Finally, repeated entries are introduced. Only this final stage is required during the processing of spanning cells. This process is illustrated in Figure 9, where Figure 9a shows an example of a nested stub (see Figure 4 for the equivalent with spanning cells) and Figure 9b shows the equivalent relational fonn, with reintroduced repeated entries indicated by italics.

(a) Nested stub. (b) Transformed stub.

Figure 9: Transformation of a Nested Stub.

4. Conclusion.

By classifying table layout and tailoring table processing accordingly, the TREATS approach to processing plain text tables can effectively support a large variety of table layouts. It has been tested on many of the examples used in the aforementioned layout classification survey and has proved to be highly effective.

We have identified a small number of limitations of our approach, where processing is affected by ambiguous layouts due to empty cells resulting from missing entries and lack of indentation to indicate where multi-line entries appear. Missing entries result in the incorrect formation of spanning cells and lack of indentation means that multi-line entries cannot always be formed. Unfortunately, there is no way to overcome these limitations automatically, due to conflicts with other types of layout. These problems can only be overcome through user interaction, but are exceptional.

Finally, whilst we have not considered hybrid tables in this paper, support for the most commonly appearing forms is available in the TREATS toolkit through suitable combination of the techniques we have described.

5. References.

[1] L. E. Hodge. W. A. Gray and N. J. Fiddian. Effective Reuse of Textual Documents Containing Tabular Information. *Proceedings of the 4th Australasian Document Computing Symposium (ADCS 99)*, Coffs Harbour, NSW. Australia, December 1999, pp 47 - 53.

[2] L. E. Hodge, W. A. Gray and N. J. Fiddian. A Toolkit to Facilitate the Integration of Tabular Information in Textual Documents with Database Applications. *Proceedings of the 4th Multiconference on Systemics, Cybernetics and Informatics (SCI 2000)*, Orlando. Florida, USA, **July** 2000.

[3] *The Chicago Manual of Style*, Thirteenth Edition, The University of Chicago Press, 1982.

[4] R. J. Beach. Tabular Typography. *Proceedings of the International Conference on Text Processing and Document Manipulation*, University of Nottingham. April 1986. pp 18- 33.

[5] T. Pavalidis and J. Zhou. Page Segmentation by White Streams. *Proceedings of the International Conference on Document Processing (ICDAR '91)*. Saint Malo, Fiance. 1991. pp 945-953.

[6] S. **Chandran** and R. Kasturi. Structural Recognition of Tabulated Data. In *Proceedings of the International Conference on Document Processing (ICDAR 93)*. 1993.

[7] K. Itonori. Table Structure Recognition Based on Textblock Arrangement and Ruled Line Position. *Proceedings of the International Conference on Document Processing (ICDAR 93)*, 1993.

[8] T. G. Kieninger. Table Structure Recognition Based on Robust Block Segmentation. *Proceedings of Electronic Imaging 98 (SPIE), Document Recognition*, 1998.

[9] J. D. Farmer. T. Toffoli and S. Wolfram, editors. *Cellular Automata: Proceedings of an Interdisciplinary Workshop*, Los Alamos, New Mexico, March 7-11, 1983.

[10] Stephan Wolfram. *Cellular Automata and Complexity: Collected Papers*. Addison-Wesley, 1994.

Recovering Structure from Unstructured Web-accessible Classified Advertisements

Richard Cole, Peter Eklund and Age Strand
School of Information Technology, Griffith University
PMB 50 Gold Coast MC
QLD 9726. Australia
{r.cole,p.eklund}@gu.edu.au, mstrand@hotmail.com

Abstract

This paper describes a research prototype system called RFCA for structuring Web-accessible rental classified advertisements based on semantic content. A hand crafted parser is used to extract various facets of the rental property being advertised including amongst others; member of room, type of garage, dwelling type (unit, house, or high rise apartment), price and contact details. The performance of the parser is measured in terms precision and recall by comparing its output to that of human expert.

The structured information once extracted is stored in a relational database and users searching for rental properties are presented with a graphical organisation of rental properties according to predefined themes. The overall result is a suite of tools for extracting, cleaning, structuring, and visually querying/browsing collection of web-accessible rental advertisements.

The mathematical and, methodological foundation for the graphical organisation of the structured information is provided by formal concept analysis. Using formal concept analysis each property is understood to be, an object possessing attributes with attribute values. The data is then conceptually organised via concept lattices dynamically according to a pre-definedconceptualscales. The, concept lattice organises rental properties into conceptual groupings. The, user then has the opportunity to view the attributes of all properties in a grouping as well as navigate back to the source advertisements.

The, interface is delivered over the web using a CGI interface and dynamic creation of image and image maps. The, ideas presented are general enough to be relevant to other web-accessible unstructured text sources.

1 Rental Formal Concept Analysis

Many newspapers hold large keyword indexed free-text collections of classified advertising on the Web and these can be searched on-line. The intention with this work is to demonstrate how such data can be value-added by extraction, cleaning and structuring. A structured database derived from free-text classifieds can then be browsed effectively. We argue that a browsing interface that structures the presentation of classifieds, related to a particular purpose (such as rental classifieds), can facilitate retrieval. More specifically, we maintain that a browsing interface using formal concept analysis gives a sense of the way that attributes within the free-text sources are distributed across the text collection, something that a keyword-based search interface cannot do.

Formal Concept Analysis (FCA) [13] has been developed during the last twenty years and successfully applied to data analysis and knowledge processing [15]. The Mathematics of FCA has been carefully described in Ganter and Wille [9] and the basic details of the theory are omitted here for brevity. There have been a number of examples of FCA applied to information retrieval and filtering [10, 1] including our own work [3. 4]. In these systems the main difficulty is attribute identification front texts. In the medical and email domains in which have worked [8. 2. 5], objects are easily identified since they correspond to documents: typical stored as a single file. In the case of Web-accessible rental classifieds the extraction task is complicated by object recognition: several rental classifieds often appear in the same advertisement and are grouped by location. (trice or the number of bedrooms. An example of such a problematic advertisement is illustrated in Fig. 1.

2 Object and Attribute Identification

For these reasons one of the first tasks of a unstructured text parser for RFCA is object recognition. disambiguating a single rental property from an advert that may list several or many properties for rent. For example, in Fig. 1, lines 3, 4, 5, 7 and 8 of the advert are individual properties which require representation as objects. In addition, there are cases where attribute recognition can also be com-

Proceedings of the **Fifth Australasian** Document Computing Symposium, Sunshine Coast, Australia, December 1, 2000.

```
FDR RENT - ARUNDEL - Phone 55948184
$300
4 Bedrm, in-grnd pool, dble garage, near
  shops and school
3 bedrm, tripple garage, immac. presented,
  close to transport
Exec. 3 Bedrm + study, pool, dble garage, all ammen.
  close to  school
$250
Leafy 3 bedrm, double garage, avail. Aug.
3 bedrm townhouse, resort fac. l.up garage,
  2 bathroom and on-suite.
Townhouse, 2 bedroom, resort fac. garage,
  near golf course and transport.
```

Figure 1: A rental classified illustrating multiple aliases for attributes (as in abbreviations such as Bedrm=bedroom), multiple objects (as rental properties described on lines 3, 4, 5, 7 and 8) in a single advert (all lines) clustered on an primary key attribute: in this case the two prices $300 and $250.

plex because multiple FCA objects may be clustered under a single attribute. For instance, in Fig. 1 $300 per week applies to the properties 3, 4 and 5 while $250 applies to properties described in lines 7 and 8.

To formalise the understanding of objects as properties having associated attributes we introduce a basic structure of formal concept analysis — the multi-valued context. A *multi-valued context* is a tuple (G, M, W, I) where G is a set of properties. M is a set of attributes. $W = \cap_{m \in M} W_m$ is a set of attribute values and $I \subseteq G \times M \times$ IF is a relation saying which rental properties have which attribute values for which attributes. The relation I is restricted so that for any rental property and attribute there is only one attribute value in I. More formally; I is a relation such that if $(g,m,w) \in I$ and if $(g, m, w_2) \in I$ then $w_1 = w_2$.

Multi-contexts are converted to single valued contexts via a mechanism called conceptual scaling. A conceptual scale for an attribute m is a triple $\mathbb{S}_m = (\text{IF}_{,,,}, M_s, I_s)$where M_s is a set of new binary attributes. The relation $I_s \subseteq \text{IF}_{,,} \times M_s$ says which new attributes are indicated by which attribute values. So for example consider the conceptual scale in Figure 2. This scale introduces the new attributes *some kind of parking, double, single, garage, carport,* and *no parking*. Each circle represents a collection of real estate properties. For example the circle marked *single* represents all properties with a *single* car spot. The circle marked *garage* represents properties with a *garage* and the circle below and connected to these two circles represents properties with *single garages*.

Each circle in the conceptual scale has associated with it a fragment of an SQL query that allows

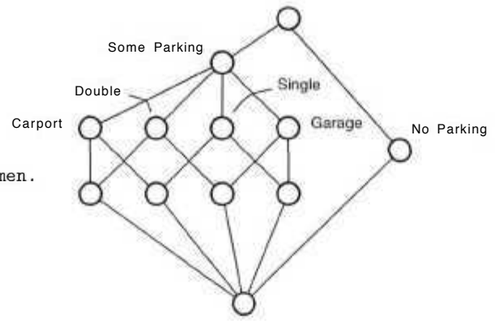

Figure 2: An example conceptual scale for parking. The scale implies that a rental property cannot have a carport that is both a double and a single.

the rental properties for that circle to be extracted from the relational database. One may notice in the conceptual scale that there is a type ordering present. For example because the circle for *garage* appears below the circle for *some parking* the set of rental properties with garages are considered a subtype of the set of properties with parking. Another way to look at this situation is to see that the type *some parking* is a super-type of the disjunction of rental properties with a single or double. garage or carport.

The concept lattice derived from the conceptual scale has the same structure as the scale hut indicates the number of rental properties classified under each circle. These numbers are calculated dynamically by interrogating the database and using the SQL fragments attached to each circle. This approach avoids having to either calculate large concept lattices or indeed try to read large lattices. It is possible to combine conceptual scales using a special diagram called a "nested line diagram" [9].

Figure 3 shows a concept lattice derived from the scale for '"Geographical position on the Gold Coast". Suburbs are clustered in the middle layer of the hierarchy and since no property in the set we considered can be in more than one suburb there are no intersections between the attributes in the third layer. For scales such as the "facilities" scale, including attributes like "close to shops", "close to transport", "close to sporting facilities" there are many instances of rental properties having a mixture the attribute. By combining two scales the user is able to see the trade-off between various facilities, cost, geographical location or number of car spaces.

Suburbs are clustered in the central layer of the diagram in Figure 3. Since suburbs are mutually exclusive sets, there are no intersections between the rental properties, the numbers of which are showing in the third layer, and the central layer. In other thematic views, such as the parking scale

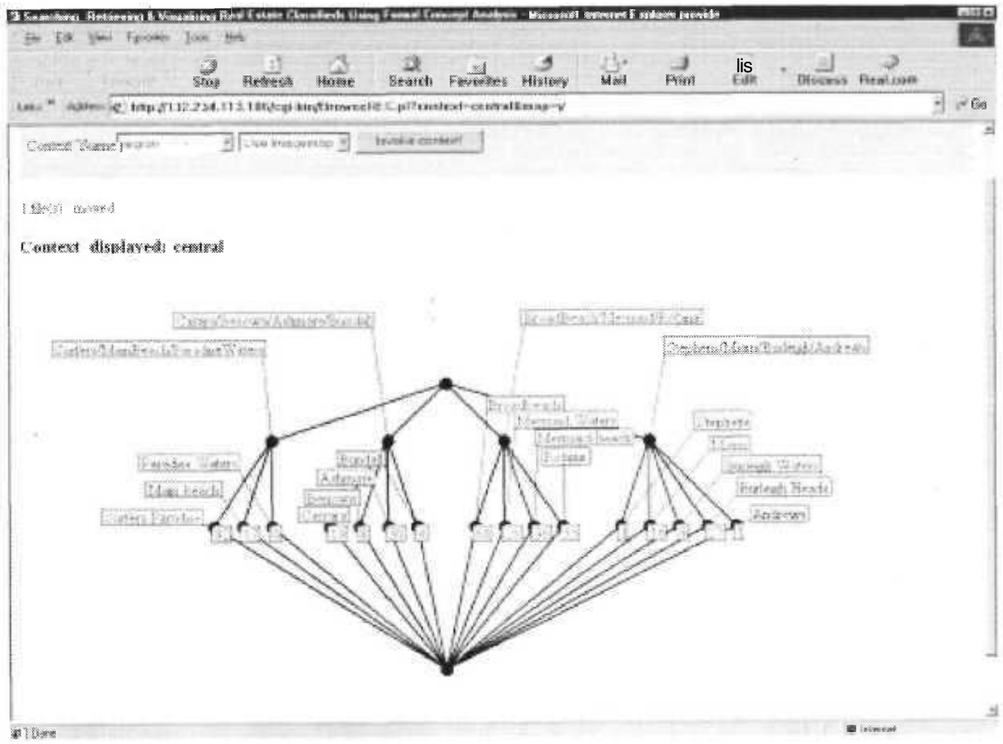

Figure 3: A concept lattice showing the breakdown of location and their geographic spread on the Gold Coast.

shown in Figure 2. there are many instances of rental properties exhibiting various attribute combinations, an example is shown in Figure 4. where the number 5 indicates the number of rental properties that exhibit the attributes double and carport. By combining two scales the user can explore the tradeoff between attributes in both scales. e.g. geographical locations and parking.

In other applications [2. 4] we have experimented with allowing the user to construct conceptual scales on the fly. However this raises questions of scalability [5] and graph layout [6]. In such a constrained task as searching and browsing rental advertisements we deem the approach outlined in the following section sufficiently flexible.

3 Browsing the Rental Classifieds

Consistent with the standard practice of formal concept analysis [9] we consider the navigation space as the direct product of the concept lattices derived from all scales. The user employs two basic operations to focus their attention at varying levels of details. The two basic operations are nesting and zooming.

Nesting allows the user to combine two conceptual scales. For example if the user is considering the concept lattice derived from the scale in Figure 2 and wants to see how the number and type of car spaces is affected by geographical area then the user can construct a nested line diagram. Such a combination is shown in Figure 5.

Zooming allows the user to restrict the set of rental properties shown in the diagram. Say for example that the user has after looking at the nested diagram in Figure 5 decided that they are interested in properties in Surfers/Main Beach with some parking. The user can zoom in by selecting the concept for "Some Parking" and "Surfers/Main Beach". In this case the next scale selected will show a concept lattice containing only the properties in Surfers/Main Beach that have some parking mentioned. The user also has the possibility to anti-zoom. That is remove the previous zooming restriction. By composing a series of zooms, anti-zooms and nestings the user is able to organise the properties

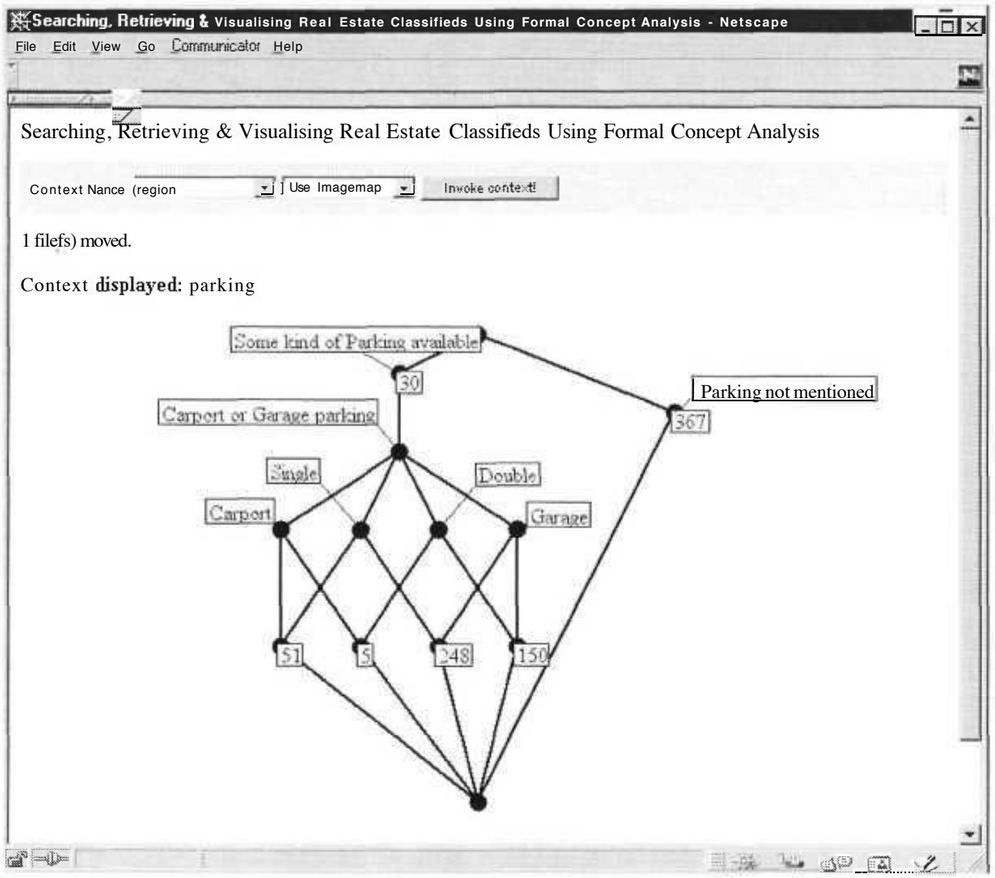

Figure 4: A concept lattices showing the types of parking available in the real estate advertisements.

at different levels of detail either focusing of properties according to some specific criteria or taking a global perspective. Although we haven't implemented zooming and nesting in the web based interface RFCA we have employed it our other systems [12, 4] we've developed. Nesting and zooming was first implemented in a system called TOSCANA[16].

By it's very nature the view of data represented by a concept lattice makes evident different levels of detail. The top concept (concepts are represented by circles) in a concept lattice refers to all objects under consideration. In the absence of zooming this will be all rental properties. As one moves down from the top concept the concepts become more specific. They refer to small and smaller sets of rental properties circumscribed by large and larger sets of attributes. For example in Figure 5 as the user moves directly downwards within the large circle one encounters the attributes for the suburb of the rental properties. Further down still following the line between large circle one encounters attributes related to parking which further restrict the rental properties under consideration.

As an information retrieval tool, zooming and nesting of conceptual scales have not been benchmarked against established IR. techniques, but the browsing metaphor is nonetheless a compelling advantage to their use.

4 Results

On the test set of 2 months worth of rental classifieds (for the Gold Coast only) extracted from a NewCorp Web site[1] (8,456 classified adverts) the system achieved logical recognition of 89% of properties, thus 11% of the property description text was discarded because price, contact number of location could not be resolved by the parser. More

[1] http://www.newsclassifieds.com.au

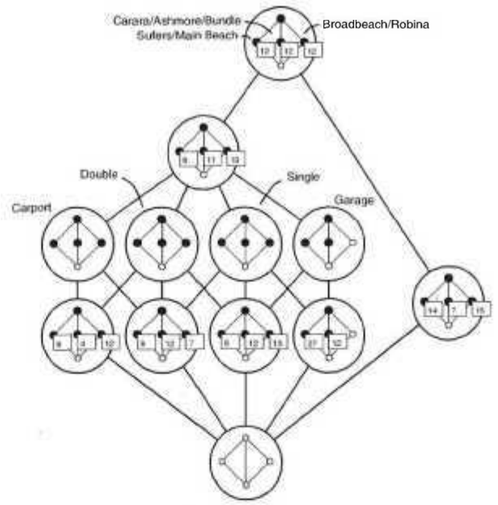

Figure 5: Example of nesting

than 1/3 of the error cases had genuine missing unresolved property listings (commonly no price was given for instance, in such a case we consider that the property listing is useless).

A LL(1) parser was constructed by hand to translate the remaining 89%, of classified advertisements into the multi-valued context representation. The multi-valued context had 64 attributes. 53 of which were single valued (true or false) attributes. The remaining 11 attributes. such as Price, were multi-valued. To access the accuracy of this translation process, precision and recall were measured for each attribute and then aggregated. A summary of the most common and most important attributes is given in Table 1.

	Location	Price	Bedroom
Frequency	100%	100%	100%
Precision	100%	100%,	100%
Recall	94.3%	100%,	98.1%
	Furnished	Car Park	Other
Frequency	26.4%	50.9%,	88.7%
Precision	100%	100%	100%
Recall	71.4%	96.3%	68.1%

Table 1: Recall and Precision for 53 unseen real estate adverts.

The precision and recall of the multi-valued attributes are calculated as the number of correctly identified attributes values as a proportion of the number of identified attributes values, and the number of correctly identified attributes values as a proportion of the number of attribute values respectively. Averaging the most important attributes — Location, Price, Bedroom, Furnished, and Car parking - weighted by their frequency yields a precision of 100% and a recall of 95% while the inclusion of other attribute reduces the recall to 90%. All real estate advertisements leave out some information about the property they are advertising because of the cost of advertising space. As a result we would expect the recall of actual information about the property being advertised to be much less. One of the strengths of formal concept analysis is that it allows the user to compose views of the data that separate objects at different levels of detail. So for example the user may have a course distinction based on cost, but a very fine distinction based on proximity to facilities contained within a single view. Table 1 shows poor recall for attributes in the group Other. When combined with the knowledge that the adverts contain only partial descriptions of the data this places a practical limit on the fineness of detail that can be usefully explored by the user. This limit would be extended if the initial data source was a database containing more extensive information about the properties for sale.

The parser being a hand crafted LL(1) parse was very fast, building the relational database storing the multi-valued context in under 8 seconds on a Pentium-Ill 300 MHz.

5 Conclusion

The purpose of this paper has been to report on a practical application of information filtering and browsing based on formal concept analysis. The system described (RFCA) is useful because it contains components that extract unstructured text from Web-accessible databases, clean the data and then perform object and attribute identification using a JavaCC parser.

Our emphasis is on the visual outcomes that can be used for text data mining showing that the visual complexity of the lattice representation can be used to explore a document collection in an intuitive human-centered interface. Future directions for the work include comparing the use of the hand crafted parser with both text classifiers based on machine learning algorithms, and the use of metadata.

References

[1] C. Carpineto, and Romano, G., A lattice conceptual clustering system and it application to browsing retrieval, *Machine Learning,* Vol. 24, pages 95-122, Kluwer Academic Publishers, The Netherlands".

[2] R. Cole, P. Eklund: Scalability in Formal Concept Analysis: A Case Study using Medical Texts. *Computational Intelligence,* Vol. 15, No. 1, pp. 11-27, 1999.

[3] R. Cole, P. Eklund: Analyzing an Email Collection using Formal Concept Analysis. *Proceedings of the European Conf. on Knowledge and Data Discovery,* pp. 309-315, LNAI 1704, Springer, Prague. 1999.

[4] R. Cole, G. Stumme: CEM - An Email Analysis Tool. *Proceedings of the 8th International Conf. on Conceptual Structures,* pp. 309-315, LNAI 1704, Springer, Darmstadt. 2000.

[5] R. Cole, P. Eklund and G. Stumme: CEM - Visualization and Discovery in Email, *Proceedings of the European Conf. on Knowledge and Data Discovery,* pp. 309-315, LNAI 1704, Springer, Prague, 1999.

[C] R. Cole, Using Force Directed Placement and Genetic Algorithms for Concet Lattice Layout, *Proceedings of Australian Computer Science Communications*. Los Alamitos, CA, 2000. IEEE Press. 2000.

[7] Michael K. Coleman and D. Stott Parker. AGLO -- Publications and Implementation. *Software - Practice and Experience,* pages 1415-1438, December 1996.

[8] R. Cole, P. W. Eklund, D. Walker: Using Conceptual Scaling in Formal Concept Analysis for Knowledge and Data Discovery in Medical Texts, *Proceedings of the. Second Pacific Asian Conference on Knowledge Discovery and Data Mining,* pp. 378-379, World Scientific, 1998.

[9] B. Ganter, R. Wille: *Formal Concept Analysis: Mathematical Foundations*. Springer, Heidelberg 1999 (Translation of: Formale Begriffsanalyse: Mathematische Grundlagen. Springer, Heidelberg 1996)

[10] R. Godin, Gecsei, J. and Pichet, C: Design of a Browsing Interface for Information Retrieval, *SIG-IR,* pages 246-267, 1987.

[11] R. Godin. and Missaoui, R. and Alaoui, H. Incremental Concept Formation Algorithms based on Galois (Concept) Lattices. Computational Intelligence, Vol. 11, number 2, pp. 246-267, 1995.

[12] G. Stumme: Hierarchies of Conceptual Scales. *Proc. Workshop on Knowledge Acquisition, Modeling and Management*. Banff, 16.-22. October 1999

[13] R.. Wille: Restructuring lattice theory: an approach based on hierarchies of concepts. In: I. Rival (ed.): *Ordered sets*. Reidel, Dordrecht-Boston 1982, 445-470

[14] R. Wille. Line diagrams of hierarchical concept systems. *International. Classification,* 11:77-86, 1984.

[15] P. Wille: Conceptual Graphs and Formal Concept Analysis. In: *The 4th International Conference on Conceptual Structures*. LNAI 1257, pages 2-18, Springer Verlag, 1997.

[16] F. Vogt, C. Wachter, R. Wille: Data Analysis based on a conceptual file. In: *Classification, data analysis and knowledge organsiation.* pages 131-140, 1991.

Implementing Shared Document Preparation with Lightweight Editing

Michael J Rees

School of Information Technology
Bond University
Qld 4229, Australia

mrees @ bond. edu.au

Abstract

Virtually all web pages are read-only, yet the first web browser allowed users to read and edit every page. Special ad-hoc mechanisms are needed to make all or part of a page editable by a user. This paper describes Pardalote lightweight editing, a document management feature for allowing many users to share the editing of a web page using only a web browser. A brief oven'iew of how Pardalote is implemented is followed by examples of shared document preparation using Pardalote. The benefits of such web document management are discussed. Future Pardalote extensions using XML precede the closing remarks.

Keywords Shared document management, cooperative document preparation, lightweight editing, I-grains, fraglets, user interface design, computer supported cooperative work.

1. Introduction

Tim Berners-Lee describes the development of the World-Wide Web and the first browsers and servers in [1]. In 1990 the excellent design of the NextStep operating systems running on the Next machine made it very straightforward to allow any web page displayed by the web browser to be edited in-situ. Only when the pressure mounted to provide browsers on several other hardware platforms was the in-browser web page editing feature abandoned. Those second stage browsers simply displayed web pages and set the browser model that still holds today.

Although the original Mosaic browser thought to provide us with annotation capability, it was not until 1995 that MIT organised a meeting [2] to discuss methods for making web pages into a collaborative medium. Many large software companies and large research projects were represented there. Each paper presented a different mechanism to achieve collaboration via the Web. Several of these solutions are still available today as commercial products.

In the same year, 1995, the working group that was to produce the WebDAV interoperability specification, World Wide Web Distributed Authoring and Versioning [3], was formed. WebDAV [4] is an extension to the HTTP protocol that supports web document metadata, namespace management (like file system directories), overwrite protection, and versioning management. In effect, WebDAV allows web documents (any file that has a URL) to be edited asynchronously by many users, providing collaborative authoring. Of course, WebDAV support must be built into browsers and servers, a not inconsiderable implementation effort. Once achieved, however, this will provide access to genuinely collaborative web documents at last.

In the view of the author at the time of writing. WebDAV offers considerable benefit in the longer term but in the context of this paper, at the cost of a heavyweight solution in terms of software implementation. Users wishing to collaborate via web pages using WebDAV will need to wait for its use to become widespread.

A lightweight solution to collaborative web document editing is one that can use existing browsers and servers. To be truly lightweight, the solution must be a convenient one for all types of users involved:

- Members of collaborative teams who wish to jointly prepare and share web documents, the ultimate end-users
- Original authors of the collaborative web documents
- Web site administrators who install the software that makes collaborative editing of web documents possible

Probably the best example of such lightweight editing is the work of the Sparrow Project [5] from Xerox PARC. The author in [6] and [7] has described examples of Sparrow's capability. The first user type, end-users, is very well catered for in Sparrow. The user is presented with a very simple and intuitive user interface to edit nominated sections of the web document (HTML page). Edits are restricted to the textual content in specially marked locations.