory and establish the relationship between the fuzzy number and the stochastic parameter with given distribution.

A fuzzy number is a convex normalized fuzzy set *A* on the real line *R* with membership function $f_A(x)$ ($x \in R$). We represent the fuzzy number based on the modified *L-R* fuzzy number. The examples of *L-R* fuzzy numbers can be referred to Dubois & Prade (1980) and Slowinski (1986). The representation for *A* is the 3-tuple of parameters $A = (a, \underline{a}, \overline{a})$, where *a*, $a - \underline{a}$ and $\overline{a} - a$ are mean, left and right spreads of the fuzzy number. The member function $f_A(x)$ is given below.

$$\begin{array}{ll} L((a-x)/(a-\underline{a})) & \text{if } \underline{a} \leq x \leq a, \\ R((x-a)/(\overline{a}-a)) & \text{if } a \leq x \leq \overline{a,} \\ 0 & \text{otherwise,} \end{array}$$

$$f_A(x) = \text{¿}\{\text{¿}\{\text{¿¿¿} \qquad (5)$$

where *L* and *R* are continuous decreasing functions such that *L*(0) = *R*(0) = 1, *L*(1) = *R*(1) = 0. If both *L* and *R* functions are linear, then the resultant fuzzy number is called triangular fuzzy number as shown in Figure 1, where f_A^L and f_A^R are *L* and *R* functions and $f_A^L(u) = f_A^L(u) = 1 - u, 0 \leq u \leq 1$. A non-fuzzy number can be treated as a special case of the fuzzy number where both the left and right spreads are zero.

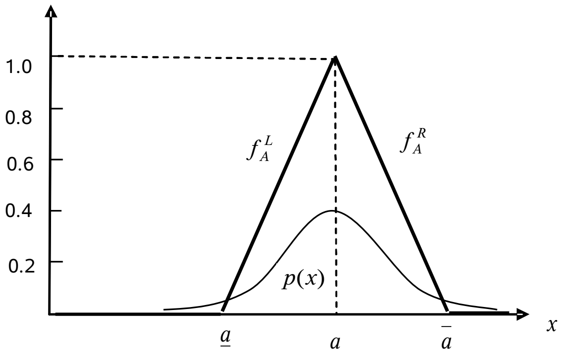


Figure 1 Relationship between a fuzzy number $A = (a, \underline{a}, \overline{a})$ and a stochastic parameter with density function *p*(*x*) (*a* is the mean of the stochastic parameter).

Figure 1 also illustrates the relationship between a fuzzy number $A = (a, \underline{a}, \overline{a})$ and a stochastic parameter with density function *p*(*x*). The uncertainty involved in many applications can be viewed as statistical uncertainty. Canz (1996) discussed the basic approach for representing statistical uncertainty using trapezoidal fuzzy numbers. Here, we follow the similar logic and represent the statistical uncertainty using modified *L-R* fuzzy numbers. First we assume a probability density function *p*(*x*) from the statistical data, and then we transform it into a possibility distribution (fuzzy set) based on the shape of *p*(*x*) and the mean and standard deviation of the statistical data. If the mean and standard deviation of the density function *p*(*x*) is β and σ respectively, then for a fuzzy number $A = (a, \underline{a}, \overline{a})$, $a = \beta \pm \varepsilon$, $\underline{a} = a - n_1 \sigma$ and $\overline{a} = a + n_2 \sigma$, where ε is related to the skewness (the third moment, Banks & Carson 1984) of the probability density function *p*(*x*), n_1 and n_2 are confidence factors. The choice of n_1 and n_2 depends on shape of *p*(*x*) and the application context. If the density function *p*(*x*) is skewed (e.g. log normal distribution), then n_1 and n_2 should be different. If *p*(*x*) is symmetric (e.g. normal distribution), then $a = \beta$, $n_1 = n_2$. However, if the parameter has non-negativity restriction then $\underline{a} = \max(0, a - n_1 \sigma)$. If $a - n_1 \sigma < 0$, this is equivalent to case that n_1 and n_2 take different values. The value setting of n_1 and n_2 depends on the requirement of the application. The choice of *L* and *R* functions is flexible and simple linear functions can satisfy the needs for most applications.

3 Problem description

To understand the scrap charge optimization problem, we start by introducing the scrap-based steelmaking process, e.g. stainless steel-making process.

3.1 Scrap-based steel-making process

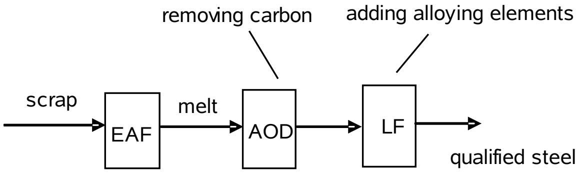


Figure 2. Scrap-based stainless steel-making process

Figure 2 shows a sketch of the scrap-based stainless steel-making process. The scrap metal is charged into an electric arc furnace (EAF) and melted. Then the melt is moved into argon-oxygen decarburization (AOD) converter where the carbon is removed by argon and oxygen injection in order to meet the steel quality requirement. After AOD, the molten steel is moved into ladle furnace (LF) for secondary metallurgical operations such as adding alloying elements, degassing and temperature homogenization.

The scrap consists of materials from different sources such as old cars, structural steel, cans, electric appliances and industrial by-products. Scraps contain also waste from the manufacturing process of steel products such as cut edges of steel plates and ends of coils. Generally the diverse materials are divided into several standard types for trade. After the scrap materials are delivered to the meltshop, the steel producers have their own internal classification system that further divides the standard types into different subtypes based on origin, supplier, chemical contents and size distribution and requirements. The total number of scrap types is about 20 for a normal carbon-steel meltshop. For stainless and speciality production, the classification can be even finer and the number of scrap types can be more than 100 (Sandberg 2005). For the stainless steel production, it is particularly important to sort grades with high content of valuable alloying metals such as nickel or molybdenum in their own classes so that the scrap can substitute for the expensive pure alloying materials to provide alloying elements in the new product. It would be good for the production if all different alloys could be separated to different classes but it seems that there are large deviations in properties (e.g. element concentrations) inside classes, sometimes larger than between classes. Some scrap classes are much more heterogeneous than others. That is, there is uncertainty for element concentration for the scrap. The uncertainty can cause errors for element concentrations in the final products.

The scrap also contains oxides, organic materials that burn in the process, and the metal itself forms oxides in the process. The *yield* of the raw material (material yield) is the share of the raw material that becomes part of the product, typically 70-95 %. We also use a yield parameter for each element (element yield coefficient). The element yield coefficients depend on the conditions in the furnace, while the material yield depends mainly on the physical and chemical properties of the material. The two yield parameters multiplied together give the net yield of an element for a raw material. The yield and element concentrations of the scrap types can be estimated based on spectrographic analysis in conjunction with analysis of the process and heat data (Ikäheimo 1999, Sandberg 2005).

The process conditions may also vary and produce errors not related to scrap. Inaccuracies in the charging process constitute a third source of error. However, the element concentrations for the scrap are the most significant source of uncertainties in the process. We can treat this uncertainty as the sources for all the errors.

3.2 Scrap charge optimization problem

The objective of the charge optimization is to fulfill a set of customer orders at the lowest cost using available raw materials. The costs include the raw material cost and an approximation of the costs in successive process stages (the AOD converter and LF). Charge optimization models usually have constraints for the chemical balance of the steel, production amounts and material resources (Kim & Lewis 1987). Here we also consider the uncertainty in element concentrations for different scrap types. The uncertainty of the element concentrations for the scrap will cause the element concentration in the final product to deviate from the product standard. The product standard is specified by lower and upper limits for each alloying element concentration. A failure happens when the concentrations of the alloying elements of the molten steel in EAF exceeds the upper limits of the product standard because it is usually difficult to remove any element from the liquid steel. Falling below lower limits is not as critical because the element contents can be increased by adding pure alloying elements. However, it causes cost increase. On the one hand, the raw material cost will increase because the pure alloying materials are more expensive than the scrap. On the other hand, additional processing cost will be incurred in the subsequent stages in LF because the processing time will be increased. Therefore, the ideal scrap mix would yield the element concentrations that are close to the lower bounds of product standard.

The orders are divided into $|K|$ batches and each batch consists of products of the same standard. The element concentrations of raw materials and the element concentration requirements of final product are represented as modified *L-R* fuzzy numbers. We constrain the failure risk based on the possibility measure to control the failure rate within the predetermined level. The following notations are introduced.

Index Set

I Set of all useful elements in the product including element iron.

I_1 Set of alloying elements.

J Set of all raw materials, $J = J_0 \cup J_1$.

J_0 Set of scrap materials.

J_1 Set of pure alloying materials, each of which is made up of one alloying element.

K Set of batches.

Parameters

a_j Yield of raw material $j \in J$.

b_i Yield coefficient of element $i \in I$.

c_j Cost of raw material $j \in J$. For the alloying materials $j \in J_1$, additional processing cost is included in material cost.

m_k Mass of products in batch $k \in K$.

$\underline{x}_{k,j}$, $\overline{x}_{k,j}$ Lower and upper bounds for charge of raw material $j \in J$ in batch $k \in K$.

$\underline{x}_j$, $\overline{x}_j$ Lower and upper bounds for usage (consumption) of raw material $j \in J$.

$\lambda_{k,i}$ Aspiration levels for controlling the upper limit of the concentration of alloying element $i \in I_1$ in batch $k \in K$.

$\tilde{p}_{ji}$ $(p_{ji}, \underline{p}_{ji}, \overline{p}_{ji})$ Concentration of element $i \in I$ in raw material $j \in J$ (fuzzy number).

$\tilde{t}_{ki}$ $(t_{ki}, \underline{t}_{ki}, \bar{t}_{ki})$ Concentration of alloying element $i \in I_1$ in batch $k \in K$ (fuzzy number).

Decision variables

x_{kj} Charge (consumption) of material $j \in J$ in batch $k \in K$.

Then the scrap charge optimization problem for minimizing the raw material costs including additional processing costs is represented as follows.

$$\min \sum_{k \in K} \sum_{j \in J} c_j x_{kj} \qquad (6)$$

s.t.

$$\sum_{j \in J_0} b_i a_j \tilde{p}_{ji} x_{kj} ¿ m_k \tilde{} t_{ki}, \quad k \in K, i \in I_1, \ (7)$$

$$Pos\left\{\sum_{j \in J_0} b_i a_j \tilde{p}_{ji} x_{kj} \leq m_k \bar{t}_{ki}\right\} \geq \lambda_{ki}, \quad k \in K,$$

$$i \in I_1, \qquad (8)$$

$$\sum_{j \in J} b_i a_j \tilde{p}_{ji} x_{kj} = m_k \tilde{t}_{ki}, \quad k \in K, i \in I_1, \ (9)$$

$$\sum_{i \in I} \sum_{j \in J} b_i a_j p_{ji} x_{k,j} = m_k, \quad k \in K,$$

$$(10)$$

$$\underline{x}_j \leq \sum_{k \in K} x_{kj} \leq \overline{x}_j, \quad j \in J, \qquad (11)$$

$$\underline{x}_{kj} \leq x_{kj} \leq \overline{x}_{kj}, \quad k \in K, j \in J. \qquad (12)$$

Constraints (7)--(9) together are used to control the concentrations of alloying elements for raw materials. Constraints (7) are general requirements for the concentrations of the alloying elements for the scrap materials. The concentrations of the alloying elements for the scrap in EAF cannot exceed the product standard. Constraints (8) are chance constraints particularly constraining the failure risk based on a possibility measure. The failure possibility that the concentrations of the alloying elements for the scrap exceed the upper limits of the product standard must be less than the predetermined levels $1-\lambda_{ki}$ for each alloying element $i \in I_1$ in each batch $k \in K$. The effect of the chance constraints is two-fold. On the one hand, they emphasize that controlling the upper limits of the alloying element concentrations is critical. On the other hand, they give flexibility to control the limits by choosing different aspiration levels, allowing explicitly small violations of the upper limit of the product standard. Constraints (9) state that the concentrations of the alloying elements in the final product must meet the product standard. Constraints (10) state the mass balances between raw materials and final products. Here the constraints are crisp. We should know that the summation of concentrations of all elements must be one in the original raw materials. Some elements may be burnt off in the process and the summation of concentrations of all elements that become part of product is fixed for a given material. The mass is the aggregate of the share of all elements that become part of products for the selected materials. Therefore, the uncertainty in element concentrations has only slight effect on mass because the increase in concentration for one element implies the decrease in concentration for other elements. The variations in mass mainly come from the possible different element yield coefficient b_i for different element $i \in I$. Constraints (11) and (12) give the bounds for raw material consumption.

4 Transformation of the fuzzy model into its crisp equivalence

In the fuzzy environment, there are two ways to treat the ordinary fuzzy constraints: tolerance constraints and soft (approximate) constraints (Dubois & Prade 1980). Tolerance constraints mean that the fuzziness of the right-hand side of the constraint is interpreted as a maximum tolerance for that of the left-hand side of the constraints. The soft constraints mean that the satisfaction of the constraints is determined by the membership function and there is compensation between the satisfaction of the constraints and fulfillment of the objective. That is, the interpretation of the fuzzy constraints based on the concept of soft constraints results in the crisp constraints that are more relaxed than the corresponding constraints assuming that there is no fuzziness in the coefficients, while the crisp constraints based on the concept of tolerance constraints are much stricter. Based on this logic, we can find later the interpretation of the chance constraints directly based on the possibility measure follows the line of soft constraints.

In case of multi-criteria decision environment where criteria are described by objective functions and possible alternatives are implicitly determined by constraints, it is reasonable that the fuzzy constraints are interpreted as soft constraints which can provide a certain degree of freedom in determining the set of feasible solutions. However, for the scrap charge problem, the interpretation of the fuzzy constraints should be stricter than that based on the concept of soft constraints because the constraints mainly impose physical restrictions on the product and must be somewhat strictly satisfied. Therefore, we interpret the chance constraints based on the likelihood measure to obtain a stricter interpretation than that based directly on possibility measure. We interpret the ordinary fuzzy constraints based on the framework of the relaxed tolerance constraints where we try to eliminate the possible conflicts from the strict tolerance constraints by dropping the less critical constraints or by weakening some constraints. For simplicity, we assume that *L*-*R* fuzzy numbers are triangular. Different *L* and *R* function types mainly affect the setting of the aspiration level and complexity for the realization of chance constraints.

4.1 Interpretation of chance constraints

Based on the operation of fuzzy numbers

$$\sum_{j\in J_0} b_i a_j \tilde{p}_{ji} x_{kj} =$$

$$\Big(\sum_{j\in J_0} b_i a_j p_{ji} x_{kj}, \sum_{j\in J_0} b_i a_j \underline{p}_{ji} x_{kj}, \sum_{j\in J_0}$$

$$, \qquad (13)$$

where $\tilde{w}_{ik}$ is also a triangular fuzzy number.

Then constraints (8) become

$$Pos\{\tilde{w}_{ki} \leq m_k \bar{t}_{ki}\} \geq \lambda_{ki} \qquad (14)$$

Let $f_{\tilde{w}_{ki}}$ (*x*) be the membership function of the fuzzy number $\tilde{w}_{ik}$ (Figure 3). $f^L_{\tilde{w}_{ki}}(u) = f^U_{\tilde{w}_{ki}}(u) = 1 - u$,$u \in [0,1]$ are *L* and *R* functions. Then based on (5) $f_{\tilde{w}_{ki}}$ (*x*) can be represented as follows.

$$f^L_{\tilde{w}_{ki}}((w_{ki} - x)/(w_{ki} - \underline{w}_{ki})) \text{ if } \underline{w}_{ki} \text{¿} x \leq w_{ki},$$

$$f^U_{\tilde{w}_{ki}}((x - w_{ki})/(\overline{w}_{ki} - w_{ki})) \text{ if } w_{ki} \text{¿} x \leq \overline{w}_{ki},$$

$$0 \qquad \text{otherwise,}$$

$$\text{¿}$$

$$f_{\tilde{w}_{ki}}(x) = \text{¿}\{\text{¿}\{\text{¿¿¿}$$

$$\text{¿}$$

$$(15)$$

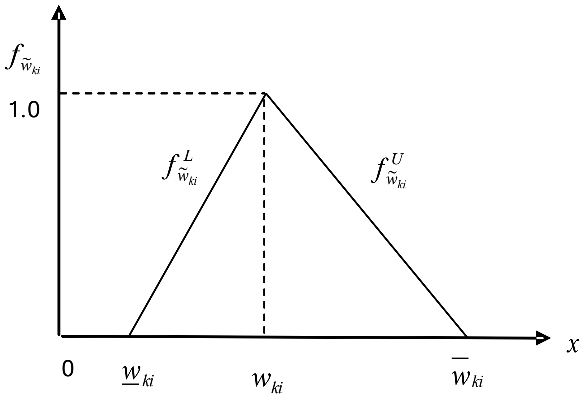


Figure 3. The membership function of a fuzzy element concentration $\tilde{w}_{ki}$.

Then the left hand side of (14) can be calculated as follows.

$$Pos\{\tilde{w}_{ki} \leq m_k \bar{t}_{ki}\} = \sup\{f_{\tilde{w}_{ki}}(x) \mid x \leq m_k \bar{t}_{ki}\}$$

$$0 \qquad \text{if } x \leq \underline{w}_{ki} ,$$

$$f^L_{\tilde{w}_{ki}}((w_{ki} - x)/(w_{ki} - \underline{w}_{ki})) \text{ if } \underline{w}_{ki} \text{¿} x \leq w_{ki},$$

$$1 \qquad \text{if } x \geq \overline{w}_{ki}.$$

$$\text{¿}$$

$$= \text{¿}\{\text{¿}\{\text{¿¿¿}$$

$$\text{¿}$$

$$(16)$$

If the realization of (14) is directly based on (16), we can see that any aspiration level λ_{ki} can be satisfied by the value (element concentration) less than w_{ki} . This implies the concept of soft constraints. However, this realization is too relaxed for the scrap charge problem because it cannot control failure rate effectively.

To enforce the chance constraints more strictly, we construct the likelihood of the *valid* element concentration with fuzzy possibility distribution. We define the likelihood of the valid element concentration for a fuzzy element concentration in the similar way as Wang et al. (2002) defined the job completion likelihood profile for the fuzzy processing time in the machine scheduling context.

Let $\mu_{\tilde{w}_{ki}}$ denote the likelihood of the *valid* element concentration for a fuzzy element concentration $\tilde{w}_{ki}$. For a given $\tilde{w}_{ki}$, $\mu_{\tilde{w}_{ki}}$ represents the likelihood of its element concentration to be valid within a certain allocated upper limit *y* (variable) for the product standard. If $\underline{w}_{ki} \geq y$, the element concentration is invalid and $\mu_{\tilde{w}_{ki}}$ = 0. If $\overline{w}_{ki} \leq y$, then the element concentration is completely valid $\mu_{\tilde{w}_{ki}}$ = 1. When $\underline{w}_{ki} \leq y \leq \overline{w}_{ki}$, $\mu_{\tilde{w}_{ki}}$ should vary from 0 to 1 based on an increasing function. We wish that $\mu_{\tilde{w}_{ki}}:\left[\underline{w}_{ki},\overline{w}_{ki}\right]\leftrightarrow[0,1]$ is a continuous bijective mapping, i.e. there is only element concentration in $\left[\underline{w}_{ki},\overline{w}_{ki}\right]$ corresponding to a given aspiration level $\mu_{\tilde{w}_{ki}}=\lambda_{ki}$. $\mu_{\tilde{w}_{ki}}$ can be constructed based on $f_{\tilde{w}_{ki}}$ as follows (Figure 4).

$$0 \qquad\qquad\qquad\qquad \text{if } y\leq\underline{w}_{ki}$$

$$f^{L}_{\tilde{w}_{ki}}((w_{ki}-y)/(w_{ki}-\underline{w}_{ki}))/2 \qquad \text{if } \underline{w}_{ki}\text{¿}\, y\leq v$$

$$(2-f^{U}_{\tilde{w}_{ki}}((y-w_{ki})/(\overline{w}_{ki}-w_{ki})))/2 \text{ if } w_{ki}\text{¿}\, y\leq$$

$$1 \qquad\qquad\qquad\qquad \text{if } y\geq\overline{w}_{ki}$$

¿

$$\mu_{\tilde{w}_{ki}}(y)=\text{¿}\{\text{¿}\{\text{¿}\{\text{¿¿¿}$$

¿

(17)

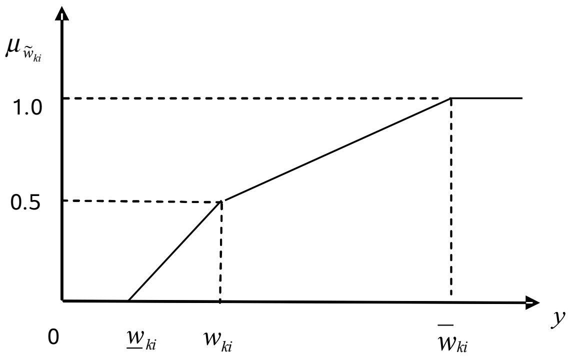


Figure 4. The likelihood of the valid element concentration for a fuzzy element concentration $\tilde{w}_{ki}$.

Our application background requires that $\lambda_{ki}>0.5$ for the chance constraints. This means that

$$\lambda_{ki}=(2-f^{U}_{\tilde{w}_{ki}}((y-w_{ki})/(\overline{w}_{ki}-w_{ki})))/2$$

$$f^{U}_{\tilde{w}_{ki}}((y-w_{ki})/(\overline{w}_{ki}-w_{ki}))=1-(y-w_{ki})/(\overline{w}_{ki}$$

¿

$\{$¿¿¿

¿

$$\Rightarrow y=w_{ki}+(2\lambda_{ki}-1)(\overline{w}_{ki}-w_{ki}) \qquad (18)$$

Then the crisp equivalence of chance constraints (8) based on (13) and (18) is given below.

$$\sum\nolimits_{j\in J_0} b_i a_j (p_{ji}+(2\lambda_{ki}-1)(\overline{p}_{ji}-p_{ji}))x_{kj}\leq m_{l}$$

$$k\in K,\ i\in I_1. \qquad (19)$$

4.2 Interpretation of ordinary constraints

For a generic ordinary inequality fuzzy constraint

$$\sum_j \tilde{d}_{ij}x_j\leq\tilde{e}_i, \qquad (20)$$

where $\tilde{d}_{ij}$ and $\tilde{e}_i$ are *L-R* fuzzy numbers, $\tilde{d}_{ij}=(d_{ij},\underline{d}_{ij},\overline{d}_{ij})$ and $\tilde{e}_i=(e_i,\underline{e}_i,\overline{e}_i)$. If a fuzzy constraint is interpreted as a tolerance constraint, then crisp equivalence is given below (Dubois & Prade 1980).

$$\sum_j d_{ij}x_j\leq e_i,$$

$$\sum_j (\overline{d}_{ij}-d_{ij})x_j\leq\overline{e}_i-e_i,$$

$$\sum_j (d_{ij}-\underline{d}_{ij})x_j\leq e_i-\underline{e}_i, \qquad (21)$$

¿

$\{$¿$\{$¿¿¿

¿

where the first, second, and third constraints impose the constraints in terms of mean, right and left spread respectively. This means that one fuzzy constraint in principle should be transformed into three crisp constraints.

To enforce the right spread constraint in (21), we can combine the first and second constraints in (21) and obtain

$$\sum_j \overline{d}_{ij}x_j\leq\overline{e}_i \qquad (22)$$

To enforce the left spread constraint in (21), we can combine the first and third constraints in (21) and obtain

$$\sum_j \underline{d}_{ij}x_j\geq\underline{e}_i\beta_i,\ \ 0<\beta_i\leq 1, \qquad (23)$$

where β_i is a scaling factor.

However, simultaneously enforcing the three constraints in (21) may cause inconsistence; especially the third constraint may conflict with the first two because the third constraint imposes the restriction in the opposite direction of the first two (22). For the current charge problem, we can choose not to enforce the spread constraints, which we will discuss a little bit later.

If the constraint (20) is an equality constraint, it can be weakened to $\sum_j \tilde{d}_{ij}x_i\subseteq\tilde{e}_i$ in the sense of Zadeh (Dubois & Prade 1980). The crisp equivalence is similar to (21) with mean constraint (the first constraint) enforced as an equality constraint and the other two constraints remaining unchanged. Now we discuss the transformations of constraints (7) and (9) for the scrap charge problem.

In terms of constraints (7), when the right spread constraint (22) applies, it is in fact the chance constraint (19) with the aspiration level $\lambda_{ki}=1$. This is the strictest case of the constraint (19), implying that enforcing right spread constraints make the chance constraint lose the flexibility to control the upper limit of the element concentration based on different aspiration levels. Therefore, we choose not to enforce the right spread constraint. The left spread constraint for constraints (7) is used to enforce the lower limit of the element concentration. However, the lower limits are not critical. Moreover, enforcing the left spread constraints can only results in shrinking the feasible region and thus increasing the cost of raw materials. Therefore, the left spread constraints are not enforced either. Then constraints (7) can be transformed based on the mean constraints only.

$$\sum_{j\in J_0} b_i a_j p_{ji} x_{kj} \text{¿} m_k t_{ki}, \quad k\in K, \quad i\in I_1. \tag{24}$$

In terms of equality constraints (9), it is unnecessary to enforce the spread constraints because adding pure alloying elements does not introduce new uncertainty. That is, we only need to enforce the mean constraints $\sum_{j\in J} b_i a_j p_{ji} x_{kj}=m_k t_{ki}$. For the batch *k*, the product is up to the standard as long as the concentration for element *i* falls between the lower limit $\underline{t}_{ki}$ and upper limit $\bar{t}_{ki}$. Therefore, we can weaken the mean constraints to

$$\sum_{j\in J} b_i a_j p_{ji} x_{kj} \geq m_k \underline{t}_{ki}, \quad k\in K, \quad i\in I_1. \tag{25}$$

In practice, it is unnecessary to worry about the upper limit $m_k \bar{t}_{ki}$ because the high cost of pure alloying materials forces the consumption of the pure alloying materials to be minimized.

Finally, the crisp equivalence of the original fuzzy programming model (6)-(12) becomes the objective function (6) in conjunction with constraints (19), (24), (25) and (10)-(12). We can see that the number of the constraints in crisp model does not increase as compared with that in the original fuzzy model because we drop the unnecessary constraints for interpretation of strict tolerance based on the application context. The crisp equivalence bears some similarity with the model resulting from adding the safety margins for element concentrations for the raw materials (Lahdelma et al. 1999). For example, the interpretation of the chance constraints (19) is similar to the effect of adding safety margins for element concentrations for raw materials. However, differences exist between these two approaches. Adding safety margins is an approach to extending the deterministic model to accommodate uncertainty and the complicated penalty cost structure to punish the possible deviation of an intermediate target in EAF is kept in the extended model. The fuzzy model directly addresses the uncertainty and each constraint in the crisp equivalence has a clear link with the fuzzy constraint. Consequently, the crisp equivalence of the fuzzy model is more compact than the extended deterministic model based on adding safety margins because no intermediate target and penalty cost is introduced in the interpretation of fuzzy constraints. That is, fuzzy approach is more straightforward. The crisp equivalence of the fuzzy model is a standard LP model which can be solved by the standard LP solver.

5 Numerical experiments

We have tested our model with modified process data from a Finnish steel plant for stainless steel production. The aim of the experiments is to reproduce the products based on the crisp equivalent model using the available materials and evaluate the uncertainty of element concentration on failure risk (failure rate) and material cost. The stainless products contain five main elements: Iron (Fe), Manganese (Mn), Chromium (Cr), Nickel (Ni), and Molybdenum (Mo). The candidate raw materials contain at least one of above five elements. To guarantee the existence of a feasible solution, we assume that the supply of pure materials that consist of only one of above five elements is unlimited. The concentrations of alloying elements for raw materials are represented by the mean and standard deviation. The concentrations of alloying elements for final products are specified by lower and upper limits. Table 1 gives the concentrations of alloying elements for some raw materials and Table 2 gives the concentrations of alloying elements for some final products.

Table 1 Concentrations of alloying elements for some raw materials

Material	Element concentrations (%)			
	Mn	Cr	Ni	Mo
FeCr-1	0.6±0	41±0	4.1±0	0±0
FeNi	0±0	5±0	30±0	0±0
Ferrite -scrap	1±0.27	12±1.9	1±0.14	0±0
LC-scrap	0±0	2±0.34	3±0.4	0±0
Acid-proof scrap	2±0.54	16± 2.57	11± 1.43	2±0.07
Scandust scrap	2±0.54	22± 3.67	6±0.77	1±0.04

Table 2 Concentrations of alloying elements for some final products

Product	Element concentrations (%)			
	Mn	Cr	Ni	Mo
710-2	[1,1.4]	[16.5,16.9]	[6.35,6.55]	[0.65,0.8]
720-1	[1.6,1.8]	[18,18.4]	[8.5,8.65]	[0,0.4]
720-4	[1.4,1.6]	[18, 18.4]	[10,10.2]	[0,0.4]
731-1	[1.6,1.8]	[17,17.4]	[9,9.15]	[0,0.4]
750-1	[1.4,1.6]	[16.7,17.1]	[11,11.15]	[2,2.2]
757-2	[1.5,1.8]	[16.6,17]	[10.5,10.7]	[2.5,2.7]

Next, we investigate failure risk and material cost based on stochastic simulation. The specific procedures are given below.

First, the fuzzy parameters such as the element concentrations for products and raw materials are realized. The realization of the concentration of element i for product k, $\tilde{t}_{ki}=(t_{ki},\underline{t}_{ki},\bar{t}_{ki})$, is straightforward. $\underline{t}_{ki}$, and $\bar{t}_{ki}$ correspond to the lower and upper limits for element concentration specified in product standard respectively, and $t_{ki}=(\underline{t}_{ki}+\bar{t}_{ki})/2$. For the concentration of element i for raw material j, $\tilde{p}_{ji}=(p_{ji},\underline{p}_{ji},\overline{p}_{ji})$, we assume that element concentrations follow normal distribution and choose suitable confidence factors n_1 and n_2. Then $\tilde{p}_{ji}$ can be realized based on the mean and standard deviation of element concentrations for raw materials as discussed in Section 2.

Second, for a fixed aspiration level $\lambda_{ki}=\lambda$ for element i in product k, we solve the crisp equivalence of the fuzzy model presented in Section 4 and obtain the selection of raw materials and related amount for producing a set of products.

Finally, stochastic simulations are performed. Random element concentrations for scrap are generated based on normal distribution. Then based on the material selection and related amount in the second step, the contribution of the scrap to the alloying elements in the product is computed. If the scrap contribution has already exceeded the upper limit of the product standard, then the failure is recorded. Otherwise, then amount of pure alloying materials is determined and the cost of raw materials is computed.

We reproduce 11 products using 17 raw materials (13 scraps + 4 pure alloying materials). We test two scenarios with different uncertainty degrees. The uncertainty degree is represented by the ratio of mean and standard deviation of the element concentrations. In the first scenario (S1), the maximum uncertainty degree among all the element concentrations in all materials is moderate and about 30%. In the second scenario (S2), the uncertainty degree for all the element concentration in all materials is doubled as compared with that of the first one. That is, the mean of each element concentration is same as that of the first scenario while the standard deviation of each element concentration is doubled as compared with that of the first scenario. We choose 6 aspiration levels and set confidence factors to 3. For each aspiration level in each scenario, ten turns of simulations are run. In each turn, 10^6 sets of random element concentrations are generated for the selected materials and then the failure rate and material cost is computed based on 10^6 outcomes. Then we aggregate the results from ten turns of simulation to obtain average failure rate and material costs. Table 3 gives the average failure rate (FR) and normalized cost (NC) for all of raw materials. The normalized cost uses the cost of the total raw materials for aspiration level λ=1 in the first scenario as reference.

Table 3 Average failure rate (FR, percentage) and normalized cost (NC, percentage) for two scenarios at different aspiration levels.

λ	S1		S2	
	FR	NC	FR	NC
0.75	21.96	98.49	24.45	100.61
0.8	11.81	98.63	10.53	101.08
0.85	5.41	98.84	4.71	101.52
0.9	2.20	99.17	1.80	101.92
0.95	0.76	99.58	0.61	102.29
1	0.10	100	0.25	102.63

Based on Table 3, the failure rate can be controlled by the proper combination of confidence factors (n_2) and aspiration levels (λ) regardless of the uncertainty degree. Here we can induce that the failure rate is unacceptable when λ<0.5 because it is too large. Therefore, the interpretation of chance constraints directly based on the possibility measure (16) is not sufficient in this context. Based on formula (19) and the representation of fuzzy number, (2λ-1)n_2 can be interpreted as the confidence factors for the aggregated element concentrations for raw materials. The choice of n_2 is associated with the uncertainty degree for element concentration and the selection of raw materials. For scenario S1, the minimum failure rate is 0.25% for n_2 = 3. If we want to decrease the failure rate further, we should increase n_2 for representing the fuzzy number. For the same (2λ-1)n_2, the failure rate is of the same order regardless of the uncertainty degree. That means, (2λ-1)n_2 plays a central role in controlling failure rate.

In terms of material cost, on the average we can see that costs increase by about 1.15% from failure rate about 5% (aspiration level 0.85) to less than 0.3% (aspiration level 1) for both scenarios. When the uncertainty degree doubles (from S1 to S2), the raw material costs increase by about 2.5% for the same order of failure rate. That is, there is a tradeoff between the decrease of the failure rate and increase of material cost. When the uncertainty degree increases, the material cost need to increase to realize the same order of failure rate. When the failure rate needs to decrease for the same scenario, the material cost also needs to increase. We know that the material cost accounts for the major cost for steel production based on the recycled scrap. If the material cost involves a large monetary value, even 0.5% cost variation is significant. The loss incurred by failure is up to the production process and the degree of violations. For slight violation, the failure may be recovered easily with moderate cost. For severe failure, the current heat becomes a waste and the production needs to be restarted. The decision about the tradeoff between material cost and failure rate is up to the decision maker and the production process.

Table 4 Cost share (CS) of pure alloying materials and the number of different types of scraps used for different scenarios at different aspiration levels.

λ	S1		S2	
	CS	Scraps	CS	Scraps
0.75	28.69	9	30.97	5

0.8	29.02	9	31.59	5
0.85	29.40	8	32.12	5
0.9	29.68	6	32.57	5
0.95	30.02	5	32.96	5
1	30.55	5	33.30	5

Finally we discuss the mechanism of controlling the failure rate. The model reallocates the materials based on the aspiration level. The reallocation includes the increase/decrease of scrap types and adjustment of the amount used for each material. Table 4 shows cost share (CS) of pure alloying materials and the number of different types of scraps selected for different scenarios at different aspiration levels. On average we can see that cost share for the pure alloying materials increases by about 1.15% from failure rate about 5% (aspiration level 0.85) to less than 0.3% (aspiration level 1) for both scenarios. When the uncertainty degree doubles (from S1 to S2) the cost share increases by about 2.5% for the same order of failure rate. Here the cost share variation for the pure alloying materials is coincident with the overall material cost variation shown in Table 3. This implies that the material cost increase for reducing the failure rate mainly attributes to the share increase of the pure alloying materials. Based on the number of scrap types selected for products, we can see the material selection can be widened as the uncertainty degree decreases.

The mechanism to control the failure rate is to increase the share of pure alloying materials in a cost-efficient way. For the same scenario, when the aspiration level decreases, the decrease in cost share of alloying materials means the increase in share of the scrap. This in turn means the overall uncertainty for the selected material increases. This results in simultaneous increase of the failure rate and decrease of the material cost because scrap is much cheaper than pure alloying material. For different scenarios, as the uncertainty degree increases, the share of the pure alloying materials must increase to realize the same the order of failure rate.

Above simulation results can provide several guidelines for the practical production. First, for the rough estimates of the element concentrations, a little bit larger uncertainty degree estimation can be used to produce the initial selection of scrap mix with less failure risk. Second, the scrap mix can be reselected with less material cost based on more precise estimation for element concentration as more process data are available. Finally, the scrap selection based on more reliable concentration estimation from large quantities of process data can be used to guide the material purchase.

6 Conclusions

Handling the scrap is one of the most important functions for scrap-based steelmaking. Using the wrong grades of scrap will result in higher raw material costs and difficulty in meeting product standard. The objective of the scrap charge optimization is to select the most economical scrap mix for each produced steel grade while minimizing the failure risk caused by uncertainties in raw material consistence. In this paper, we first presented the fuzzy chance constrained model for the scrap charge optimization based on product recipe and then transformed it into a crisp model based on the application context for solution. Simulation shows that the failure risk can be hedged by the proper combination of the aspiration levels and confidence factors for representing the fuzzy number. The mechanism of controlling the failure risk against uncertainty is to increase the share of the expensive pure alloying materials in the controlled manner. Therefore, there is a tradeoff between controlling failure risk and increase of the material cost. The model can be used for both determining the scrap mix for short-term operation of the melting shops based on available materials and for guiding the material purchase for long-term planning. The model can also be applied in other scrap-based production processes such as aluminum and copper (Lahdelma 1998).

References

AFS,1986. Least cost charge. *American Foundrymen's Society Transactions*.

Banks J. & Carson J.S., 1984. Discrete-event system simulation. Prentice-Hall Inc., Englewood Cliffs, New Jersy.

Bliss, N. G., 1997. Advances in Scrap Charge Optimization. *American Foundrymen's Society Transactions* 105, 27-30.

Canz T., 1996. Fuzzy linear programming in DSS for energy system planning. Working paper WP-96-132, International Institute for Applied System Analysis, Austria.

Dubois D. & Prade H., 1980. Systems of Linear fuzzy constraints. *Fuzzy Sets and Systems* 3, 37-48.

EUROFER (European Confederation of Iron and Steel Industries), 2001. Annual report. http://www.eurofer.org/publications/pdf/2001-AnnualReport.pdf

Ikäheimo J., 1999. Adaptive raw material optimization system for a steel plant. Masters's thesis. Helsinki University of Technology, Systems Analysis Laboratory, Espoo, Finland.

Kall P., Wallace S.W., 1994. Stochastic programming. John Wiley & Sons, Chichester.

Kim J., Lewis R.L., 1987. A Large Scale Linear Programming Application to Least Cost Charging for Foundry Melting Operations. *American Foundrymen's Society Transactions* 87-123, 735-744.

Lahdelma R., 1998. AMRO - adaptive metallurgical raw material optimization. Technology Programme SULA 2 Energy in Steel and Base Metal Production, Final Report, TEKES, Helsinki, 195-202.

Lahdelma R., Hakonen H., Ikäheimo J., 1999. AMRO - Adaptive metallurgical raw material optimization, IFORS'99, Beijing, China 16-20 August, 1999.

Liu B., Iwamura K., 1998. Chance constrained programming with fuzzy parameters. *Fuzzy Sets and Systems* 94, 227-237.

Rautaruukki, 2001. Steel for future. Sustainability report 2000-2001. http://www.bl.uk/pdf/eis/rautaruukki2001is.pdf.

Sandberg E., 2005. Energy and scrap optimization of electric arc furnace by statistical analysis of process data. Licentiate thesis, Division of Process Metallurgy, Luleå University of Technology, Sweden.

Slowinski R., 1986. A multi-criteria fuzzy linear programming method for water supply system development planning. *Fuzzy Sets and Systems* 19, 217-237.

Taha H. A., 1992. Operations research, New York: Macmillan.

Turunen J., 2000.Raw material optimization for copper foundry(In Finnish). Masters's thesis. Helsinki University of Technology, Systems Analysis Laboratory, Espoo, Finland.

Wang C., Wang D., Ip W.H., Yuen D.W., 2002. The single machine ready time scheduling problem with fuzzy processing times. *Fuzzy Sets and Systems* 127, 117-129

Watanabe T., Ellis H., 1994. A joint chance constrained programming model with row dependence. *European Journal of Operational Research* 77, 325-343.

Werners B., 1987. Interactive multiple objective programming subject to flexible constraints. *European Journal of Operational Research* 31, 342-349.

Wilson E., Kan M., Mirle A., 2001. Intelligent technologies for electric arc furnace optimization. ISS Technical paper.

Zadeh L.A., 1978. Fuzzy sets a basis for a theory of possibility. *Fuzzy Sets and Systems* 1, 3-28.

Zimmermann H.-J., 1978. Fuzzy programming and linear programming with several objective functions. *Fuzzy Sets and Systems* 1, 45-55.

Simulating processes of language emergence, communication and agent modeling

Timo Honkela, Ville Könönen, Tiina Lindh-Knuutila and Mari-Sanna Paukkeri
Adaptive Informatics Research Centre
Helsinki University of Technology, Finland
firstname.lastname@tkk.fi

Abstract

We discuss two different approaches for modeling other agents in multiagent systems. One approach is based on language between agents and modeling their cognitive processes. Another approach utilizes game theory and is based on modeling utilities of other agents. In both cases, we discuss how different machine learning paradigms can be utilized for acquiring experience from the environment and other agents.

1 Introduction

In this paper, we discuss how the emergence of subjective models of the world can be simulated using different approaches in learning and what is the role of communication and language. We consider, in particular, the role of unsupervised learning in the formation of agents' conceptual models, the original subjectivity of these models, the communication and learning processes that lead into intersubjective sharing of concepts, and a game theoretical approach to multiagent systems including reinforcement learning processes.

An intelligent agent usually has a purpose or a goal, probably set by the designer of the agent, and the agent tries to act rationally for satisfying the goal. For making rational decisions, the agent has to model its environment, perhaps including other agents. In the following, we consider some specific issues related to the modeling.

1.1 Complex and non-stationary environments

In the easiest case, the environment in which the agent is located is static, i.e. all properties of the environment remain constant for the whole life-time of the agent. However, often the situation is not so simple. The properties of the environment may vary with time, i.e. the environment is called *non-stationary*. The non-stationarity may be due to the environment itself or the limited resources of the agent. For example, in many real problem instances, the learning agent is not capable to sense the real state of the environment and thus some relevant properties of the environment remain hidden.

Another example of non-stationarity are multiagent systems. In these systems, properties of the environment for each individual agent depend on the actions of all agents located in the real environment. Thus for acting rationally, agents should model also other agents in the system.

Pfeifer and Scheier (1999) stress that the behavior of an agent is always the result of system-environment interaction. It cannot be explained on the basis of internal mechanisms only. They illustrate that the complexity that we as observers attribute to a particular behavior does not always indicate accurately the complexity of the underlying mechanisms. Experiments with very simple robots that merely react to stimuli in their environment have shown that rather complex behavior can emerge.

1.2 Subjective versus intersubjective

Moore and Carling (1988) state that "[l]anguages are in some respect like maps. If each of us sees the world from our particular perspective, then an individual's language is, in a sense, like a map of their world. Trying to understand another person is like trying to read a map, their map, a map of the world from their perspective." In many computational approaches to semantics and conceptual modeling, an objective point of view has been used: It is assumed that all the agents have a shared understanding and representation of the domain of discourse. However, Moore and Carling's statement emphasizes the need for explicit modeling of the other's point of view.

This requires modeling of the subjective use of language based on examples, and, furthermore, to model intersubjectivity, i.e. to have a model of the contents of other subjective models.

As an example related to the vocabulary problem, two persons may have different conceptual or terminological "density" of the topic under consideration. A layman, for instance, is likely to describe a phenomenon in general terms whereas an expert uses more specific terms.

1.3 Learning agents

Different machine learning techniques can be utilized for adding adaptivity to agent-based systems. There are basically three major learning paradigms in machine learning: *supervised learning*, *unsupervised learning* and *reinforcement learning*. In supervised learning, there exists a teacher having knowledge of the environment, in the form of input-output pairs, and the learning system for which the environment is unknown. The teacher provides samples from the environment by giving correct outputs to inputs and the goal of the learning system is to learn to emulate the teacher and to generalize the samples to unseen data. In unsupervised learning, contrary to supervised learning, there exists no external teacher and therefore no correct outputs are provided. Reinforcement learning lies between supervised and unsupervised learning: Correct answers are not provided directly to the learning system but the features of the environment are learned by continuously interacting with it. The learning system takes actions in the environment and receives reward signals from the environment corresponding to these action selections.

2 Unsupervised learning of conceptual systems

In the following, we consider the unsupervised learning of conceptual systems. We first study the modeling an individual agent that learns to create a conceptual space of its own and learns to associate words and expressions with conceptual space. The notion of *conceptual space* is taken from Gärdenfors who also presents the basic motivation and framework for dealing with conceptual spaces (Gärdenfors, 2000). After considering one individual agent, we consider a multiagent system in which a shared conceptual system is formed in a self-organized manner.

2.1 One agent

The *Self-Organizing Map (SOM)* (Kohonen, 1982, 2001) is an unsupervised learning model which is often considered as an artificial neural network model, especially of the experimentally found ordered "maps" in the cortex. The SOM can be used as a central component of a simulation model in which an agent learns a conceptual space, e.g. based on data in which words are "experienced" in their due contexts (Ritter and Kohonen, 1989). This approach has been used, for instance, to analyze the collection of Grimm fairy tales. In the resulting map, there are areas of categories such as verbs and nouns. Within the area of nouns a distinction between animate and inanimate nouns emerges (Honkela et al., 1995). The basic idea is that an agent is able to form autonomously a conceptual mapping of the input based on the input itself.

2.2 Community of agents

The basic approach how autonomous agents could learn to communicate and form an internal model of the environment applying self-organizing map algorithm was introduced, in a simple form, in (Honkela, 1993). Later we developed further the framework that would enable modeling the degree of conceptual autonomy of natural and artificial agents (Honkela et al., 2003). The basic claim was that the aspects related to learning and communication necessitate adaptive agents that are partially autonomous. We demonstrated how the partial conceptual autonomy can be obtained through a self-organization process. The input for the agents consists of perceptions of the environment, expressions communicated by other agents as well as the recognized identities of other agents (Honkela et al., 2003). A preliminary implementation of a simulated community of communicating agents based on these ideas did not succeed to fully demonstrate the emergence of a shared language (Honkela and Winter, 2003).

When language games (Wittgenstein, 1953) was included in the simulation model, it resulted in a simple language emerging in a population of communicating autonomous agents (Lindh-Knuutila, 2005). In this population, each agent was able to create their own associations between the conceptual level and the emerged words, although each agent had a slightly different conceptual representation of the world. The learning paradigm for the conceptual learning was fundamentally unsupervised, but the language learning tested has been so far supervised, i.e. the communicating agents are provided feedback of the outcome of the game as well as the "right answer". The reinforcement learning and the unsupervised learning models for language games remain to be implemented.

3 Game theoretical approach for multiagent systems

As discussed in Section 1, an intelligent agent usually has a goal. For studying agent based systems formally, e.g. for developing learning algorithms for intelligent agents, it is useful that the goal can be expressed mathematically. Traditionally approach is to define an *utility function* for the agent, i.e. there is a scalar value connected to each possible action measuring the fitness of the action choice for satisfying the goal of the agent. The utility function is often initially unknown and must be learned by interacting with the environment. Machine learning techniques can be utilized in this learning process, consider e.g. (Könönen, 2004b; Könönen and Oja, 2004; Könönen, 2004a).

3.1 One agent

In single-agent systems, achieving the rational behavior is simple: the agent always selects an action with the highest utility value. There exists a vast number of learning methods that utilize, in one way or another, the rational decision making in agent-based systems.

3.2 Community of agents

The branch of science studying decision making in single-agent systems is called *decision theory*. However, when there exists multiple active decision makers (agents) in the same environment, decision theory is not suitable for achieving rational behavior any more. *Game theory* is an extension of decision theory to multiagent systems. In game theory, agents explicitly model the dependency of their utility functions on the actions of all agents in the system. The goal of the agents is to find *equilibrium actions*, i.e. the actions that maximize their utility values assuming the all agents will use the same equilibrium. Jäger (2006) applies game theory to examine the shape formation in the conceptual spaces of the agents (Gärdenfors, 2000).

Theoretically, game theory provides a perfect tool for modeling multiagent systems. In practice, there are many problems with game theory. For example there can exist multiple equilibria and the agents should coordinate which one they will select. For calculating an equilibrium, the agents should not only model their own utility function but also the functions of all other agents in the system. This is often intractable and therefore some other methods, e.g. Bayesian techniques, should be used for creating more coarser models of other agents. By using these "external" models, the game theoretical problem reduces to the simpler problem solvable by using decision theory.

4 Discussion

Language does not need to be viewed plainly as a means for labeling the world but as an instrument by which the society and the individuals within it construct a model of the world. The world is continuous and changing. Thus, the language is a medium of abstraction rather than a tool for creation and mediation of an accurate "picture" of the world. The point of view chosen is always subject to some criteria of relevance or usefulness. This is not only true for the individual expressions that are used in communication but also concerns the creation or emergence of conceptual systems. It makes sense to make such distinctions in a language that are useful in one way another.

In this paper, we have discussed the role of unsupervised learning in the formation of agents' conceptual models in a non-stationary environment and a game theoretical approach for multiagent systems. In the future, we will study game theoretical models for language games and how to apply machine learning approaches for solving these games.

References

Peter Gärdenfors. *Conceptual Spaces*. MIT Press, 2000.

Timo Honkela. Neural nets that discuss: a general model of communication based on self-organizing maps. In S. Gielen and B. Kappen, editors, *Proceedings of ICANN'93, International Conference on Artificial Neural Networks*, pages 408–411, Amsterdam, the Netherlands, September 1993. Springer-Verlag, London.

Timo Honkela and Juha Winter. Simulating language learning in community of agents using self-organizing maps. Technical Report A71, Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo, Finland, 2003.

Timo Honkela, Ville Pulkki, and Teuvo Kohonen. Contextual relations of words in Grimm tales analyzed by self-organizing map. In *Proceedings of ICANN-95, International Conference on Artificial Neural Networks*, volume 2, pages 3–7. EC2 et Cie, 1995.

Timo Honkela, Kevin I. Hynnä, and Tarja Knuuttila. Framework for modeling partial conceptual autonomy of adaptive and communicating agents. In *Proceedings of Cognitive Science Conference*, 2003.

Gerhard Jäger. Convex meanings and evolutionary stability. In *Proceedings of 6th Int. Conf. on the Evolution of Language*, 2006.

Teuvo Kohonen. *Self-Organizing Maps*. Springer, Berlin, Heidelberg, 2001.

Teuvo Kohonen. Self-organizing formation of topologically correct feature maps. *Biological Cybernetics*, 43(1):59–69, 1982.

Ville J. Könönen. Policy gradient method for team Markov games. In *Proceedings of the Fifth International Conference on Intelligent Data Engineering and Automated Learning (IDEAL-2004)*, pages 733–739, Exeter, UK, 2004a.

Ville J. Könönen. Asymmetric multiagent reinforcement learning. *Web Intelligence and Agent Systems: An International Journal (WIAS)*, 2(2):105–121, 2004b.

Ville J. Könönen and Erkki Oja. Asymmetric multiagent reinforcement learning in pricing applications. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN-2004)*, pages 1097–1102, Budapest, Hungary, 2004.

Tiina Lindh-Knuutila. Simulating the emergence of a shared conceptual system in a multi-agent environment. Master's thesis, Helsinki University of Technology, Laboratory of Computer and Information Science, 2005.

Terence Moore and Chris Carling. *The Limitations of Language*. Macmillan Press, Houndmills, 1988.

Rolf Pfeifer and Christian Scheier. *Understanding Intelligence*. MIT Press, Cambridge, MA, 1999.

Helge Ritter and Teuvo Kohonen. Self-organizing semantic maps. *Biological Cybernetics*, 61(4):241–254, 1989.

Ludwig Wittgenstein. *Philosophical Investigations*. Blackwell Publishers, Oxford, 1953.

Program Comprehension Theories and Prolog based Methodologies

Erkki Laitila

A Department of Mathematical Information Technology, SwMaster Oy
University of Jyväskylä, P.O. Box 35,
FIN-40014 Jyväskylä Sääksmäentie 14, 40520 JKL
erkki.laitila@swmaster.fi

Abstract

Software maintenance is said to account for more than 50 % of all software efforts. Of this the attempts to understand the code can take 70 %. In spite of its importance, program comprehension is not well understood. This paper tells how Prolog can be used in modeling source code. An essentially new method, symbolic analysis, is presented and compared to static and dynamic analyses, which form the bulk of the current practices . We will show how a multi-paradigm tool, Visual Prolog, can serve as an excellent model formulation tool in describing complex source code to cover typical program understanding models

1 Introduction

Programmers are frequently struggling with program comprehension (PC) tasks when planning and preparing changes into code. A typical maintenance task is a confirmed change request. Before the change can safely be implemented, the code must at first be understood (von Mayrhauser 97).

Although a number of new technologies have been created since the 70's, writing object oriented source code is not easy. Bezevin has described development difficulties in his illustratively (05). An important theme in his document is the impedance mismatch between business and software. Problems writing code the problems in understanding it because of the mismatch.

A number of theories have been written for understanding procedural programs (Pennington 87) and a few for understanding object-oriented programs (Burlington 02). The integrated mental model is a well-established framework with three approaches: 1) top-down model, 2) bottom-up model and 3) situation model that refers to domain information, too (von Mayrhauser 97).

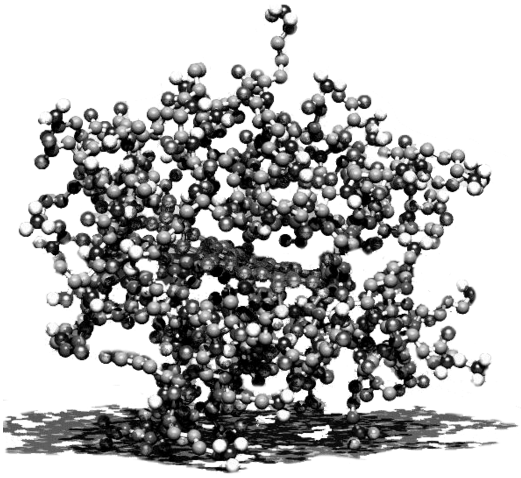


Figure 1: An analogy to program bindings.

Figure 1 illustrates molecular structures with bindings. It resembles source code in that both it also has bindings. The most important point in software is to understand how to match different approaches like domain information and program flows, and plans behind them. Object-oriented code causes some problems of its own, because invocations are hidden in modules. Further, dynamic bindings of objects are not visible, they must be traced step-by-step by reading the code.

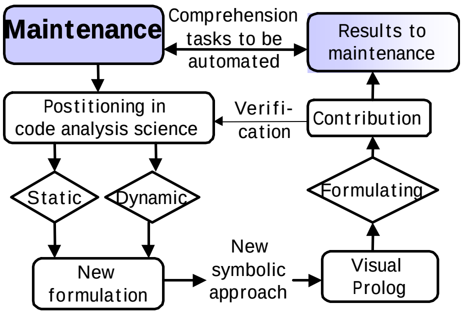


Figure 2. Symbolic approach.

We will build a unified concept framework starting from maintenance (Figure 2) to connect existing comprehension requirements applying strong reductionism by using Visual Prolog in the formulation. This new symbolic approach, or rather a symbolic model, is comparable with static and dynamic analyses.

This paper is organized as follows. Section 2 describes related work while Section 3 considers the selected approach. Section 4 presents a layered architecture of a symbolic model for tools. Section 5 contains the essential definitions. Section 6 presents a small Java example and Section 7 the architecture shortly. Section 8 describes a tool named JavaMaster and Section 9 Visual Prolog understanding in general. Finally, Section 10 concludes the paper.

2 Source code models

Widely used methods for PC include reverse engineering, UML–diagrams, document browsing, and code exploration by editor. These methods provide static information. Dynamic behavior information is much harder to capture, because capturing it requires debugging the complete installation.

The focus of software science is on developing models. It has been argued that the paradigm of object-oriented programming is to be replaced by a paradigm of model engineering (Bezevin 2005). The main idea of model driven architectures (MDA 2005) is summarized as “everything is a model”. However, there are many modeling technologies and no consensus on how to use them. Some alternatives are grammarware, xml, and ontologies, including semantic web and MDA.

OMG has, as a non-scientific organization, presented the main requirements for a modeling evaluation framework (OMG 05):

- Traceability: data for the model should be traceable to its original source.
- Modularity: each model should be divided into smaller units.
- Transformability: the model should allow transformation for other purposes.
- Executability: the model and possibly the code should be executable for verification purposes.

The most detailed model, FAMIX (Famix 1999), was created in a large project. Unfortunately, it contains only a static part. Dynamic models have not been implemented to cover all the source code. One essential problem in describing dynamic models in class based architectures is how to express logic connections between elements. The best way, a class-oriented architecture, like Java or C++, can connect things is by creating an association class combined with two association end objects. This produces two excess class types and the correspondent objects. It makes modeling in MDA very complex.

3 Research focus

For the research we set a goal to define a unified concept to find a model between written code and its behavior in order to match these with user domain information (Figure 3). In this figure, the X-axis represents written code (what), the Y-axis its object-based behavior (when), and the Z-axis user concepts (how).

If we mark the syntax by the letter X, we can use the notation Y = f (X) to describe a control flow starting from any starting symbol X. A control flow contains a group of methods calling each other with dependencies included:

```
Y =  y(Start) = f1 • f2 •.˜˜ • Target.
```

3.1 Control flow is essential

The control flow y(Start) builds a semantic behavior model for a current situation (Section 5, later). By using this invocation vector as input, all interesting information relating to the referred methods can be collected from their statements transitively. This information, when filtered and grouped according to user options, builds a pragmatic high abstraction situation model (von Mayrhauser 97). It will create a new projection Z that is in fact a group of candidates that are explaining the selected use case. A candidate is any symbolic name written by the programmer, like a class, a method or a domain name such as bluetooth or a database symbol if these terms are mapped into a model dictionary.

As a result we will get a useful mapping between all the dimensions because the tool can visualize the results in many ways, for example resembling Figure 1.

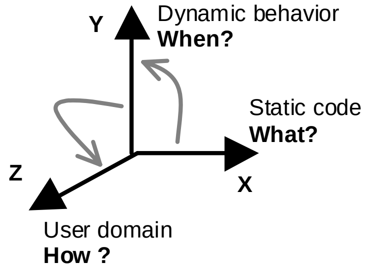


Figure 3. Reductionist research model.

3.2 Reasons for selections

Why Java was selected as a "patient", a language to be analyzed? It is a popular and much studied programming language but can suffer from comprehension problems. So showing what the research can contribute to Java, in respect to the problem pointed at, has essential value for the research.

Why Visual Prolog was selected as a "doctor", a language to implement the tool? The reason was that Java models contain a great amount of heavy logic and object-based information, so a multi-paradigm tool is needed. There are not many such tools.

4 Program Comprehension architecture

There are numerous design patterns for creating data models and for interactive systems. Because of its complex structures, the most essential implementation requirement for reverse engineering is layered architecture. For our work we have selected a combination of two existing architecture models: MDA and PCMEF.

MDA defines the following levels:

- M0 - source code files.
- M1 - parsed structures as classes.
- M2 - a model based on class definitions.
- M3 - a meta meta model where the most essential term is a definition element. It is especially suitable for transformation.

The second reference architecture, PCMEF, has the following practical structure:

- Foundation (F) is the base of the model, it defines the notation for information
- Entity (E) is the unit to make the model
- Mediator (M) is a facade to the model classes that refer to user interface.
- Controller (C) is a feature to control the actions and events of the tool.
- Presentation (P) is the display side of the user interface.

We combined these two methods to connect code and models into maintenance:

Table 1. Resulting Architecture Layers.

M0	Java files and packages in disk
M1	Parse trees as Java semantics
M2	Higher level notation, a symbolic language as a foundation (F)
M3	Complete symbolic model for the code with symbolic elements (E)
M4	Mediator (M) to master the model
M5	Controller (C) to implement messaging
M6	User interface for solving user hypotheses as model queries.
M7	Presentation (P) in visualizing maintenance task information.

5 Definitions for symbolic model

Traditional programming languages work numerically. Numeric evaluation means that the program calculates terms that are directly compatible with language types (Sun 03). Therefore it cannot handle unknown information, external symbols or interfaces. Contrary to this, using unknown external information, like a variable X, as a reference is not a problem in symbolic evaluation.

5.1 Symbolic analysis

Symbolic evaluation of code was first described by Cheatham et al (Cheatham 1979). We define symbolic analysis as a process to simulate original source code to obtain symbolic results for program comprehension purposes. The following list contains the mapping from the analysis into Visual Prolog:

1. Symbolic term is an individual Vip term.
2. Symbolic definition is a Vip clause to define relations between terms.
3. Symbolic evaluation is a task (goal or subgoal) to unify and solve the values for a symbolic definition.
4. Simulation is a process to execute successive symbolic evaluations for a query.

5. Symbolic result set is a set of outputs.
6. Symbolic presentation is a view to display symbolic results.

5.2 Source code

UML has defined a large framework for object-oriented development (Arlow, Neustadt 2002). It is too complex to be used for evaluation. So the smallest possible definition for Java information is needed for evaluation purposes in order to reach maximal throughput.

A simplified set for UML can be defined to be a Java information model. It contains the following terms: element, message, influence, block and result set. The main benefit of this simplified approach, compared with UML, is that symbolic information can be evaluated intensively. Furthermore, because the elements contain their type information, these can be translated into UML diagrams (class, state and sequence diagrams). This is an excellent tool for porting and verification purposes.

6 Java example

Figure 4 gives is a very small excerpt from the beginning of a typical Java program.

```
  class Server   {
     main(String[] args) {
     if (port == 0)
        port = Int.parseInt(args[0])
     new Server(port);
   }
 }
```

In exploring code, programmers use a short set of questions (Letovksy 86) and alternative approaches (Pennington 1987). The most typical questions being asked are: what, why and how. Let us have some examples:

- What does class Server do?
- Why did constructor Server start?
- How does it work?

The most useful way to start investigating this is to follow the main call tree (Fig 4).

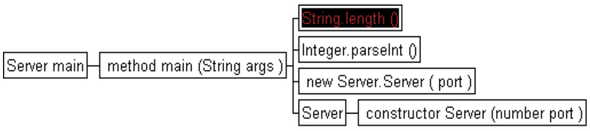

Server main
method main (String args)
String.length ()
Integer.parseInt ()
new Server.Server (port)
Server
constructor Server (number port)

Figure 4. Beginning of call tree for class Server.

The call tree is important for tracing other features, too, because it defines active code and causal orders. Some main benefits of symbolic analysis are partial evaluation, simulation of selected methods and indeterministic, exhaustive search of terms including simulation of dynamic bindings.

7 Implementation

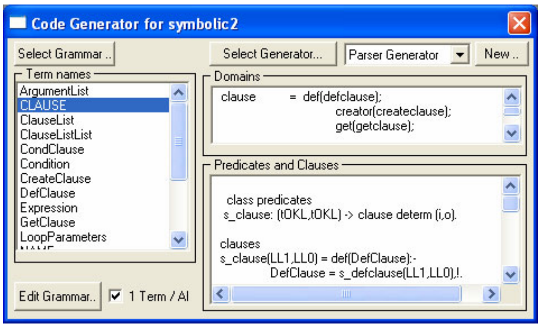
In this Section a data flow from code into user interface is described.

7.1 Layer M0: Language

In order to write a source code tool a grammar is needed. For Java there is an excellent specification made by Sun (Sun 2003). Grammars define syntax and any valid type of a parse tree for each program. The structure of a parse tree corresponds to the order in which different parts of the program are executed. Thus, grammars contribute to the definition of semantics (Webber 2005).

7.2 Grammar technology

PDC has created a grammar notation, GRM, and a parser generator for PDC Prolog described in the Toolbox (PDC Toolbox 1990). It is based on difference lists and top-down parsing. GRM-notation makes it possible to seamlessly combine syntax and wanted semantics of each term. Still, it is the responsibility/possibility of the user to define the semantics for parsing orders, for example, correctly (Laitila 1996).



```
clause       = def(defclause);
                  creator(createclause);
                  get(getclause);
```

```
 class predicates
s_clause: (tOKL,tOKL) -> clause determ (i,o).

clauses
s_clause(LL1,LL0) = def(DefClause):-
        DefClause = s_defclause(LL1,LL0),!.
```

Figure 5. SwToolFactory: Universal generator.

SwMaster Oy has developed a Visual Prolog – based tool named SwToolFactory (Fig 5). It is completely object-oriented and contains the following features.

- Class MetaGrammar contains common features for a formal grammar.

- Class MetaLanguage contains language dependant tools and features.
- Object MetaTerm instantiates grammar terms and features like a parser generator.
- There is a collection of numerous code generators like a pretty printer generator and generators for XML, AST etc.

In various reengineering projects for C, Cobol, VB and C++, SwMaster has created a unified concept framework (Figure 6) for formal languages as follows. For reading code the features needed are scanner (1), parser (2) and a saving element (3): either a model weaver or a simple symbol table. The code is stored into Visual Prolog as hierarchical facts or as symbolic objects (8).

The code information can be re-translated into syntax by using three features: a model iterator (4), a code generator (5) and a pretty printer (6). There are a number of mathematical correspondences between the parts of this “diamond”. A very interesting feature created by this structure is the possibility to make translations from one language into another by switching the terms in the input side (7) and the output side (8). By combining (7) and (8) from different languages we can form a term translator.

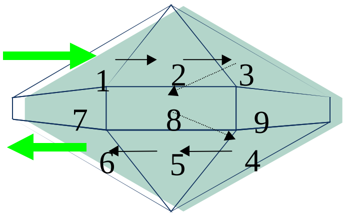


Figure 6. Diamond, a unified concept framework and programming model for formal languages.

This design concept is useful in many kinds of Visual Prolog transformation tasks. As an example of a code generator for a pretty printer is presented. The shortest Visual Prolog code for a pretty printer can be defined as a concatenated list of syntax terms and recursive invocations of lower level calls in the syntax order (below, *gen*-prefix is used). A recursive term generator *gen_Statement* for the *for*-statement of Java1.5 is shortly:

```
gen_Statement(
   for(ForControl,Statement)) =
    concatList(["for", "(",
    gen_ForControl(ForControl), ")",
    gen_Statement(Statement)]).
```

7.3 Defining semantics

For defining semantics of source code numerous principles have been developed including denotational semantics and attribute grammars. The best approach for Prolog is argued to be natural semantics (Sethi 1996), which gives us an excellent tool to investigate the validity of Prolog implementations. An *if*-command is a good example. In the GRM-grammar notation:

```
Statement =
  “if” Condition then Statement ->
    iff(Condition, Statement).
```

A Prolog based analyzer could provide a number of different ways to analyze *iff*-statements: subterm search, deterministic evaluation, code simulation, indeterministic coverage test, traversal of evaluating results, information for displays, model making etc (Heering 2005). Overall, natural semantics is a good way to document and to validate these sub-models.

7.4 Layer M1: Parse tree for Java

When GRM-notation is used, each term has a defined parse tree that is defined in the right side of the term. The start symbol term of Java3 is a compilation unit. The other main Java terms for analysis are class declaration, type declaration, statement, expression, identifier and literal.

7.5 Layer M2: Symbolic language

In a normal practice, parse trees are converted into Abstract Syntax Trees (AST) for analysis purposes (VMCAI - konferenssi 2005). However, ASTs suffer from many problems because of their strong connection with the original syntax. This is why we use another way to raise the abstraction level. We built a completely new language, Symbolic, instead of using Java itself. This simplifies programming and it makes it possible to optimize code items according to analysis needs. In the symbolic notation the nearest equivalent for Java statement is clause. Table 2 below describes categories, identification, purpose and principal use for the term clause.

Table 2. Categories for symbolic clauses

Id	Purpose	Principal use
def	Definition	Identification, types
crea	Creator	Dynamic bindings
set	Changes	Data flow
get	Invocation	Control flow
ref	Reference	Variable references
op	Operation	Math & relat operators
loop	Loop	Loop analysis
path	Condition	State analysis
val	Constant	Evaluating values
other	Non-core term	Terms, not important

We defined Symbolic to be an independent language, so that it has all the “diamond features” (Figure 6). It is a class with its domains, parsers and generators. Symbolic is particularly aimed at evaluation, display and explanation purposes.

For translation from Java into Symbolic each Java statement is converted into one or more symbolic clauses depending on the statement. So a symbolic clause is not backwards compatible, but when the context of successive symbolic clauses maintains a correct sequence, the clauses can be retranslated into Java with their original meaning. This is useful for program specialization purposes.

7.6 Symbolic translation

Vip6 has an excellent function notation for translation purposes. Below, a translation rule to translate a Java if-statement into a pathClause of Symbolic is shown. The predicate returns the term in another language. Condit is an abbreviation for condition and SL for statement list. Translation takes place at the Prolog level, and not by replacing source structures like traditional translators do when using complex external rules (TXL, Software Tech. Laboratory 2004).

```
statement2clause(iff(Condit,SL)) =
   pathClause(condition2cond(Condit),
   stmntList2clauseList(SL)).
```

The translation principle is excellent for the use of Vip applications, especially in multi-layer architectures. It is best to keep the translation programming apart from the syntax level, due to the complexities and peculiarities of the latter.

7.7 Capturing control flows

In Section 1 (Figure 1) it is argued that programs resemble molecular structures. In Section 6 a call tree was presented as a base for models. UML, which covers the vast majority of approaches for the software developer, contains a 4+1 model to summarize all diagrams (Arlow, Neustadt 2002). In order to identify all behavior features of UML, a behavior model is defined to be a sum of all causal chains (Mellor, Balcer 2002). These are captured by “chopping” successive method invocations (f) for the selected approach (Figure 7).

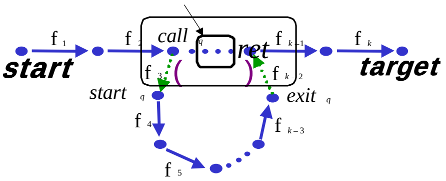


Figure 7. Control flow in chopping (Reps 2000).

The most used principle of source code analysis is called slicing (Weiser 1984). It means detecting effects of source code from a certain point. It is not very useful for program comprehension, because it is only a low-level approach. Chopping is a more general approach (Reps 2000). Chopping is considered in terms of producing relevant information regarding a part of the program between Start and Target as determined by the person attempting to understand the program. Chopping is essential because it can be used in cause-effect analysis which is particularly relevant in trouble-shooting (von Mayrhauser 1997). The purpose of the following rule is to solve the chopping equation (Figure 7):

```
Target = fk •. fk-1 •.... •.f (Start).
```

In a symbolic notation it can be either nested:

```
Target = f (f ( ... f( Start ) )  )
```

or successive:

```
Target= [f(XK) •f(XK-1).• f(“Start”)].
```

In Vip, collecting chops (f) could be written:

```
chopping(Target,[f(This)|Sequence]):-
  f(NextId, Arguments),
  NextId:chopping(Target, Sequence).
```

7.8 Layer M3: Object-oriented model

Initially the symbolic code for each class was stored inside classes. However, tree based structures are difficult to analyze, because 1) identifying elements is not easy and 2) there is no easy way to point to specified points to trees. Further, the user considers the code elements as parallel, not as hierarchical units. That is why an object-oriented core is needed for the symbolic model. In our case it is very natural to use the categories of Symbolic language as the base for model objects.

To implement an object-oriented model, a model weaver is needed. It divides the parse trees into smaller element objects as shown in Table 2. Let us consider a row X in Table 2. The purpose of the Symbolic Model Weaver is to create a *Symbolic <X> Element* for each X and to store the host into the parent element and the handle of the parent into the child element. Each block in the code is replaced by a link element and the replacement is stored into the parent as a link node. In accordance with the principle above, each element should have a minimum number of child elements. Because the elements of each object are in the same order as in the original code, it is guaranteed that elements can be evaluated correctly. This makes it possible to execute models, which is the most challenging feature of modeling technologies (Section 2). The principle uses, extensively, Prolog's logic notation and Visual Prolog's object notation together. In Figure 8 there is a class diagram illustrating a program implementation.

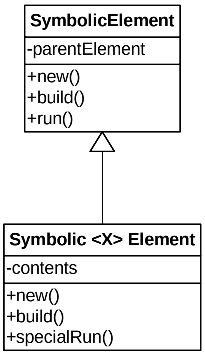


Figure 8. Principles of symbolic model elements.

The super class *SymbolicElement* inherited from the class *Symbolic,* contains the methods to build each sub-model, to run the sub-model and to traverse the results. There is a method *specialRun* that runs the contents of each element object.

7.9 Layer M4: Mediator

Class *Mediator* is a facade to combine the most essential data classes and activities. The Symbolic model is run via a class *ModelEngine* that has a run-command which starts a new thread.

It is said that hypothesis is the only driver for program understanding (von Mayrhauser 97).The user wants to investigate special parts of the software by questions such as: what, why, when and how. To specify a run, a start and target combination is entered by the user.

The results of the query are a collection of Symbolic clauses combined with the query notation. Each explanation has the form: Explanation = query + answer.

7.10 Layers M5, M6: Controller and UI

Controller receives signals from the model and updates displays according to those signals. UI is described later in Section 8.

7.11: Layer M7: Presentations

As stated earlier, each explanation contains the question and its handle and the results. By collecting explanations systematically the user can update his/her situation model, which can contain information like:

- What does class *Server* do?
- Why did X start Y?
- How does method Z work?

Symbolic clause is the foundation for all displayed data. In source code analysis theory there are about 20 different visualization graphs and presentation types (Famix 1999). Most of them can be transformed from the result set of the symbolic model for Java.

8. Tool JavaMaster

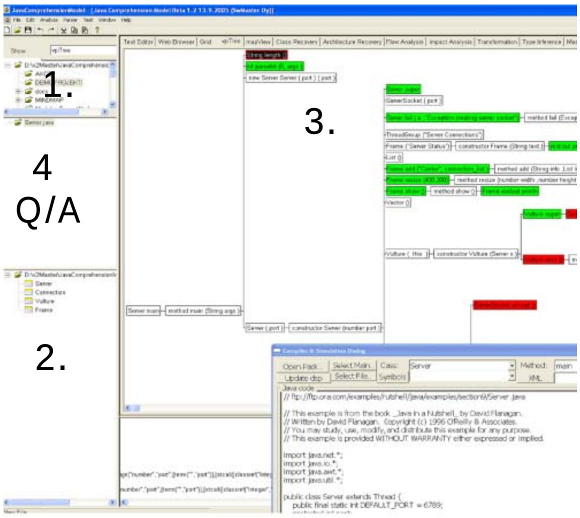
According to the description above, a tool was developed, the development of which started in 2005. The current implementation parses Java 1.5 without problems. The tool contains about 40.000 lines of code and 400 classes. Some statistics about the code:

1. Java 1.5 grammar file (300 lines)
2. Java parser (2556 lines)
3. Symbolic language (90 definition lines)
4. Java to Symbolic translator (2200 lines)

It is useful to note that the symbolic translator is shorter than Java parser. This shows that semantic translation is a very expressive feature in code transformation.

8.1 Tool features

Figure 10 shows the principles of the tool. The source code is loaded from the tree (1). It provides a program model for another tree (2). Then the user selects the function *Worklist*, for which the tool's response is to display a set of navigation questions in (4 Q/A). Program information is displayed by abstract controls (3). Here a relatively large call tree is shown. The user sets hypotheses and formulates queries related to the suggestions (Q) which will help the user in focusing on the most essential control-flows and objects in order to understand the critical references and method invocations (Wiedenbeck et al 2002).



```
// ftp://ftp.ora.com/examples/nutshell/java/examples/section9/Server.java

// This example is from the book _Java in a Nutshell_ by David Flanagan.
// Written by David Flanagan.  Copyright (c) 1996 O'Reilly & Associates.
// You may study, use, modify, and distribute this example for any purpose.
// This example is provided WITHOUT WARRANTY either expressed or implied.

import java.net.*;
import java.io.*;
import java.awt.*;
import java.util.*;

public class Server extends Thread {
    public final static int DEFAULT_PORT = 6789;
```

Figure 10. User interface of JavaMaster.

9 Prolog comprehension

Visual Prolog resembles Java in most information needs (Sections 3 and 5). It contains program flows, classes, objects and dependency models. Some essential differences are non-determinism, strong recursive nature, declarative syntax, and flow analysis. Further, public attributes are not allowed and predicates are forming much more compact units than in methods of Java. Most of these differences make understanding easier. From the programmers point-of-view it would be best not to use facts, because facts can cause side effects, but it is not always possible to avoid this.

Understanding side effects is essential in Prolog (Walker et al 1997). In Figure 11 there are two cases, A and B, using predicates *p*, *q* and *r*. Predicate p causes a side effect *note(X)* to case A but not to B. It has direct influence on predicate q but not on r. Still, if r is called after q then B is dependent of *note* indirectly.

Summary: The side effects can be detected completely by symbolic analysis because the phenomen in Figure 11 is a control flow.

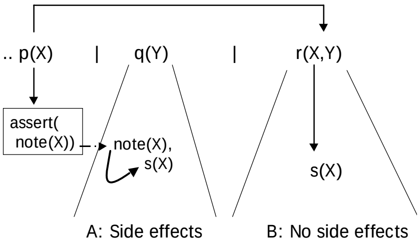


Figure 11. Prolog's side effect model.

9.1 Influence model

Each Prolog program can be thought as an active control flow, which has inputs and outputs on several levels. In Figure 12 inputs (callers) are up and outputs (callees) down. Pure logic, including temporary effects, is in the base line. A vertical distance for each influence type can be defined as follows: Com-components are the most distant lines as well as external actions that cannot be repeated. The most distant lines are main influences, compared with side effects that are intermediate data and not among the wanted results.

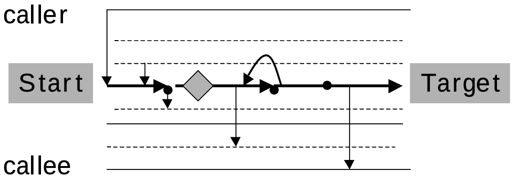


Figure 12. Prolog's influence model.

This model can be useful in planning refactoring for Vip classes.

9.2 Visual Prolog application

In order to understand a complete Vip application we can use the following main approaches (the related figure in parenthesis):

- Static: How has it been written? (F. 11)
- Dynamic: How does it behave? (F. 12)
- Symbolic: What are the explanations for candidates between starts and targets referring to our focus? (F. 10)

The symbolic approach is the most general one. It is pragmatic by nature. It could give us useful snapshots from the user's approach combining the dimensions of Figure 3!

10 Conclusions

A seamless architecture for a symbolic program comprehension methodology has been described in this paper. Using Visual Prolog proved to be a right choice. When one of the most characteristic features of traditional languages is that they are leading into more and more specific applications, Visual Prolog has, as shown in this paper in Sections 2..9, because of its declarative notation, another rather valuable feature; it is leading into wider, more abstract thinking, thus creating more general approaches and theories related to the software science.

Symbolic analysis can, because of its theory of grammars, semantic translations, symbolic models and explanation features, be an important step in future in getting closer to a grand unified theory of programming language understanding.

Acknowledgements

This research is a part of a dissertation concerning a symbolic program comprehension methodology. It is funded by COMAS of Jyväskylä University. SwMaster Ltd has enabled the initial research and product development for the tool since 2000.

References

Jean Bezevin. Object to Model Paradigm Change, <url:http://rangiroa.essi.fr/ cours/ systeme2/02-mda.pdf>

Jean-Marie Burkhardt, Francoise Detienne and Susan Wiedenbeck. "Object-Oriented Program Comprehension: Effect of Expertise, Task and Phase", Empirical Softw. Eng. 7 (2), 2002, pp. 115-156.

Thomas Cheatham, Glen Holloway and Judy Townley. "Symbolic Evaluation and the Analysis of Programs", IEEE Trans. Softw. Eng. 5 (4), 1979, pp. 402-417.

Serge Demeyer, Sander Tichelaar, Patrick Steyaert. FAMIX 2.0: The FAMOOS Information Exchange Model (1999), <uri: http://www.iam.unibe.ch/~famoos>

Jan Heering, Paul Clint. Semantics of Programming Languages, Tool Approach,<url: http://homepages.cwi.nl/~jan/semantics >

Erkki Laitila. Visual Prolog: Industrial Applications (only in Finnish). Jyväskylä, Finland: Teknolit, 1996. 228 p.

Stan Letovsky. "Cognitive process in program comprehension". In: E. Soloway & S. Iyengar (Ed.) Empirical Studies of Programmers: Norwood, NJ: Ablex, 1986.

MDA. The Architecture of Choice for a Changing World, Model Driven Architectures. <url: http://www.omg.org/mda/>

Leszek Maciaszek, Bruc Lee Liong. Practical Software Engineering, Addison-Wesley, 2005.

Stephen J. Mellor, Marc J. Balcer. Executable UML, Addison-Wesley,2002.

OMG. Object Management Group <url:www.omg.org>

Nancy Pennington. "Stimulus Structures and Mental Representations in Expert Comprehension of Computer Programs", Cognitive Psychology, 1987.

Thomas Reps. “Program Analysis via Graph Reachability”, PLDI-Conference, 2000.

Ravi Sethi. Programming Languages - Concepts and Constructs. Addison-Wesley, 1996.

Sun. Grammar of Java <url:http://java.sun.com/docs/books/jls/ third_edition>, 2003.

Software Technology Laboratory. The TXL Programming Language: www.txl.ca <url:www.txl.ca>. Technical report, 2004.

Prolog Development Center A/S. PDC Prolog Toolbox, <url:http://www.pdc.dk>,1990

Jim. Arlow, Ila Neustadt. UML and the Unified Process, Addison-Wesley, 2002.

Prolog Development Center A/S. Visual Prolog, 2006, <url:http://www.visualprolog.com>

Anne-Liese von Mayrhauser, Marie Vans. "Hypothesis-driven understanding processes during corrective maintenance of large scale software". CSM 1997, 1997, pp. 12-20. IEEE Computer Soc.

VMC05, Verification, Model Checking, and Abstract Interpretation Conference, Paris, 2005.

Adrian Walker, Michel McCord, John Sowa, Walter Wilson. Knowledge Systems and Prolog, 1987, Addison Wesley.

Adam Webber. Modern Programming Languages, c/o Amazon, 2005.

Marc Weiser. Program slicing. IEEE Transactions on Software Engineering, 10(4):352–357, 1984.

Norman Wilde,and Ross Huitt, "Maintenance Support for Object-Oriented Programs", IEEE Trans. Software Eng. 18 (12), 1992, pp. 1038-1044.

Norman Wilde, Allen Chapman, Paul Matthews, Ross Huitt, Describing Object Oriented Software: What Maintainers Need to Know, EDATS, In International Journal on Software Engineering, pages 521–533, 1994.

Describing Rich Content: Future Directions for the Semantic Web

Timo Honkela and Matti Pöllä
Helsinki University of Technology
Laboratory of Computer and Information Science
P.O. Box 5400 FI-02015 TKK, Finland
{timo.honkela, matti.polla}@tkk.fi

Abstract

In this position paper, we discuss some problems related to those semantic web methodologies that are straightforwardly based on predicate logic and related formalisms. We also discuss complementary and alternative approaches and provide some examples of such.

1 Introduction

It is clear that the use of standardized formats within computer science is beneficial. For instance, the widespread use of the World Wide Web would not have been possible without the adoption of HTML. Similarly, there are serious attempts to create standards for metadata, data about data, so that a piece of art stored in electronic form would include information about, e.g., its creator, source, identification and possible access restrictions. Moreover, metadata usually includes also a textual summary of the contents, a content description that provides information for the organization and search of the data.

1.1 Subjective and complex paths from data to metadata

Especially if pictorial or sound data is considered, an associated description, metadata, is useful. The description may be based on a pre-defined classification or framework, for instance, like an ontology within the current semantic web technologies. However, even if something like the identity of the author or the place of publishing can rather easily be determined unambiguously, the same is not true for the description of the contents. In the domain of information retrieval and databases of text documents, Furnas et al. (1987) already found that in spontaneous word choice for objects in five domains, two people favored the same term with less than 20% probability. Bates (1986) has shown that different indexers, well trained in an indexing scheme, might assign index terms for a given document differently. It has also been observed that an indexer might use different terms for the same document at different times. The meaning of an expression (queries, descriptions) in any domain is graded and changing, biased by the particular context. Fig. 1 aims to illustrate the challenging aspects of the overall situation. Human beings perceive, act and interact within complex environments. It has occasionally been assumed that much of the underlying conceptual structure within human mind is readily given by some way, even already when we are born. There is a vast body of literature related to this question which we do not touch upon here. It suffices to note that the discussion in this paper is based on the assumption that the conceptual systems are mainly emergent: they are created, molded and shared by individuals in interaction with each other and the rest of the accessible part of the world.

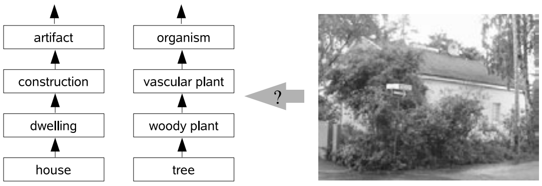


Figure 1: An illustration of the basic problem of symbol grounding: what is the process that is capable of generating ontological structures based on complex raw perceptions and activities.

Vygotsky (1986, originally published in 1934) has stated that "... the world of experience must be greatly simplified and generalized before it can be translated into symbols. Only in this way does communication become possible, for the individual's experience resides only in his own consciousness and is, strictly speaking, not communicable.” Later, he continues: “The relation of thought to word is not a thing but a process, a continual movement back and forth from thought to word and from word to thought. In that process the relation of thought to word undergoes changes which themselves may be regarded as development in the functional sense.” This means in practice that conceptualization is a complex process that takes place in a socio-cultural context, i.e., within a community of interacting individuals whose activities result into various kinds of cultural artifacts such as written texts.

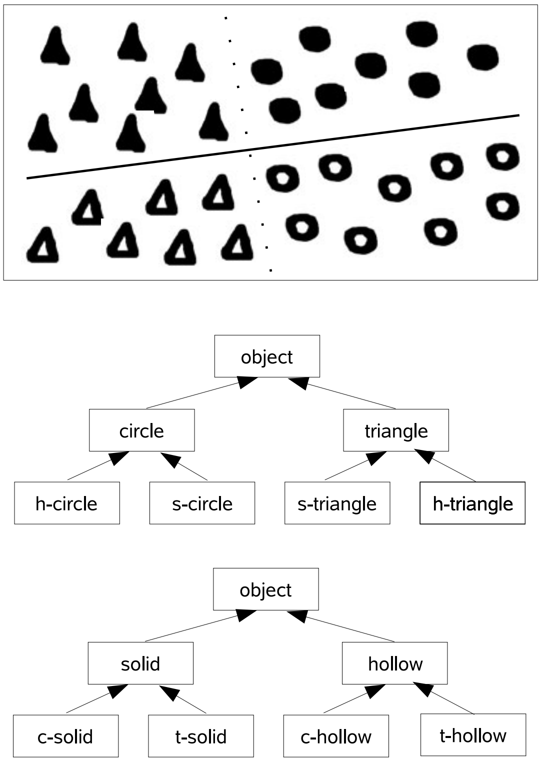


Figure 2: An illustration of two conflicting conceptual systems on the surface level. These systems prioritize the distinctive features differently.

It is a very basic problem in knowledge management that different words and phrases are used for expressing similar objects of interest. Natural languages are used for the communication between human beings, i.e., individuals with varying background, knowledge, and ways to express themselves. When rich contents are considered this phenomenon should be more than evident. Therefore, if the content description is based on a formalized and rigid framework of a classification system, problems are likely to arise. Fig. 2 shows a simple example of two conflicting formalizations.

Natural languages have evolved to have a certain degree of compositionality to deal with such situations. Vogt (2006) has developed a simulation model that considers the transition of holistic languages versus compositional languages. Holistic languages are languages in which parts of expressions have no functional relation to any parts of their meanings. It is to be noted, though, that the compositionality is a matter of degree in natural languages. One cannot assume that, for instance, each noun would refer to one concept and the conceptual structures would follow isomorphically the composition of the syntactical structures. For instance, the existence of collocations complicates the situation.

1.2 From two-valued logic to adaptive continuous-valued models

In addition to the different ways of creating conceptual hierarchies discussed above, the inherent continuous nature of many phenomena makes it impossible to determine exactly, in a shared manner the borderlines between some concepts or how some words are used. Actually, we prefer to consider concepts as areas in high-dimensional continuous spaces as suggested by Gärdenfors (2000). Fig. 3 illustrated the basic problem of dividing continuous spaces into discrete representations. Various approaches have been developed to deal with this phenomenon, fuzzy set theory as a primary example Zadeh (1965).

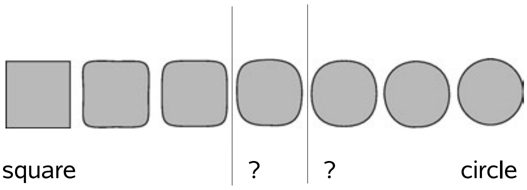


Figure 3: The obvious difference between continuous and discrete. The continuous can be discretized but there is a certain cost associated. The cost relates to the need to learn and deal with an increasing number of explicit symbols.

The basic semantic web formalisms are based on predicate logic and other symbolic representations and are subject to most of those problems that earlier AI formalisms have. In order to alleviate the problems related to the traditional approach, there are already examples of research projects in which some soft computing approaches, including fuzzy logic, probabilistic modeling and statistical machine learning, are applied. Even a collection named “Soft Computing in Ontologies and Semantic Web” has recently been published Ma (2006). In the collection, a related topic to the discussion above is presented by Holi and Hyvönen (2006). They consider modeling uncertainty in semantic web taxonomies in particular in the domain of geographical information. Nikravesh (2006) presents an approach which is based on the use of, e.g., fuzzy logic, evolutionary computation and the self-organizing map.

In this position paper, we outline some complementary and sometimes even alternative approaches to the core methodologies currently applied within the semantic web research and development.

2 Alternatives to highly formalized metadata

It may be useful not to define any artificial limitations for the descriptions. For instance, when the domain develops into directions which did not exist when the classification system was developed, problems arise.

Moreover, if the content it described using large enough body of text the for better recall, i.e., higher likelihood for finding the information is greater. However, tools for ensuring precision are needed. Precision refers to the number of relevant retrieved documents over the total number of retrieved documents.

If a word or an expression is seen without the context there are more possibilities for misunderstanding. Thus, for human reader the contextual information is often very beneficial. The same need for disambiguation can also be relevant within information systems. As the development of ontologies and other similar formalizations are, in practice, grounded in the individual understanding and experience of the developers and their socio-cultural context, the status of individual items in a symbolic description may be unclear. Fig. 4 illustrates the influence of varying socio-cultural contexts. Namely, if one considers the pragmatic meaning (if not semantic meaning) of “summer’s day” in a Shakespeare sonnet, it is obvious that there is a great difference between the contexts that are being referred to, for example, in Scandinavia and in Sahara.

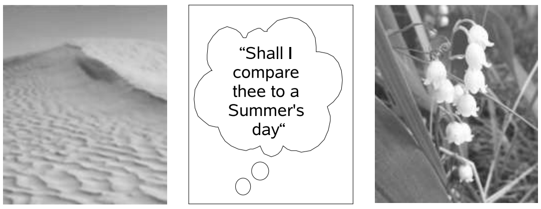


Figure 4: Different contexts potentially related to the expression given.

Similarly, the methods that are used to manage data should be able to deal with contextual information, or even in some cases provide context. The Self-Organizing Map by Kohonen (2001) can be considered as an example of such a method. Often it is even possible to find relevant features from the data itself Kohonen et al. (1997). However, a computerized method – using a some kind of autonomous agent – does not provide an “objective” classification of the data while any process of feature extraction, human or artificial, is based on some selections for which there most often are some well-grounded alternatives.

Below, we describe some examples in which the ideas and principles outlined above have been applied.

3 Case studies

We consider three cases which exemplify complementary and alternative approaches to the use of (first order) logic-based formalisms in knowledge representation.

3.1 Color labeling

The concept of color cannot be adequately studied only by considering the logico-semantic structure of color words. One has to take into account the color as a physical phenomenon. Color naming also requires consideration of the qualities of the human color perception system. A thorough study of color naming, thus, would require consideration of at least linguistic, cognitive, biological, physical and philosophical aspects Hardin (1988).

Subjective naming of color shades has been studied in a demonstrative web site where users can pick color shades and give name labels (or ’tags’) to the colors. In this case color shades are represented as a RGB tuple indicating the intensities of red, green and blue color.

As the database of tagged color shades grows, interesting properties can be found by looking at the distributions of individual tags. For example, the tag ’red’ would result in a reasonably narrow distribution centered around the point (1.0; 0.0; 0.0) in the RGB space. Other tags, however, can have much more variation in the way people place them in the color space.

For example, the tag ’skin’ or ’hair’ is an example of a highly subjective name for a color shade due to various skin and hair color conceptions. A related theme is the fact that the domain for which a color name is an attribute has a clear influence on which part of the RGB space the color name refers to. Gärdenfors (2000) lists as an example the differences between the quality of redness of skin, book, hair, wine or soil. Fig. 5 below gives an example how the color space can be divided in a substantially different manner in two languages.

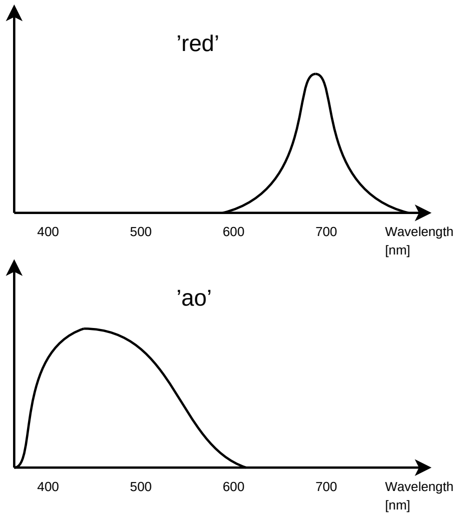


Figure 5: Illustration of fuzzy color definitions. The word ’red’ is usually referred to light wavelengths close to 700 nm while the Japanese term ’ao’ is associated with a wider scale of wavelengths. In English the term ’ao’ would cover both ’blue’ and ’green’.

3.2 WEBSOM and PicSOM

The WEBSOM method was developed to facilitate an automatic organization of text collections into visual and browsable document maps (Honkela et al., 1997; Lagus et al., 2004). Based on the Self-Organizing Map (SOM) algorithm (Kohonen, 2001), the system organizes documents into a two-dimensional plane in which two documents tend to be close to each other if their contents are similar. The similarity assessment is based on the full-text contents of the documents. In the original WEBSOM method (Honkela et al., 1996) the similarity assessment consisted of two phases. In the first phase, a word-category map (Ritter and Kohonen, 1989; Honkela et al., 1995) was formed to detect similarities of words based on the contexts in which they are used. The Latent Semantic Indexing (LSI) method (Deerwester et al., 1990) is nowadays often used for similar purpose. In the second phase, the document contents were mapped on the word-category map (WCM). The distribution of the words in a document over the WCM was used as the feature vector used as an input for the document SOM. Later, the WEBSOM method was streamlined to facilitate processing of very large document collections (Kohonen et al., 1999) and the use of the WCM as a preprocessing step was abandoned.

The PicSOM method (Laaksonen et al., 1999, 2002; Koskela et al., 2005) was developed for similar purposes than the WEBSOM method for content-based image retrieval, rather than for text retrieval. Also the PicSOM method is based on the Self-Organizing Map (SOM) algorithm (Kohonen, 2001). The SOM is used to organize images into map units in a two-dimensional grid so that similar images are located near each other. The PicSOM method brings three advanced features in comparison with the WEBSOM method. First, the PicSOM uses a tree-structured version of the SOM algorithm (Tree Structured Self-Organizing Map, TS-SOM) (Koikkalainen and Oja, 1990) to create a hierarchical representation of the image database. Second, the PicSOM system uses a combination of several types of statistical features. For the image contents, separate feature vectors have been formed for describing colors, textures, and shapes found in the images. A distinct TS-SOM is constructed for each feature vector set and these maps are used in parallel to select the returned images. Third, the retrieval process with the PicSOM system is an iterative process utilizing relevance feedback from the user. A retrieval session begins with an initial set of different images uniformly selected from the database. On subsequent rounds, the query focuses more accurately on the user’s needs based on their selections. This is achieved as the system learns the user’s preferences from the selections made on previous rounds.

WEBSOM and PicSOM methods are a means for content-driven emergence of conceptual structures. Fig. 6 illustrates how the clustering structure discernable on a self-organizing map can correspond to a hierarchical structure. This basic principle can be applied in multiple ways to provide a bridge between the raw data directly linked with some phenomenon and the linguistic and symbolic description of its conceptual structure. Fig. 6 also shows why the SOM is gaining popularity as a user interface element replacing, e.g., traditional menus. With suitable labeling of the map, the user can easily browse the map zooming into details when necessary. The structure of the map

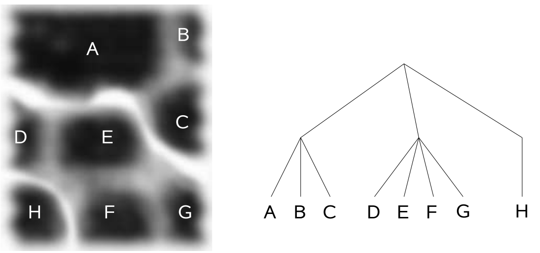


Figure 6: The emergent structure of an organized self-organizing map reflects the structure of the underlying data. The clustering that is visible on the left can be interpreted as the tree diagram on the right.

depends on the data and its preprocessing. The preprocessing can be used to bring up relevant structures. For instance, it may be useful to increase or decrease the weight of some variables. If we consider a map of an e-mail collection, one can decrease the weight of those variables that are commonly related to unsolicited e-mail messages. This leads into the situation in which the SOM can allocate more resources on modeling useful e-mail. The same idea applies to many application areas in which a particular point of view is more relevant than some other.

3.3 Quality assessment of medical web sites

The web has become an increasingly important source of medical information replacing much of the area of medical self-help literature. The consequences of relying on medical information found on web sites can be crucial in terms of making decisions about treatment. Hence there is need for assessing the quality of these web sites to help the layman decide which information he/she should rely on.

Currently, quality labeling of medical web sites is done by various organizations of medical professionals. The assessment process is usually done completely by hand requiring a large amount of manual work by trained physicians in searching web sites for the required elements (including, for example, proper contact information). The EU funded project MedIEQ[1] aims to develop tools to facilitate the process of web site quality labeling.

The MedIEQ project applies mostly the current semantic web technologies to describe the web site contents. However, already in the early stages of the project, it has become apparent that some kinds of contents are difficult to analyze using the structured approach. For instance, the target audience of a medical web site is a property by which the site is being labeled in the assessment process. As it turns out, it is less than trivial to automatically detect whether a site is intended to be read by laymen or medical professionals. Further, the division of the target audience types into crisp categories is often subjective by nature.

[1] http://www.medieq.org/

4 Conclusions

We have presented some motivation why the core technologies in Semantic Web should not solely rely on predicate logic and related formalisms. We have argumented for a certain data-driven approach in which the original data is analyzed automatically rather than relying on hand-crafted ontologies and their use as a basis for choosing descriptors in the metadata. We have given examples of such an approach mainly using the Self-Organizing Map as the core method.

References

M. J. Bates. Subject access in online catalog: a design model. *Journal of the American Society of Information Science*, 37(6):357–376, 1986.

S.C. Deerwester, S.T. Dumais, T.K. Landauer, G.W. Furnas, and R.A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41:391–407, 1990.

G. W. Furnas, T. K. Landauer, L. M. Gomez, and S. T. Dumais. The vocabulary problem in human-system communication. *Communications of the ACM*, 30(11):964–971, 1987.

P. Gärdenfors. *Conceptual spaces: The Geometry of Thought*. MIT Press, 2000.

C.L. Hardin. *Color for Philosophers - Unweaving the Rainbow*. Hackett Publishing Company, 1988.

M. Holi and E. Hyvönen. *Soft Computing in Ontologies and Semantic Web*, chapter Modeling Uncertainty in Semantic Web Taxonomies, pages 31–46. Springer, 2006.

T. Honkela, V. Pulkki, and T. Kohonen. Contextual relations of words in grimm tales analyzed by self-organizing map. In F. Fogelman-Soulié and P. Gallinari, editors, *Proceedings of ICANN'95, International Conference on Artificial Neural Networks,* pages 3–7, Paris, France, October 1995. EC2 et Cie, Paris.

T. Honkela, S. Kaski, K. Lagus, and T. Kohonen. Newsgroup exploration with WEBSOM method and browsing interface. Technical Report A32, Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo, Finland, 1996.

T. Honkela, S. Kaski, K. Lagus, and T. Kohonen. WEBSOM—self-organizing maps of document collections. In *Proceedings of WSOM'97, Workshop on Self-Organizing Maps*, pages 310–315. Espoo, Finland, 1997.

T. Kohonen. *Self-Organizing Maps*. Springer, 2001.

T. Kohonen, S. Kaski, and H. Lappalainen. Self-organized formation of various invariant-feature filters in the adaptive-subspace SOM. *Neural Computation*, 9:1321–1344, 1997.

T. Kohonen, S. Kaski, K. Lagus, J. Salojärvi, J. Honkela, V. Paatero, and A. Saarela. Self organization of a massive text document collection. In *Kohonen Maps*, pages 171–182. Elsevier, Amsterdam, 1999.

P. Koikkalainen and E. Oja. Self-organizing hierarchical feature maps. In *Proc. IJCNN-90, International Joint Conference on Neural Networks, Washington, DC*, volume II, pages 279–285, Piscataway, NJ, 1990. IEEE Service Center.

M. Koskela, J. Laaksonen, M. Sjöberg, and H. Muurinen. PicSOM experiments in TRECVID 2005. In *Proceedings of the TRECVID 2005 Workshop*, pages 267–270, 2005.

J. Laaksonen, M. Koskela, and E. Oja. Picsom: Self-organizing maps for content-based image retrieval. In *Proceedings of IEEE International Joint Conference on Neural Networks (IJCNN'99)*, pages 2470–2473, 1999.

J. Laaksonen, M. Koskela, and E. Oja. Picsom - self-organizing image retrieval with mpeg-7 content descriptions. *IEEE Transactions on Neural Networks, Special Issue on Intelligent Multimedia Processing*, 13(4):841–853, 2002.

K. Lagus, S. Kaski, and T. Kohonen. Mining massive document collections by the WEBSOM method. *Information Sciences*, 163:135–156, 2004.

Z. Ma. *Soft Computing in Ontologies and Semantic Web*. Springer, 2006.

M. Nikravesh. *Soft Computing in Ontologies and Semantic Web*, chapter Beyond the Semantic Web: Fuzzy Logic-Based Web Intelligence, pages 149–209. Springer, 2006.

H. Ritter and T. Kohonen. Self-organizing semantic maps. *Biological Cybernetics*, 61(4):241–254, 1989.

P. Vogt. Cumulative cultural evolution: Can we ever learn more? In S. Nolfi et al., editor, *Proceedings of SAB 2006, From Animals to Animats 9*, pages 738–749, Berlin, Heidelberg, 2006. Springer.

L. Vygotsky. *Thought and language*. MIT Press, 1986, originally published in 1934.

L. A. Zadeh. Fuzzy sets. *Information and Control*, 8: 338–353, 1965.

Julkaisuluettelo – 2006 – List of Publications

	Konferenssijulkaisuja – Conference Proceedings	**€**
1.	Hyvönen E., Seppänen J., Syrjänen M., eds. (1984): STeP-84 – Symposium Papers, Vol. 1.	6
2.	Hyvönen E., Seppänen J., Syrjänen M., eds. (1984): STeP-84 – Tutorial and Industrial Papers, Vol. 2.	6
3.	Karjalainen M., Seppänen J., Tamminen M., eds. (1986): STeP-86 – Symposium Papers: AI and Philosophy, Vol. 1	6
4.	Karjalainen M., Seppänen J., Tamminen M., eds. (1986): STeP-86 – Symposium Papers: Methodology, Vol. 2.	6
5.	Karjalainen M., Seppänen J., Tamminen M., eds. (1986): STeP-86 – Symposium Papers: Applications, Vol. 3.	6
6.	Mäkelä M., Linnainmaa S., Ukkonen E., eds. (1988): STeP-88 – Contributed Papers: Applications, Vol. 1.	6
7.	Mäkelä M., Linnainmaa S., Ukkonen E., eds. (1988): STeP-88 – Contributed Papers: Methodology, Vol. 2.	6
8.	Syrjänen M., Oivo M., Salonen P., eds. (1990): STeP-90 – Tutorial and Symposium Papers	6
9.	Hyvönen E., Seppänen J., Syrjänen M., eds. (1992): STeP-92 – New Directions in AI – Papers, Vol. 1	10
10.	Hyvönen E., Seppänen J., Syrjänen M., eds. (1992): STeP-92 – New Directions in AI – Symposia, Vol. 2	10
11.	Hyvönen E., Seppänen J., Syrjänen M., eds. (1992): STeP-92 – New Directions in AI – Workshops, Vol. 3	10
12.	Carlsson C., Järvi T., Reponen T., eds. (1994): STeP-94 – Multiple Paradigms for Artificial Intelligence	10
13.	Alander J., Honkela T., Jakobsson M., eds (1996): STeP-96 – Genes, Nets and Symbols	12
14.	Koikkalainen P., Puuronen S., eds (1998): STeP-98 – Human and Artificial Information Processing	14
15.	Hyötyniemi H., ed. (2000): Feedback to the Future Systems, Cybernetics and Artificial Intelligence	14
16.	Hyötyniemi H., ed. (2000): STeP 2000 – Millennium of Artificial Intelligence – 'AI of Today': Symposium on Applications	14
17.	Hyötyniemi H., ed. (2000): STeP 2000 – Millennium of Artificial Intelligence – 'AI of Tomorrow': Symposium on Theory	14
18.	Ala-Siuru P., Kaski S., eds. (2002): STeP 2002 – The Art of Natural and Artificial	14
19.	Hyötyniemi H., Ala-Siuru P., Seppänen J. eds. (2004): STeP 2004 – Life, Cybernetics and Systems Sciences	14
20.	Hyvönen E., Kauppinen T., Salminen M., Viljanen K. Ala-Siuru P, eds. (2004): STeP 2004 – Web Intelligence	14
21.	Alander J., Ala-Siuru P., Hyötyniemi H., eds. (2004): STeP-2004 – Origin of Life and Genetic Algorithms	14
22.	Honkela T., Raiko T., Kortela J., Valpola H., eds. (2006): Ninth Scandinavian Conference on Artificial Intelligence (SCAI 2006)	18
23.	Hyvönen E., Kauppinen T., Kortela J., Laukkanen M., Raiko T., and Viljanen K., eds. (2006): STeP-2006 – New Developments in Artificial Intelligence and the Semantic Web	18

	Symposiojulkaisuja – Symposium Publications	**€**
1.	Keskinen K., toim. (1989): Epävarmuus, päätöksenteko ja tietämystekniikka	6
2.	Karanta I., ed. (1989): Hyperteksti ja tekoäly	6
3.	Hynynen J., Linnainmaa S., eds. (1990): The French-Finnish Colloquium on AI Applications	6
4.	Karanta I., Seppänen J., eds. (1993): Kaaostutkimus ja tietotekniikka – katastrofeja, attraktoreja, ja fraktaaleja	12
5.	Stark S., Tyystjärvi K., eds. (1990): Ajattelevatko koneet? – Tekoäly, tietotekniikka ja robotiikka	12
6.	Hautamäki A., Nyman G., toim. (1992): Kognitiotiede ja koneäly	12
7.	Lehtola., Jokiniemi J., eds. (1992): Embedded and Real-Time Systems	14
8.	Kurenniemi E., ed. (1992): A Bibliography on SETI	10
9.	Linnaluoto S., Seppänen J., eds. (1992): SETI – Search for Extraterrestrial Intelligence	12
10.	Bulsari A., Saxén B., eds. (1993): Neural Network Research in Finland	14
11.	Lehtola A., Jokiniemi J., eds. (1993): Conceptual Modelling and Object-Oriented Programming	14
12.	Hyvönen E., Seppänen J., toim. (1995): Keinoelämä – Artificial Life. Tekniikkaa, luonnontiedettä, filosofiaa ja taidetta	14
13.	Haaparanta L., Hyvönen E., Seppänen J., Silvonen J., toim. (1995): Älyn ulottuvuudet ja oppihistoria – Matka logiikan, psykologian ja tekoälyn juurille	14
14.	Gefwert C., Orponen P., Seppänen J., toim. (1996): Logiikka, matematiikka ja tietokone – Logic, Mathematics and the Computer	14
15.	Honkela T., toim. (1999): Pelit, tietokone ja ihminen – Games, Computers and People	14

	SCAI – Scandinavian AI Conference Proceedings	**€**
1.	Jaakkola H., Linnainmaa S., eds. (1989): SCAI' 1989 – Proceedings of the Second Scandinavian Conference on Artificial Intelligence, Vols. 1-2, Tampere	10
2.	Grahn G., ed. (1997): SCAI' 1997 – Proceedings of the Second Scandinavian Conference on Artificial Intelligence, Vol. 1-2, Helsinki	10
3.	Alander J., ed. (1997): Proceedings of the Third Nordic Workshop on Genetic Algorithms and their Applications (3NWGA), Vaasa	10

(See also: SCAI 2006 proc. item 22 in Conference Proceedings above.)

	Kansainvälisiä kokouksia – International Conferences	**€**
1.	Bulsari Abhay, Kallio Sirpa, eds. (1995): Engineering Applications of Artificial Neural Networks. Proceedings of the International Conference EANN '95, Otaniemi, 21-23 August 1995, Finland.	14
2.	Pylkkänen Paavo, Pylkkö Pauli, eds. (1995): New Directions in Cognitive Science. Proceedings of the International Symposium, Saariselkä, 4-9 August 1995, Lapland, Finland.	14

Tiedotuslehti – Arpakannus – Magazine	**€**
1986 alkaen 2-4 numeroa/v – 2-4 issues/year 1986-1996	6
1997 alkaen ajoittain – Since 1997 occasional issues	6

To order, call **+358 9 5617 830**, send a fax to **+358 9 5607 5365** or email to **office@stes.fi**.

When ordering publications, please state the **name(s) of the publication(s)** and the **amount of copies** you want to buy. Please also give **your post address** (name, street address, zip/postal code and country) for delivering the publications to you by post. Shipping costs 5€ will be added to the total sum of your order.

Contact information

Finnish Artificial Intelligence Society, Susanna Koskinen, c/o Toimistopalvelu Hennax T:mi,
Henrikintie 7 D, FIN-00370 HELSINKI
WWW: http://www.stes.fi
email: office@stes.fi
phone: +358 40 535 2617