Conferentie informatiewetenschap 1999 : Centrum voor Wiskunde en Informatica, 12 november 1999 : proceedings

Eindhoven University of Technology
Department of Mathematics and Computing Science

Conferentie Informatiewetenschap 1999

Centrum voor Wiskunde en Informatica
12 november 1999

Proceedings

Edited by P. De Bra and L. Hardman

99/20

ISSN 0926-4515

editors: prof.dr. J.C.M. Baeten
prof.dr. P.A.J. Hilbers

Reports are available at:
http://www.win.tue.nl/win/cs

Computing Science Reports 99/20
Eindhoven, December 1999

Conferentie Informatiewetenschap 1999

Centrum voor Wiskunde en Informatica
12 november 1999

Proceedings

edited by P. De Bra and L. Hardman

A modular structure for electronic scientific articles

F.A.P. Harmsze[1], M.C. van der Tol[2] and J.G. Kircz[1]
[1]Van der Waals-Zeeman Institute, University of Amsterdam
Valckenierstraat 65, 1018 XE Amsterdam, The Netherlands
[2]Speech Communication, Argumentation Theory and Rhetoric, University of Amsterdam
Spuistraat 134, 1012 VB Amsterdam, The Netherlands

Abstract

We have developed a modular structure for electronic articles on experimental science. Modular articles consist of different types of explicitly characterised modules and explicitly characterised links expressing different types of relations. The modules can be located, retrieved and consulted both separately and in conjunction with other modules.

The project

At present, we face a revolution in the dissemination and handling of scientific papers. Most major publishers make an important share of their publications available via an Internet site and many independent new initiatives are launched. With a few exceptions, these electronic publications are in fact reproductions of paper-based products. As always, with the introduction of a new technology, the first steps in a new era are characterised by the translation of the old methods and models into the new situation. Only when the intrinsic characteristics of the new technologies are fully appreciated do real novel developments get a chance.

In the project 'Communication in Physics', we try to go a step further and propose a new model for the creation and evaluation of electronic scientific articles, taking into account the intrinsic features of the new medium, the requirements of adequate scientific communication as well as the societal traditions on which regular scientific communication is based. This model is intended to work in a fully electronic environment, where all papers are linked to each other and where new scientific contributions are added to the existing pool of papers in an organic way.

Rather than concentrating on the capability of present-day software, we choose an analytical approach. In other words, we design a new way of presenting scientific results, based on the assumption that appropriate software will become available in the foreseeable future. We analysed the role of articles in scientific communications following the standard literature (Garvey 1979, Meadows, 1999). Subsequently, we draft a profile of the interactants in the communication process. Here, we rely on discourse and argumentation studies concerning rational communication (Van Eemeren et al., 1993). This way we are able to connect the characteristics of scientific articles with the various stages in the communication process. This leads to a series of specific requirements that electronic scientific articles have to satisfy to allow for effective and efficient communication. These requirements include: a) dissemination requirements like indexing and logistics tools as well as proper identification and registration of intellectual ownership and integrity of the work, and b) creation requirements, resulting in authoring tools and electronic templates.

Based on our analyses of the role of the article and the communication criteria in academia, we conclude that in an electronic environment the traditional linear essay form becomes obsolete and has to be replaced by a modular framework (Kircz, 1998).

In order to ensure that our model is grounded in scientific practice, we developed the model in conjunction with an analysis of a coherent corpus of printed articles in the field of experimental physics. In this analysis, we identify different types of information and relations in the corpus and re-organise that information in a novel, modular structure. We found that the modular structure indeed allows for the creation of scientific articles that meet the necessary requirements.

In an earlier presentation in this series of conferences, we gave an outline of our programme ((Harmsze et al., 1996)). In this contribution, we would like to present the final model, which will soon be fully reported elsewhere (Harmsze, 2000). In that thesis, the model will be specified in terms of instructions to authors and will provide recommendations for software implementation.

The framework

Because electronic media are suitable for multiple (re)usage and reshuffling of information units, as well as for additions of new components to published work; our guiding principle is 'modularity'(Kircz, 1998). We develop a structure for modular articles, based on the idea that an electronic article can be made up of well-defined *modules* and *links* that, following the SGML-philosophy, can be identified with tags. In our modular framework, we define the modules that can represent the different types of information in an article. In order to guarantee and express the coherence of the information in and between different modules, we introduce a systematic way of linking the modules, both within the same article and between different publications. Thus, a modular article represents a sub-network of information within the network of all published information. In our model, both modules and links are explicitly characterised 'information objects' that can be handled using state of the art database management and information retrieval techniques.

Modules

We define a module as a uniquely characterised, self-contained representation of a conceptual information unit that is aimed at communicating that information. Not its length, but the coherence and completeness of the information it contains makes it a module. Modules can be located, retrieved and consulted separately as well as in conjunction with related modules.

The relations between modules can be expressed not only in links, but also in the composition of elementary modules into higher-level, *complex modules*. We define a complex module as a module that consists of a coherent collection of (elementary or complex) modules and the links between them. Using a metaphor, elementary modules are 'atomic' entities that can be composed into a 'molecular' entity: a complex module.

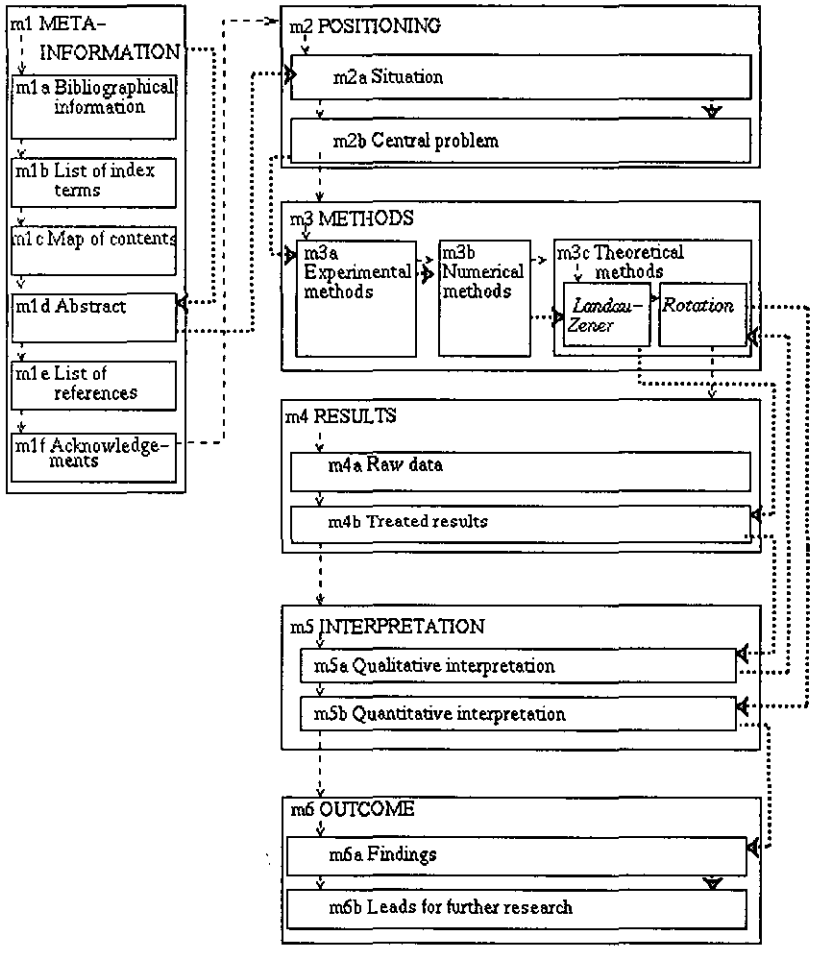
We distinguish two types of complex modules: *compound* modules and *cluster* modules. In a compound module, related (albeit possibly dissimilar) modules are aggregated to form a new module on a higher level. An example of an aggregated module is the module 'Experimental methods' that is composed of lower-level modules representing the various components of a measuring device. In our corpus we encounter molecular beam apparatuses that have, as relatively independent components, things like: one or more sources of a particle beam, a beam transport system, an interaction chamber and a detector. The central concept of a cluster module is the generalisation of specific concepts, focused on in its constituent modules. An example of a cluster module is the module 'Raw data' composed of various elementary modules reporting the results of the same general type of measurements involving different molecules.

In order to be able to determine what is 'similar information' to be grouped together and represented in a self-contained module and, subsequently, in order to be able to determine how to tag the resulting module, we need an unambiguous typology of scientific information. Therefore, we introduce a typology by which we characterise the information from four complementary points of view. In this typology, we incorporate the characterisations from two classical points of view: the domain-oriented characterisation that can be expressed in keywords and the characterisation by specified bibliographic data. In addition, we introduce a characterisation by the range of the information and a characterisation by its conceptual function, i.e. by the role the information plays in the scientific problem-solving process.

By characterising information by its range, so-called *microscopic*, *mesoscopic* and *macroscopic* modules can be introduced. A microscopic module represents information that belongs only to one particular article, e.g., information concerning the specific problem addressed in that article. A mesoscopic module functions at the level of an entire research project; it is created for multiple use in several articles issued from the same project. For example, information about the experimental set-up that has been used in a series of experiments can be represented in a mesoscopic module and connected to several articles reporting experimental results. A macroscopic module represents information that transcends the level of the research project; this type of firmly established information is given in, e.g., books, lecture notes.

Our main division in modules is based on the characterisation of the information by its conceptual function. Our starting point is the prototypical section structure of scientific papers: Introduction, Methods, Results, Discussion and Conclusions. This sequence represents the normal flow of a scientific narrative, but the way it is used in practice presupposes that the article will, indeed, be read sequentially from the beginning to the end. One of the main arguments in favour of modularity is that knowledgeable readers hardly read articles sequentially but browse through them, looking for useful bits and pieces. In our approach, we take that behaviour as our starting point and define our modules as entities that can be read independently. Thus, every module represents only one well-defined aspect of the article. Of course, this independence does not mean that one module is in general sufficient to understand the whole work. Modularity enables the reader to zoom in immediately on those aspects he/she is interested in. If so desired, the whole work, i.e., all the related modules and if needed the necessary related information presented in meso- and macromodules, can be retrieved and read as if it were a traditional article.

We derive a list of distinctive conceptual functions for our corpus. From this analysis, we distinguish the following modules (fig. 1) based on these conceptual functions.



- *Positioning* is a complex module consisting of the module *Situation*, describing the embedding of the work, and the module *Central Problem*, stating the why of the work in question. In this complex module all the information the reader needs to know about the background of the problem in question and the particular aspects dealt with in the article, is grouped together. Separating the two constituent modules allows the reader to make a choice: to read only the *Central Problem* in case he/she is conversant with the subject, or to be introduced in the background as well by reading both constituent modules. It is immediately clear that the module *Situation*, that reviews the embedding of the work, can be replaced by a pointer, linking the work in question to a description elsewhere. Such an introduction is a typical kind of mesoscopic information. This way, the enormous redundancy of information presented in introductions of articles can be avoided. It goes without saying that the model *Central Problem* is an essential module, as this module

provides the intentions of the author of a particular article, given the context. For an informed reader, this module can play a decisive role in the decision to drop the article or to consult the rest of it as well.

- *Methods* is a complex module that can be built up from separate modules representing the theoretical, experimental, and/or numerical methods employed. If an article is one of a series, a substantial part of the information about the methods can be represented in mesoscopic modules for multiple use; e.g., in a pure experimental article using a standard instrument and employing a standard theory, both the *Experimental Method* and the *Theoretical Method* can be described elsewhere. In fact, this is already often the case. However, paper forces the author to repeat, in his/her own words, the description of the methods, whilst now a simple link suffices.
- The complex module *Results* allows readers to inspect the results without reading the whole article, for example if only a number is looked for. One of its two constituents is the module *Raw Data*. In printed articles, these data are hardly ever published, as that would require too much space. In an electronic environment, on the other hand, these data can become directly available to the reader. By doing so, the reader is able to use the data without the preferred interpretation of the originator. This enables the reader to merge his/her own data directly with the presented data for comparison and analysis. It also allows different people to apply different methods for data reduction to the same data. The second constituent of the module *Results* is the module *Treated Results.* Here the raw data are handled according to the author's choice for data reduction and further treatment. The module *Treated Results* presents the smoothed data in the usual form in figures and tables, as we are familiar with in traditional journals.
- The module *Interpretation* contains the core of the scientific reasoning in the article. Here, the author interprets the experimental results in the light of a theoretical model, for example, by comparing them with theoretical results and experimental results obtained by others. An important observation in our analysis is that it is this module that maintains most of the characteristics of a classical paper. One can argue that our procedure in fact strips the traditional article from those components that can be presented as independent entities. The remaining core, the real scientific reasoning, argumentation and conjectures, remains an essay-like text. It is this part, in fact representing knowledge rather than pure data or quantitative information, that is the most difficult to deal with.
- Within the complex module *Outcome*, we distinguish a compulsory module *Findings*, in which the author tries to answer the central questions stated in the module *Central Problem*, and an optional module *Leads to Further Research*, in which ideas and suggestions for new work are expressed. A reader who wants to learn about what happened without the how and why can simply consult the modules *Findings* and *Treated Results.*
- Besides the conceptual modules, we define a module *Meta-Information* that comprises all traditional metadata. We mention two important ingredients that are very important, given the complexity of a system of modules and links: 1) the *Abstract*, which in a modular environment has to be rethought, and 2) a clear graphical *Map of contents*. With regard to the *Abstract*, the main obstacle is that no clear theory is available about its role and content. In the standard literature many do's and dont's for writing an abstract are given, but no systematic work has been done in order to define the proper roles of an abstract as a representation of the underlying information. In a separate research programme Maarten van der Tol is tackling this problem for a modular environment (Van der Tol, 1999).

Links

In the present practice of hypertext linking, the relations between the linked objects are often left unclear to the reader. A standard hyperlink only indicates that the author has some relation in mind between, for example, a blue underlined word and something else. In a standard HTML-document full of links, we are directed from nowhere to everywhere and back.

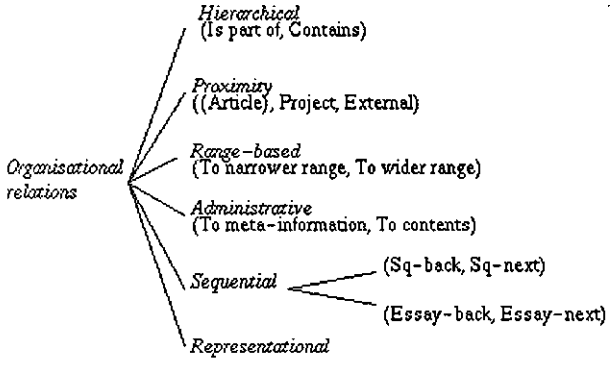
In our modular structure, a link is defined as an explicitly characterised directed connection, between modules or parts thereof (e.g., words or sentences), that represents one or more different kinds of relevant relation. Characterising links by the relations they express and by the modules they connect enables the reader, firstly, to make a well-considered choice, whether or not to follow the link and, secondly, to take the links into account in the process of locating and retrieving relevant information. This way, a link becomes a proper information object with clear characteristics. In a retrieval situation, the reader can now seek for modules and links, therewith enhancing the whole disclosure process. For this reason we also endow each link with the bibliographic data of the author who identified these relations and created the link. This way it becomes possible that a commentator on a modular article adds links to an already-published work. These links can strengthen the original work, but

they can also challenge the results by, e.g., pointing to incompatible results of others. Thus, by endowing the object "link" with the traditional bibliographic data, we ensure the authenticity and priority of each information object when new links or modules are added to published work. Links and modules now have an equal standing.

In our analysis, we identify different types of relation that are relevant in modular scientific articles, and formulate a typology for the links in the modular structure. We distinguish two main classes of relations: *organisational relations* and *scientific discourse relations.*

Organisational relations

In the class of organisational relations, which express the organisational coherence of the modular network, we distinguish the following six types of relation (fig. 2):

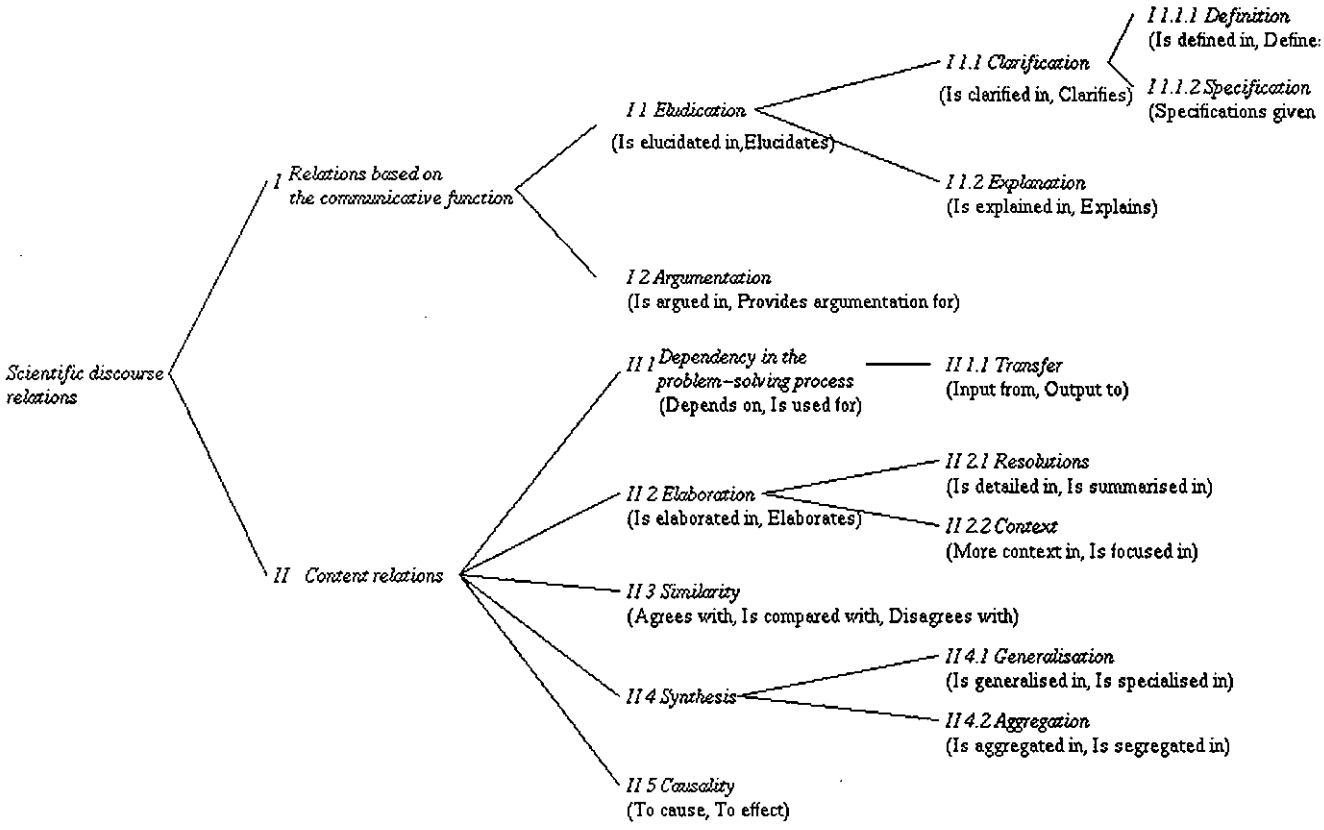


1. *hierarchical*: an asymmetric relation between complex modules and their constituent modules,
2. *proximity-based*: a symmetric relation between linked modules expressing whether they are part of the same collection (in particular, the same article or set of articles),
3. *range-based*: an asymmetric relation expressing the difference in range between linked modules,
4. *administrative*: an asymmetric relation between conceptual modules and the module representing their meta-information,
5. *sequential*: an asymmetric relation between modules linked to form a complete or a more easy-going reading path,
6. *representational*: an asymmetric relation between different representations of the same information (e.g., between texts, tables and figures).

An important aspect of links based on organisational relations is that they can often be assigned semi-automatically, provided the authors have appropriate authoring tools at their disposal.

Scientific discourse relations

The second main class of relations: scientific discourse relations, allows authors to indicate why they refer to another module or another part of the same module. Following speech communication research, we arrive at two subclasses of scientific discourse relations. One class is based on the *communicative function*; the other type consists of *Content relations* between two relata (a full overview is given in fig. 3).



1. *Communicative function* relations

The two basic aims of the author are to increase the reader's understanding of the message or to increase his/her acceptance of it. In order to understand or accept a module, readers may need additional information, for instance about the causes of a certain phenomenon. The author can make that information available to the readers by means of a link. The target of the link then consists of , e.g., a figure, a statement or a whole module, which has a particular communicative function with respect to the source of that link; for instance that of an explanation. Hence, this asymmetric relation can be made explicit by the characterisation of the link.

In practice we can often easily make a distinction between *Elucidation* links and *Argumentation* links. In the case of elucidation, the aim is at increasing the reader's understanding. Within the *Elucidation* relations, we make a further distinction between *Explanation* and *Clarification*. An explanation is given when the author anticipates that part of the intended readership will not understand how a particular state of affairs has come into being. When the author anticipates that part of the intended readership will not understand what he/she means by a particular text or figure, he/she will make a clarification available in the module or through a link to another module. A further refinement is then possible between a *Definition* relation and a *Specification* relation. Thus, the author can, for instance, connect a difficult term to an "encyclopaedic" macroscopic module by a link expressing a *Definition* relation.

In the argumentative case, the aim is to increase the reader's acceptance of a standpoint. These are cases where the author can presume that not every reader of the indented readership will immediately accept a particular statement.

2 *Content* relations

The second subclass of scientific discourse relations comprises *Content* relations, such as *Dependency*, *Elaboration*, *Similarity*, *Synthesis* and *Causality*.

The *Dependency* in the problem-solving process of the reported research is an asymmetric relation between steps in that process. A link can express the fact that the source depends on the target in the way in which, for instance,

results depend on to generated them. A special case is a *Transfer* relation, if items are taken from one module and included in another. This is often the case with mathematical formulae or values that are used as input in calculations.

With an *Elaboration* relation, we indicate an asymmetric relation where the target contains an elaboration of the statement in the source. A mesoscopic sketch of the *Situation* can provide more information than a short statement in a *Situation* module at the microscopic level. Within this class, we can make a further distinction between *Resolution* relations that point to more fine-grained information, i.e. more details, and *Context* relations, pointing to more broad sweeping accounts of the subject, i.e. mor context. We link information that is similar in relevant details, e.g., results of the same kind of investigation by different authors, by links expressing *Similarity* relations.

In the case of *Synthesis* relations we deal with: a) *Aggregation* expressed in links in which the source of the link is a component of the target, and (b) *Generalisation*, where more-or-less the same concepts are grouped together (for instance in the case where, on the microscopic level, specific parameters of an apparatus are fully described in an *Experimental Methods* meso-module).

As a final example we identify the *Causal* relations in which clear cause and effect relations are covered.

Applicability of our model

We developed the model in conjunction with an analysis of a corpus of articles published by a single research group in the field of experimental molecular dynamics. However, a short inspection of examples of publications in other domains showed that modular structures for other types of publications could be derived from our model.

To test the model, we rewrote two strongly related articles from our corpus as modular electronic article(demo in progress). Although the modular framework is explicitly intended for the creation and evaluation of new work, we found, recasting old work in the new mould, that modular electronic articles can meet our pre-defined requirements better than linear articles. In particular:

- The possibility of multiple usage enhances the author's efficiency
- The explicit labelling of modules and links allows for better information retrieval.
- The reader can selectively locate, retrieve and consult precisely those parts of the published works that are relevant, so that the reader's efficiency is increased.
- As the modular structure is more systematic and explicit, modular publications can be clearer than linear ones.

Acknowledgements

This work is part of the 'Communication in Physics' project of the Foundation Physica; it is financially supported by the Foundation Physica, the Shell Research and Technology Centre Amsterdam, the Royal Dutch Academy of Sciences, the Royal Library, and Elsevier Science NL.

Bibliographic references

(Garvey, 1979) W.D. Garvey, *Communication: the essence of science - Facilitating information exchange among librarians, scientists, engineers and students*. (Pergamon Press, Oxford, 1979)

(Meadows, 1998) A.J. Meadows, *Communicating research*, (Academic Press, San Diego, 1998)

(Van Eemeren et al., 1993) Eemeren, F.H. van, R. Grootendorst, Sally Jackson and Scott Jacobs, *Reconstructing argumentative discourse. Studies in rhetoric and communication*. (The Univerity of Alabama Press, Tuscaloosa, 1993)

(Kircz, 1998) J.G. Kircz, Modularity: the next form of scientific information presentation? *Journal of Documentation*, Vol.54,no.2,March 1998, p.210-235. Electronic version: http://www.wins.uva.nl/projects/commphys/papers/jkmodul.htm

(Harmsze et al., 1996) F.A.P. Harmsze, M. van der Tol and J. Kircz, 'Naar een modulair model voor natuurwetenschappelijke informatie in elektronische artikelen. In: *Informatiewetenschap 1996, Wetenschappelijke bijdragen aan de Vierde Interdisciplinaire Conferentie Informatiewetenschap (Delft, 13 december 1996)*. Van der Meer (Werkgemeenschap Informatiewetenschap, 1996).pp. 53-71. Electronic vérsion: http://www.wins.uva.nl/projects/commphys/papers/delft/delft.htm

(Harmsze, 2000) F.A.P. Harmsze, *A modular structure for scientific articles in an electronic environment*, PhD thesis, to be published, Amsterdam 2000

(Van der Tol, 1999) M.C. van der Tol, The abstract as an orientation tool in modular electronic articles. To be published in the proceedings of the First International Conference on Document Design, Tilburg, December 17 and 18, 1998 Electronic version: http://www.wins.uva.nl/projects/commphys/papers/docdes/docdes.html

User Modeling in Adaptive Hypermedia Applications

Hongjing Wu, Geert-Jan Houben[1], Paul De Bra

Department of Computing Science
Eindhoven University of Technology
PO Box 513, 5600 MB Eindhoven
the Netherlands

phone: +31 40 2472733
fax: +31 40 2463992
email: {hongjing,houben,debra}@win.tue.nl

Abstract: A hypermedia application offers its users a lot of freedom to navigate through a large hyperspace. The rich link structure of the hypermedia application can not only cause users to get lost in the hyperspace, but can also lead to comprehension problems because different users may be interested in different pieces of information or a different level of detail or difficulty. Adaptive hypermedia systems (or AHS for short) aim at overcoming these problems by providing adaptive navigation support and adaptive content. The adaptation is based on a *user model* that represents relevant aspects about the user. The adaptive navigation support and the adaptive content will reduce the orientation problems and the comprehension problems respectively.

The AHS AHA [DC98], that we developed at the Eindhoven University of Technology, and its predecessors [CD97, DC97], did only support user knowledge at the page level and not on the level of large abstract concepts, which limits adaptation to a low knowledge level. Other current AHS appear to suffer from similar problems. As part of the redesign process for AHA we have developed a reference model for the architecture of adaptive hypermedia applications, named AHAM (for Adaptive Hypermedia Application Model) [DHW99], which is an extension of the Dexter hypermedia reference model [HS90, HS94]. In AHAM knowledge is represented through hierarchies of large composite abstract concepts as well as small atomic ones. AHAM also divides the different aspects of an AHS into a *domain model* (DM), a *user model* (UM) and a *teaching model* (TM). This division provides a clear separation of concerns when developing an adaptive hypermedia application.

In this paper, we specifically concentrate on the user modeling aspects of AHAM, but also describe how they relate to the domain model and the teaching model. First, we introduce general concepts involved in adaptive hypermedia applications and their design and use the AHAM reference model to describe the different aspects, specially domain model, user model and teaching model. We show how user features are modeled and how the actual adaptation process is performed. We illustrate this general approach for user modeling and adaptation by considering the AHS AHA.

Keywords: adaptive hypermedia, user modeling, adaptive presentation, adaptive navigation, hypermedia reference model

---

[1] On leave at the Department of Mathematics and Computer Science of the University of Antwerp (UIA), Belgium.

1. Introduction

Hypermedia systems in general and Web-based systems in particular are becoming increasingly popular as tools for user-driven access to information. One of the characteristic properties of hypermedia applications is that they offer users a lot of freedom to navigate through a large hyperspace. By choosing between hyperlinks users can follow very different paths through the hyperspace in order to access the contents contained in that hyperspace. Unfortunately, this rich link structure of the hypermedia application causes some serious usability problems:

- A typical hypermedia system always presents the same links on a page, regardless of the path a user followed to reach this page. When the system wants to providing navigational help, it does not know which part of the link structure is most important for the user. So, if for example it wants to provide a map of the structure, it does not know exactly what is relevant for this particular user: the map cannot be simplified by filtering (or graying) out links that are less relevant for the user. Not having personalized maps is a typical *navigation problem* of hypermedia applications.
- Navigation in ways that the author did not anticipate also causes *comprehension problems*. For every page the author makes an assumption about the user's foreknowledge. However, there are too many ways to reach a page to make it possible for an author to anticipate all possible variations in foreknowledge when a user visits that page. Therefore, a page is always presented in the same way. This often results in users visiting pages containing a lot of redundant information for them ("too much information") and pages that they cannot fully understand because they lack some expected foreknowledge ("too little information or wrong information").

Adaptive hypermedia systems (or AHS for short) aim at overcoming these problems by providing adaptive navigation support and adaptive content. Adaptive hypermedia is a recent area of research on the crossroad of hypermedia and the area of user-adaptive systems, with applications in educational applications, on-line information systems, on-line help systems, information retrieval systems, etc. The goal of this research is to improve the usability of hypermedia systems by making them personalized. The personalization or adaptation is based on a *user model* that represents relevant aspects about the user. The system gathers information about the user by observing the use of the application, and in particular by observing the *browsing* behavior of the user. An overview of systems, methods and techniques for adaptive hypermedia can be found in [B96].

AHA [DC98] is an AHS system developed out of Web-based courseware for an introductory course on hypermedia at Eindhoven University of Technology. In AHA knowledge is considered at the same level of abstraction as the contents: every concept is represented by a page and knowledge represents whether or not a user knows the concept associated with a page. The user's knowledge about a given concept is a binary value: *known* or *not known*.

AHAM (for Adaptive Hypermedia Application Model) [DHW99] is a reference model for the architecture of adaptive hypermedia applications. It is an extension of the Dexter hypermedia reference model [HS90, HS94], and evolved partially from a redesign process of AHA. The central issue in AHAM is the fact that performing both "useful" and "usable" adaptation in a given application depends on three factors:

- There must be a *domain model* that describes how the information content is structured in terms of relationships between high level and low level concepts, and that indicates how these abstract concepts are tied to pages (that can actually be presented).
- By observing the user's behavior a fine-grained *user model* must be maintained to represent the user's preferences, knowledge (related to concepts from the domain model),

goals, navigation history and possibly other relevant aspects.

- The adaptation of the presentation of both content and link structure to the user's preferences and knowledge level is performed by the system based on some "intelligence". In most AHS this intelligence is default (implicit in the system), but AHAM recognizes that an author can explicitly provide a *teaching model* consisting of *pedagogical rules*.

The key elements in AHAM are thus the *domain model* (DM), *user model* (UM) and *teaching model* (TM). This division of adaptive hypermedia applications provides a clear separation of concerns when developing an adaptive hypermedia application. The main shortcoming in many current AHS is that these three factors or components are not clearly separated:

- The relationship between pages and concepts is too vague (e.g. [PDS98]) or too strict (e.g. [DC98]).
- The pedagogical rules can not be defined at the conceptual level but only at the page level (e.g. [DC98], [BSW96a], [BSW96b]).
- There is a mismatch between the high level of detail in the user model and the low reliability of the information on which an AHS must update that user model: for example, do access times represent reading times?

In this paper, we concentrate on the user modeling aspects of AHAM, but also describe how these aspects relate to the domain model and the teaching model. The rest of the paper is organized as follows. In Section 2 we introduce AHAM as a reference model for the design of hypermedia applications. A more detailed description can be obtained in [DHW99]. Note that some information on authoring support for adaptive hypermedia applications (in the context of AHAM) is available in [WHD99]. In Section 3 we focus on the aspect of user modeling and its influence on the adaptation. We show how user features are represented using attribute/value pairs in the context of AHAM. We describe the role of *knowledge values* in the representation of the user's knowledge on concepts from the domain model. We also illustrate how the actual adaptation process is based on the user model and how the maintenance of the user model can be facilitated. As a concrete example of the use of AHAM, Section 4 addresses the specific details of user modelling and adaptation in the AHA system. The way in which the domain model, user model and teaching model are implemented in this specific system are described. Note that this has led to a better insight in the consequences of some original design decisions, which resulted in a number of improvements to the original system. Using this approach we have thus been able to pinpoint some shortcomings in current adaptive hypermedia systems, not just in AHA. Section 5 concludes and indicates some future developments.

2. AHAM, a Dexter-based Reference Model

The most important aspects of hypermedia applications are the information nodes and the link structure connecting these nodes. In the Dexter reference model [HS90, HS94] this is captured in what Dexter calls the *Storage Layer*. This layer represents the application author's view on the application domain in terms of concepts. We call this view the *domain model* DM.

In adaptive hypermedia applications the central role of DM is shared with a *user model* UM. UM represents the relationship between the user and DM by keeping track of how much the user knows about each of the concepts in the application domain.

In order to perform adaptation (based on DM and UM) the author needs to specify how the user's knowledge influences the presentation of the information (from DM). In AHAM this is expressed by means of a *teaching model* TM consisting of pedagogical rules: the rules in TM model the explicit "intelligence" that the author wants the system to use in the adaptation process. An adaptive engine (as part of the AHS) then uses these rules to manipulate link anchors (from the Dexter

model's *Anchoring*) and to generate what the Dexter model calls the *Presentation Specifications*. Figure 1 shows the global structure of adaptive hypermedia applications in the AHAM model, just like Dexter focusing on the Storage Layer.

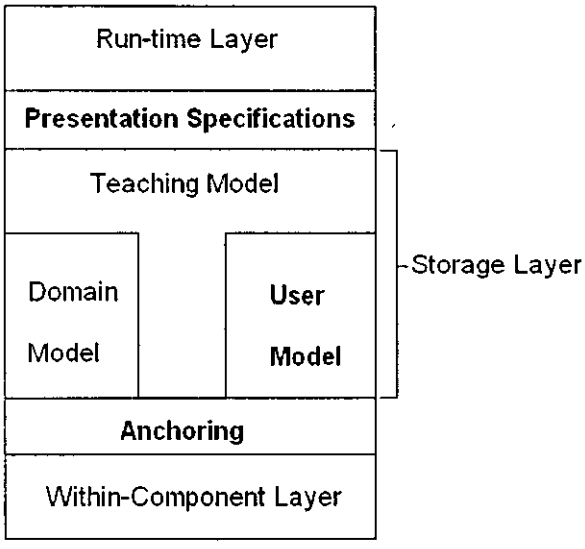


Figure 1: global structure of adaptive hypermedia applications.

2.1 The domain model

A *component* is an abstract notion in an AHS. It is a pair (uid, cinfo) where uid is a globally unique (object) identifier for the component and cinfo represents the component's information that consists of:

- a set of attribute-value pairs;
- a sequence of anchors (for attaching links);
- a presentation specification.

A *concept* is a component representing an abstract information item from the application domain. An *atomic concept* corresponds to a fragment of information: these are the primitive (non-adaptable) information units in the model with attribute and anchor values that belong to the Within-Component Layer. A *composite concept* has two "additional" attributes:

- a sequence of children (concepts);
- a constructor function (to denote how the children belong together).

There are a number of constraints. For example, the children of a composite concept are all atomic concepts (then it is a *page* (sequence of fragments) or in typical hypertext terminology a *node*) or they are all composite concepts. Figure 2 illustrates a part of a concept hierarchy.

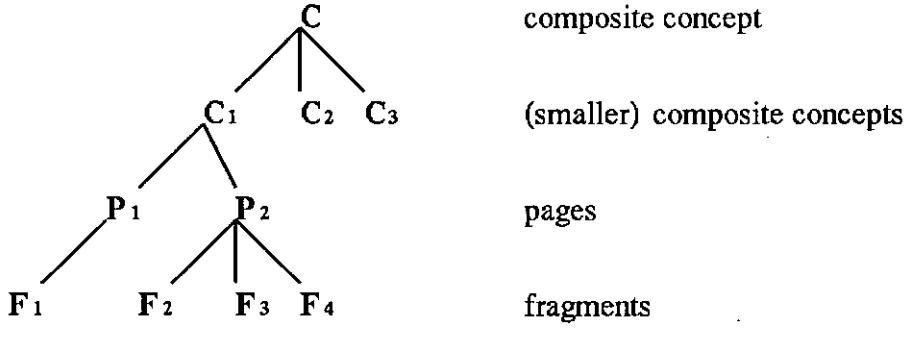


Figure 2: Example concept hierarchy.

An *anchor* is a pair (aid, avalue), where aid is a unique identifier for the anchor within the scope of its component and avalue is an arbitrary value that specifies some "location" within a concept component. Anchor values of atomic concepts belong to the Within-Component Layer, while anchor values of composite concepts are identifiers of concepts that belong to that composite.

A *specifier* is a tuple (uid, aid, dir, pres), where uid is the identifier of a concept, aid is the identifier of an anchor, dir is a direction (FROM, TO, BIDIRECT, or NONE), and pres is a presentation specification.

A *concept relationship* is a component, with two additional attributes:

- a sequence of specifiers;
- a concept relationship type.

The most common type of concept relationship is the type **link**, which corresponds to links in most hypermedia systems (with typically at least one FROM element and one TO or BIDIRECT element). In AHAM we consider other types of relationships as well, which play a role in the adaptation, such as for example relationships of the type **prerequisite**. When C_1 is a prerequisite for C_2 it means that the user should read C_1 before C_2. It does not mean that there must be a link from C_1 to C_2. It only means that the system somehow takes into account that reading about C_2 is not desired before some (enough) knowledge about C_1 has been acquired.

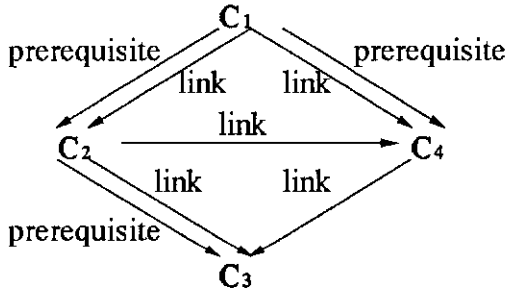


Figure 3: Example concept relationship structure.

Together, the atomic concepts, composite concepts and concept relationships build the *domain model* DM of an application.

2.2 The user model

An AHS associates a number of *user model attributes* with each concept component of DM. For every user the AHS maintains a *table-like structure*, in which for each concept the (user-specific) attribute values for that concept are stored. Note that these attribute values do not necessarily just represent the *knowledge level* of a concept. In Section 3 we look at the user model in more detail.

Note that since the user model consists of "named entities" for which we store a number of attribute/value pairs, there is no reason to limit these "entities" to pure *concepts* and their *knowledge level*. These concepts can be used to represent other user features, such as preferences, goals, background and hyperspace experience. For the AHS (and AHAM) the actual meaning of concepts is irrelevant.

2.3 The teaching model

The adaptation of the information content and of the link structure is based on a set of *rules*. In an application these rules build the connection between DM, UM and the presentation (specification) to be generated. Consider the following example:

- In DM one expresses that a concept C_1 is a prerequisite for concept C_2, and that there is a hypertext link from C_1 to C_2.
- In UM one expresses how much the user knows about concept C_1 (and C_2).

A rule is then needed to determine how "relevant" or "desirable" concept C_2 is, depending on the knowledge level about C_1. Other rules then express the effect of "relevance" on *adaptive navigation* (AN) and *adaptive presentation* (AP):

- (AN) One rule expresses how the relevance of C_2 influences the presentation of links in graphical maps.
- (AN) Another rule expresses how the relevance of C_2 influences the presentation of link anchors for links leading to C_2.
- (AP) The relevance of C_2 may influence the presentation of C_2 itself. For instance, an undesirable fragment may be hidden or grayed out.
- (AP) The relevance of elements of a composite may influence their selection or order. Fragments on a page may be sorted from most relevant to least relevant.

A *generic pedagogical rule* is a tuple (R, CRT, PH, PR), where R is a "triggered" rule, CRT is an optional concept relationship type, PH is the "phase" for the execution of the rule (either **pre** or **post**) and PR is a Boolean "propagate" field which indicates whether this rule may trigger other rules (to avoid infinite loops). The phase **pre** is executed during the generation of the page, while **post** is executed afterwards and is used for updating the user model. If a generic rule contains more than one concept it must have a CRT and then applies to all concept relationships of type CRT. Rules about just one concept do not have a CRT.

The syntax of the permissible rules depends on the AHS. Generic pedagogical rules are often system-defined, meaning that an author does not need to specify them. Author-defined rules always take precedence over (conflicting) system-defined rules. (Note that some AHS do not provide the possibility for authors to define their own generic pedagogical rules.)

A *specific pedagogical rule* is a tuple (R, PH, PR), where rule R uses (concrete) concepts from DM. (PH and PR are as for generic rules.) While specific rules are typically used to create exceptions to generic rules they can also be used to perform some ad-hoc adaptation based on concepts for which DM does not provide a relationship. Specific pedagogical rules must always be defined by the author. They take precedence over generic rules that would apply to the same concepts.

The *teaching model* TM of an AHS is the set of (generic and specific) pedagogical rules.

An AHS does not only have a DM, UM and TM, but also an *adaptive engine* AE. The adaptive engine provides the implementation dependent aspects while DM, UM and TM describe the information and adaptation at the conceptual, implementation independent level. It is a software environment that performs the following functions:

- It offers generic page selectors and constructors (used to determine which page to display when the user follows a link to a composite concept, or how to build a presentation for a page).
- It optionally offers a (very simple programming) language for describing new page selectors and -constructors.
- It performs adaptation by executing the page selectors and constructors (e.g. selecting a page, selecting fragments, manipulating link anchors).
- It updates the user model (instance) each time the user visits a page (by triggering the necessary pedagogical rules for the **post** phase).

An *adaptive hypermedia application* is a 4-tuple (DM, UM, TM, AE), where DM is a domain

model, UM is a user model, TM is a teaching model, and AE is an adaptive engine.

3. User Modeling and Adaptation in AHAM

3.1 Representation of user features using (attribute/value) pairs

By definition adaptive hypermedia applications reflect some features of the user in the user model. This model expresses various visible aspects of the system that depend on the user and that are visible to that user. Generally, there are five user features that are used by existing AHS [B96]:

- knowledge,
- user goals,
- background,
- hyperspace experience,
- preferences.

Almost every adaptive presentation technique relies on the user's *knowledge* as a source of adaptation. The system has to recognize the changes in the user's knowledge state and update its user model accordingly. The user's knowledge is often represented by an overlay model (overlay over the concepts from DM) or a (simpler) stereotype user model. As many adaptation techniques require a rather fine-grained approach, stereotype models are often too simple to provide adequate personalization and adaptation. Overlay models on the other hand are generally hard to initialize. Acceptable results are often achieved by combining the two kinds of modeling: starting with stereotype modeling and then moving towards a more fine-grained overlay model. Using the AHAM definition for user model, it is fairly straightforward how a user's knowledge state can be represented by associating a *knowledge value* attribute to each concept.

Apart from the concept's identifier (which may be just a name) a typical AHS will store not only a *knowledge value* for each concept, but also a *read* value which indicates whether (and how much) information about the concept has been read by the user, and possibly some other attribute values as well. While the model uses a table representation, implementations of AHS may use different data structures. For instance, a logfile can be used for the *read* attribute. The table below illustrates the (conceptual) structure of a user model for a course on hypermedia.

concept name (uid)	Knowledge value	read	...
Xanadu	well learned	true	...
KMS	Learned	true	...
WWW-page1	well learned	true	...
WWW-page2	not known	false	...
WWW	Learned	false	...
...	...	...	...

Table 1: Example user model (instance).

Note that AHAM's user model UM has enough expressive power to model all user features that current AHS take into account: the knowledge value of a concept can be a Boolean, discrete or continuous value depending on the choice of the author.

The second kind of user feature is the user's *goal.* The user's goal or task is a feature that is related with the context of the user's working activities rather than with the user as an individual. It is the most volatile of all user features. One possible representation of the user's current goal uses a set of pairs (Goal,Value), where Value is the probability that Goal is the current goal of the user. This representation perfectly matches the way in which AHAM models the user's state.

Two features of the user that are similar to the user's knowledge of the subject but that functionally differ from it, are the user's *background* and the user's *experience* in the given hyperspace. By background we mean all the information related to the user's previous experience *outside* the subject of the hypermedia system. By user's experience in the given hyperspace we mean how familiar is the user with the structure of the hyperspace and how easy can the user navigate in it. Again, these features can be modeled in AHAM using concepts' attribute/value pairs.

For different possible reasons the user can prefer some nodes and links over others or some parts of a page over others. This is used most heavily in information retrieval hypermedia applications. In fact in most adaptive information retrieval hypermedia applications *preferences* are the only information that is stored about the user. These user features differ from other ones, since in most cases they cannot be deduced by the system: the user has to inform the system directly or indirectly about the preferences. AHAM's concepts' attribute/value pairs can again be used to model the user's preferences.

From the above descriptions we can conclude that although a user model needs to represent (five) very different aspects of a user, all of these kinds of aspects can be implemented as sets of *concepts with associated attribute/value pairs.*

3.2 Adaptation based on the user model

The adaptive engine realizes adaptive presentation and adaptive navigation according to the pedagogical rules that are either system-defined or written by the author and that depend on the user model. Below we give a number of examples to show how pedagogical rules are used to do adaptation. The syntax used for the rules is arbitrary and only exemplary: AHAM does not prescribe any specific syntax. Normally every AHS will provide its own syntax for defining pedagogical rules.

Example 1 *For atomic concepts (fragments) let us assume that the presentation specification is a two-valued (almost Boolean) field, which is either "show" or "hide". When a page is being accessed, the following rule sets the visibility for fragments that belong to the page, depending on their "relevance" attribute-value.*

< access(C) and F IN C.children and F.relevance = true => F.pres := show, pre, false >

Here we simplified things, by assuming that we can treat C.children as if it were a set, whereas it really is a sequence. It is common to execute rules for generating presentation specifications in the **pre** phase, as done in this example.

Example 2 *The following rules set the presentation specification for a specifier that denotes a link (source) anchor depending on whether the destination of the link is considered relevant and whether the destination has been read before. For simplicity we consider a link with just one source and one destination.*

< CR.type = link and CR.cinfo.dir[1] = FROM and CR.cinfo.dir[2] = TO and CR.ss[2].uid.relevant = true and CR.ss[2].uid.read = false => CR.ss[1].pres = GOOD, pre, false >

< CR.type = link and CR.cinfo.dir[1] = FROM and CR.cinfo.dir[2] = TO and CR.ss[2].uid.relevant = true and CR.ss[2].uid.read = true => CR.ss[1].pres = NEUTRAL, pre, false >

< CR.type = link and CR.cinfo.dir[1] = FROM and CR.cinfo.dir[2] = TO and CR.ss[2].uid.relevant = false => CR.ss[1].pres = BAD, pre, false>

These rules say that links to previously unread but "relevant" pages are "GOOD". Links to previously read and "relevant" pages are "NEUTRAL" and links to pages that are not "relevant" are "BAD". In the AHA system [DC98] this results in the link anchors being colored blue, purple or black respectively (unless the user changes these preferences), while in ELM-ART [BSW96a] and Interbook [BSW96b] the links would be annotated with a green, yellow or red ball.

3.3 Maintenance of user model

To record the reading history of the user and the evolution of the user's knowledge, the system updates the user model based on the observation of the user's browsing process. The rules that the author has defined in TM describe how to keep track of the evolution of the user's knowledge.

Example 3 *The following rule expresses that when a page is accessed the "read" user-model attribute for the corresponding concept is set to true in the post phase:*

< access(C) => C.read := true, post, true >

This rule also says (through PR being true) that it will trigger other rules that have *read* on their left-hand side.

Example 4 *The following rule expresses that when a page is "relevant" and it is accessed, the knowledge value of the corresponding concept becomes "well-learnt" in the pre phrase. This is somewhat like the behavior of Interbook [BSW96b].*

< access(C) and C.relevant = true => C.knowledge := well-learnt, pre, true>

In Interbook, as well as in AHA [DC98], knowledge is actually updated in the **pre** phase. At the end of Section 4 we shall describe why this option is chosen, and which problems it creates. In general one wishes to have the option to base some adaptation on the knowledge state *before* accessing a page and some adaptation on the knowledge state *after* reading the page.

4. User Modeling and Adaptation in the AHA system

AHA [DC98] is a simple adaptive hypermedia system. We first describe the properties of the version that is currently being used for two on-line courses and one on-line information kiosk. At the end of this section we indicate planned changes for the next version of AHA.

- In AHA the *domain model* consists of three types of concepts: *abstract concepts*, *fragments* and *pages*. Concepts are loosely associated with (HTML) pages, not with fragments.
- The *user model* consists of:
  - Color preferences for link anchors which the user can customize. (These preferences result in "non-relevant" link anchors to be hidden if their color is set to black, or visibly "annotated" if their color is set to a non-black color, different from that of "relevant" link anchors.)
  - For each abstract concept, a Boolean *knowledge* attribute. (*True* means the concept is known, *false* means it is not known.)

- For each page, a Boolean *read* attribute. (*True* means the page was read, *false* means it was not read.) AHA actually logs access and reading times, but they cannot be used in a more sophisticated way in the current version.

- AHA comes with a *teaching model* containing system-defined generic pedagogical rules. It offers a simple language for creating author-defined specific pedagogical rules (but no author-defined generic rules).

The *domain model* can only contain concept relationships of the types that are shown below. The influence of these relationships on the adaptation and the user model updates is defined by system-defined generic pedagogical rules. In AHA all rules are executed in the **pre** phase and are triggered directly by a page access, thus eliminating the need for propagation (causing problems that we describe later).

- When a page is accessed, its *read* attribute in the user model is updated:

  < access(P) => P.read := true, pre, false >

- The relationship type *generates* links a page to an abstract concept. A *generates* relationship between P and C means that reading page P generates knowledge about C:

  < access(P) => C.knowledge := true, pre, false >

  This "generation" of knowledge in AHA is controlled by a structured comment in an HTML page:

  ```
  <!-- generates readme -->
  ```

  This example *generates* comment denotes that the concept `readme` becomes known when the page is accessed.

- The relationship type *requires* links a page to a composite concept which is defined by a Boolean expression of concepts. Although in principle this composite concept is unnamed, we shall use a "predicate" or "pseudo attribute of the page" to refer to it: *P.requires* is used as a Boolean attribute of which the value is always that of the corresponding Boolean expression. A *requires* relationship is implemented using a structured comment at the top of an HTML page, e.g.:

  ```
  <!-- requires ( readme and intro ) -->
  ```

  This example expresses that this page is only considered relevant when the concepts `readme` and `intro` are both known. In AHA, links to a page for which *requires* is *false* are considered BAD, and reading such a page does not generate knowledge. Below we give the rules that determine how the link anchors will be presented. They are very similar to the rules in Example 2 (Subsection 3.2):

  < CR.type = link and CR.cinfo.dir[1] = FROM and CR.cinfo.dir[2] = TO and CR.ss[2].uid.requires = true and CR.ss[2].uid.read = false => CR.ss[1].pres = GOOD, pre, false >

  < CR.type = link and CR.cinfo.dir[1] = FROM and CR.cinfo.dir[2] = TO and CR.ss[2].uid.requires = true and CR.ss[2].uid.read = true => CR.ss[1].pres = NEUTRAL, pre, false >

  < CR.type = link and CR.cinfo.dir[1] = FROM and CR.cinfo.dir[2] = TO and CR.ss[2].uid.requires = false => CR.ss[1].pres = BAD, pre, false>

- The relationship type *link* only applies to pairs of pages in AHA.

In AHA structured HTML comments are used for specifying author-defined specific pedagogical rules about the conditional inclusion of fragments in HTML pages. With a fragment F we can associate a "pseudo attribute" *requires* to indicate the condition, just like for whole pages. The syntax is illustrated by the following example:

```
<!-- if ( readme and not intro ) -->
... here comes the content of the fragment ...
<!-- else -->
... here is an alternative fragment ...
<!-- endif -->
```

AHA only includes fragments when their *requires* "attribute" is *true*.

We conclude this section with an illustration of one specific shortcoming which we have found in both AHA [DC98] and Interbook [BSW96b]: the "new" knowledge values are calculated in the **pre** phase (and in fact these systems do not support a two-phase approach at all). When a user requests a page, the knowledge generated by reading this page is already taken into account during the generation of the page. This has desirable as well as undesirable side-effects:

- When links to other pages become relevant *after* reading the current page it makes sense to already annotate the link anchors as relevant when presenting the page. Once a page is generated its presentation remains static while the user is reading it (and rightfully so). The new knowledge thus needs to be taken into account before the page is actually read.

- Pages contain information that becomes relevant or non-relevant depending on the user's knowledge. In some cases the relevance of a fragment may depend on the user having read the page that contains this fragment. This means that a fragment may be relevant the first time a page is visited and non-relevant thereafter, or just the other way round. By already taking into account the knowledge before the page is generated for the first time a different "first time version" becomes impossible to create.

5. Conclusion

We have developed a reference model for adaptive hypermedia applications, named AHAM. The description of adaptive applications in terms of AHAM has provided us with valuable redesign issues for the AHS AHA. The two most important ones are:

- The division of an adaptive hypermedia application into a *domain model*, *user model*, and *teaching model* provides a clear separation of concerns and will lead to a better separation of orthogonal parts of the AHS functionality in the implementation of the next version of AHA. We believe that any system which supports this separation of concerns will not only result in a cleaner implementation, but also in a more usable authoring environment.
- By representing AHA in the AHAM model we have identified another shortcoming: the lack of a two-phase application of rules. We found that this shortcoming is present in other AHS as well.

In this paper we have focused on the user modeling aspects of AHAM, and have described how they relate to the domain model and the teaching model. We have shown how user features are modeled and how the actual adaptation process is performed. We have illustrated this general approach for user modeling and adaptation by explicitly considering the AHS AHA. The description of these user model aspects at an abstract level sets AHAM apart from other descriptions of AHS which are too closely related to the actual implementation of these AHS.

References

[B96] Brusilovsky, P., "Methods and Techniques of Adaptive Hypermedia". User Modeling and User-Adapted Interaction, 6, pp. 87-129, 1996. (Reprinted in Adaptive Hypertext and Hypermedia, Kluwer Academic Publishers, pp. 1-43, 1998.)

[CD97] Calvi, L., De Bra, P., "Using Dynamic Hypertext to create Multi-Purpose Textbooks". Proceedings of ED-MEDIA'97, Calgary, pp. 130-135, 1997.

[DC97] De Bra, P., Calvi, L., "Creating adaptive hyperdocuments for and on the Web". Proceedings of the WebNet'97 Conference, Toronto, pp. 149-165, 1997.

[DC98] De Bra, P., Calvi, L., "AHA: a Generic Adaptive Hypermedia System". Proceedings of the Second Workshop on Adaptive Hypertext and Hypermedia, Pittsburgh, pp. 5-11, 1998.

[DHW99] De Bra, P., Houben, G.J., Wu, H., "AHAM: A Dexter-based Reference Model for Adaptive Hypermedia". Proceedings of ACM Hypertext'99, Darmstadt, pp. 147-156, 1999.

[HS90] Halasz, F., Schwartz, M., "The Dexter Reference Model". Proceedings of the NIST Hypertext Standardization Workshop, pp. 95-133, 1990.

[HS94] Halasz, F., Schwartz, M., "The Dexter Hypertext Reference Model". Communications of the ACM, Vol. 37, nr. 2, pp. 30-39, 1994.

[PDS99] Pilar da Silva, D., "Concepts and documents for adaptive educational hypermedia: a model and a prototype", Proceedings of the Second Workshop on Adaptive Hypertext and Hypermedia, Pittsburgh, pp. 33-40, 1998.

[WHD99] Wu, H., Houben, G.J., De Bra, P., "Authoring Support for Adaptive Hypermedia", Proceedings ED-MEDIA'99, Seattle, pp. 364-369, 1999.

Adaplix: Towards Adaptive Websites

Nico Jacobs

K.U.Leuven
Dept. of Computer Science
Celestijnenlaan 200 A
B-3001 Heverlee, Belgium
Nico.Jacobs@cs.kuleuven.ac.be

Abstract

Adaptive webpages help the user in finding relevant information by tailoring their content and lay-out specific to the visiting user. In this paper we describe Adaplix, a system we implemented to extend HTML by introducing conditional statements. We describe how this system can be extended with an inductive logic programming component to learn the user's browsing preferences. We also report on some preliminary experiments with this learning component.

1 Introduction

More and more people go on the Internet to look for information. These often unexperienced users have difficulties to find the information they're looking for. There are several techniques that try to alleviate this problem. Search engines (e.g. Altavista) try to index as many webpages as possible. This approach suffers from the fact that a query often results in too many suggested webpages of which many are irrelevant to the subject. Portal sites (e.g. Yahoo!) classify sites in a hierarchical directory of subjects, which solves the problems of the previous approach at the cost of creating and maintaining such a directory.

Once an interesting website has been found, the user still has to navigate through the site to find the information she's looking for. Adaptive webpages can help to solve this problem. The idea behind adaptive webpages is to let a page adapt itself to the user. The techniques mentioned in the previous paragraph try to help the user on a global level (starting from nothing but some keywords, return relevant webpages). Adaptive webpages however help the user on a much more local level: given that the user wants to see this page, how can the information on that page be changed (content as well as lay-out) so that it is as interesting for this user as possible.

In this paper we first introduce the Adaplix system (section 2) which allows the author of a website to use conditional statements in the webpages. In the

next step (section 3) we extend this system with a component which learns the browsing preferences of users using the Tilde [3] system and we discuss (section 4) some experiments. In section 5 we briefly compare this approach with other systems and conclude.

2 Adaplix

In the field of adaptive hypermedia there are two major approaches. One consists of letting the author of a document add metadata to the document which describes how the document should be altered based on a model of the user (e.g. don't show this link unless the user is already familiar with concept C). The clear advantage of this approach is that the author can use all his domain knowledge to make the document as useful as possible. Maintaining the user model however is not an easy task. Consider the example above, how do we know the user is familiar with concept C? Usually the author of a document has to provide information describing how the user model should be changed (e.g. when the user is already familiar with concept B and has consulted document A, then the user is familiar with concept C). [5] has more information on this approach. An other approach (used in systems such as WebWatcher [8]) is to use machine learning techniques to learn the interest of a user and then automatically alter each document based on these interests (e.g. highlighting interesting links). The advantage of this approach is that there is no need for an expert to manually denote all the possible changes that can be made when presenting a document, which is particularly interesting when the content of the document changes very often (e.g. a website with news articles). However these systems are only capable of small adaptations of the presented information such as highlighting information or adding links to similar documents.

We developed Adaplix, a prototype which combines both approaches. The author of a webpage can add extra information about how a page can be adapted to the user. Adaplix also tries to learn the browsing preference of the user. In this way, the author can create pages with constructions such as "if the user would be interested in page A but is not interested in page B then show link C" where the learning component decides upon the interest of the user in pages A and B based upon the browsing behaviour of that user in the past. In this section we describe how Adaplix allows the author to alter pages, whereas the learning component is described in the next section.

Information on the World Wide Web (WWW) is offered by http servers. A browser (client) can contact the server to retrieve information. The server will then look up the requested webpage and send it to the client which displays this information. In order to obtain adaptive webpages, one has to alter this scheme in that the static information stored on the server has to be altered before it is presented to the user. The information can be altered either in the server or in the client. Altering the page at the server side is easier to implement and maintain because all programs only need to run on one machine (no need for platform independent implementation plus the possibility for code optimalisation) and all data is stored on one machine. Moreover, the adaptive webpages can be viewed

```
<!-- #adaplang {if (recstring ''BirthYear'') > 1990
  then print ''<BODY BACKGROUND="colordots.gif"''
  else print ''<BODY BACKGROUND="greyrelief.gif"''}>
<P><FONT SIZE=+1>W</FONT>elcome to my homepage.
```

Figure 1: Example of HTML code with Adaplix directive

with any browser. However, it's difficult to scale up this approach to a large group of users. If pages get altered on the client side, it's easier to scale up. Also the user profile resides on the user's computer which is beneficial from a privacy point of view. The main problem however is in writing the Adaplix-agent such that it can run on many computer platforms and browsers.

Because of this problem we implemented Adaplix as a program which resides on the server side. Because we wanted Adaplix to be server independent, we wrote it as a set of common gateway interface (CGI)-scripts in Perl [14]. When the user requests a webpage, this request is redirected to the CGI-script. This script requests the page from the server. Based upon the user model and the directions of the author of the webpage the page is altered and the resulting page is returned to the user[1].

In order to be able to adapt a page to the user's interest Adaplix needs information about the user as well as directions about how to change the page. The information Adaplix collects about the user comes from two sources. When a user first logs in into the Adaplix system, the systems asks some questions (minimal an identifier and a password but the website designer can extend this list of requested information). Furthermore each time the user visits a page on the website, Adaplix logs the URL and the time the page was visited.

This information can be used by the author of the webpage to indicate how the webpage can be adapted to the user. This can be done by writing condition HTML statements: if the user satisfies certain conditions use HTML code H_t else use HTML code H_f. Because HTML code can specify lay-out as well as content of a page, this simple schema is also very flexible: the author can use it to change the background, the font size, the text and/or pictures displayed etc. Tests can be based on the information the user provides during the first log in as well as on information extracted from the logs (has the user already visited page X more than N times, has the user visited page Y before date D,...). Figure 1 shows an example of a piece of HTML code with Adaplix conditional tests. It tells Adaplix to use colordots.gif as background file if the user is born after 1990, greyrelief.gif otherwise.

3 Learning Browsing Preferences

The Adaplix system logs all pages visited by a user together with the time of visit. Based on this information we can calculate how long a user visited a page. We use this as an indication of the interest the user has for this page. If we can identify a

[1] all links in the page are replaced by links to the CGI-script

correlation between attributes of a page and the duration of the visit of a certain user to that page, we can use this information to predict how long the user will visit an yet unvisited page (and hence — if our hypothesis that duration of visit is an indication for interest holds — we can predict the interest of a user in an yet unvisited page).

So the task that remains is identifying correlations between webpages and the duration of the visit to each of these pages. This can be seen as an inductive learning task. Each visit to a webpage can be seen as one example, and the task is to induce from a set of examples one general hypothesis predicting the duration of a visit to a page based upon the attributes of that page. Since we're only interested in a rough approximation of the duration of the visit, we can discretise the duration values into three classes. In this way we transform the browsing preferences learning task into the task of inductively building a classifier.

The input of the system consists of classified examples. Each example consists of the following information:

- title: each word occurring in the title of the document
- counts: for each page we count the number of images, level 1, 2, 3 and 4 headers, frames, http-links, ftp-links, email-links and the global number of words occurring on that page
- words: the system keeps a vector of frequently occurring words [2]. Each example contains a list of words from this vector that occur in the document, together with their frequency
- class: the discretised duration of the visit to the page
- previous page: link to the information about the page visited before the current page

A link to the information of the previous page is included because the duration of a visit to a page can be dependent on previous pages visited, e.g. a user who doesn't like pages with lots of links on it, unless it is a page visited right after a page which had the word "homepage" in the title.

We use inductive logic programming (ILP) [10, 6] as a learning methodology because this is a very flexible technique which is able to learn hypothesis which take into account relations between objects (such as the relation between pages visited in one session, words in one document, ...). ILP systems are also able to use 'background information'[3] in the learning process. By this we can for example use a lexicon (such as WordNet [9]) in the learning process.

In the experiments the ILP-system Tilde [3] was used because it is very efficient in constructing classifiers in the ILP setting. Moreover, experiments show that it can handle large datasets [4]. Tilde builds a first order equivalent of a decision tree. Figure 2 shows an example of a decision tree built by Tilde. If the page contains ftp-links the duration of the visit will be medium. Else it depends on

[2] stop words like 'the', 'a', ... are left out

[3] general information which holds for all examples

```
mid(A)
ftprefcnt(A,B) , B < 1 ?
+--yes: wordcnt(A,C) , C < 56 ?
|       +--yes: short_visit
|       +--no:  long_visit
+--no: medium_visit
```

Figure 2: Part of a decision tree produced by Tilde

the number of words in the document: with less than 56 words the prediction is a short visit, else a long visit.

4 Experimental Results

The system described in the previous section was implemented and used to analyze the visits to an experimental site. This site contains 181 pages on nine different subjects. A total of 69 users visited this site, resulting in 1496 visits to pages. The duration was discretised in three intervals (short, medium and long visit). This data was used in the following experiments. All reported testing accuracies are obtained from a six-fold cross-validation.

A first problem to solve is the discretisation of the numeric values (number of images, headers, links, ...). Tilde is able to discretise automatically. However one has to specify the number of intervals by hand. With three intervals Tilde gets a 10% better test accuracy than with two intervals. Further increasing the number of intervals doesn't increase the accuracy (it only increases the time to build the decision tree).

Next we analyze whether the system can predict better than just always predicting the main class. The main class is `short visit`, with 41% of the examples belonging to this class. The decision tree built by Tilde for an individual user[4] however has a testing accuracy of 62%, which is significantly better. We also tried to build a tree without distinguishing between individual users which resulted in a testing accuracy of 58%.

When we analyse the importance of the used attributes it turns out that not using the words occurring in the title of a page results in a tree with the same accuracy as when one doesn't use the words in the vector of frequent words or when one uses both. Also surprising is the fact that allowing the system to use information about the previous visited page doesn't result in a more accurate tree.

More information about these experiments can be found in [7].

5 Conclusion

In this paper we described the Adaplix system. The goal of Adaplix is to provide a system to website developers to enable them to make their site adapt itself to

[4] we used the user that visited the highest number of pages

the visitors. To this extend we first created a system which transforms HTML extended with conditional statements into plain HTML. The conditional statements contain tests based on certain properties of the user. For the system to be able to test these properties, it has to log the user's browsing behaviour. Since we already have this information, our next step is to use it to learn a profile of the user's browsing preferences. This is treated as a inductive classification tree building problem for which the Tilde system is used. In this way we combine two common approaches to adaptive websites: defining the possible changes manually or using machine learning techniques to find useful information for the user. We also investigated whether inductive logic programming can be used to learn user preferences in web browsing. As the preliminary results show Tilde was able to accurately predict the time the user spends at a page (accuracy comparable with WebWatcher).

This approach can be compared with systems like Webwatcher [11, 8], Fab [2, 1] and Do-I-Care [13, 12]. The main difference with the above systems is the combination of machine learning techniques together with the classic approach of adaption predefined by the author of the document. We also use ILP, a very flexible learning technique which allows for the use of many useful types of background knowledge (such as a lexicon, translations and domain specific knowledge).

Future work will be to integrate the learning component in the Adaplix system so that the learned knowledge can be effectively used when building adaptive websites. We use almost no background knowledge in these experiments. Further experiments can indicate the effect of the use of the types of background knowledge mentioned above.

Also on the learning aspect further extensions are possible. We only learned from individual users. Techniques from the domain of collaborative filtering can be integrated in the system to learn from other users as well. Since ILP is a very flexible learning technique this can easily be integrated.

Acknowledgements: The author wishes to thank Lieven Desmet, Jos Devloo, Roel Hertoghs and Guy Wellens for implementing the basic Adaplix system. Tom Ghelen implemented and experimented with the learning component. Hendrik Blockeel helped with the Tilde system. Thanks also go to Luc De Raedt and Denise Pilar da Silva for the interesting and stimulating discussions. Nico Jacobs is financed by a specialization grant of the Flemish Institute for the promotion of scientific and technological research in the industry (IWT).

References

[1] M. Balabanovic and Y. Shoham. Fab: content-based collaborative recommendation. *Communications of the ACM*, 40(3):66–72, March 1997.

[2] Marko Balabanovic. An adaptive web page recommendation service. In W. Lewis Johnson and Barbara Hayes-Roth, editors, *Proceedings of the First International Conference on Autonomous Agents (Agents'97)*, pages 378–385, New York, February 5–8, 1997. ACM Press.

[3] H. Blockeel and L. De Raedt. Top-down induction of first order logical decision trees. *Artificial Intelligence*, 101(1-2):285–297, June 1998.

[4] H. Blockeel, L. De Raedt, N. Jacobs, and B. Demoen. Scaling up inductive logic programming by learning from interpretations. *Data Mining and Knowledge Discovery*, 3(1):59–93, 1999.

[5] Peter Brusilovsky. Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 6(2-3):87–129, 1996.

[6] L. De Raedt. Logical settings for concept learning. *Artificial Intelligence*, 95:187–201, 1997.

[7] T. Ghelen. Adaplix: Leren van gebruikersinteresses uit bezochte webpagina's. Master's thesis, Department of Computer Science, Katholieke Universiteit Leuven, 1999.

[8] Thorsten Joachims, Dayne Freitag, and Tom Mitchell. WebWatcher: A tour guide for the World Wide Web. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI-97)*, pages 770–777, San Francisco, August 23–29 1997. Morgan Kaufmann Publishers.

[9] G. A. Miller, Beckwith R., C. Fellbaum, D. Gross, and K. J. Miller. Introduction to wordnet: An on-line lexical database. *Journal of Lexicography*, 3(4):234–244, 1990.

[10] S. Muggleton and C. D. Page. A learnability model for universal representations. In S. Wrobel, editor, *Proceedings of the 4th International Workshop on Inductive Logic Programming*, pages 139–160, Sankt Augustin, Germany, 1994. GMD.

[11] T. Joachims R. Armstrong, D. Freitag and T. Mitchell. Webwatcher: A learning apprentice for the world wide web. In *AAAI Spring Symposium on Information Gathering*, 1995.

[12] Brian Starr, Mark S. Ackerman, and Michael Pazzani. The do-I-care agent: Effective social discovery and filtering on the web. Technical Report ICS-TR-96-50, University of California, Irvine, Department of Information and Computer Science, August 1996.

[13] Brian Starr, Mark S. Ackerman, and Michael Pazzani. Do-I-Care: Tell me what's changed on the Web. In *Proceedings of the AAAI Spring Symposium on Machine Learning in Information Access Technical Papers*, March 1996.

[14] Larry Wall, Randal L. Schwartz, Tom Christiansen, and Stephen Potter. *Programming Perl*. Nutshell Handbook. O'Reilly & Associates, 2nd edition, 1996.

The role of standards in digital longevity

K. van der Meer(1) and J.J.M. Uijlenbroek(2)
(1) Delft University of Technology
Zuidplantsoen 4
2628 BZ Delft
The Netherlands
(2) Het Expertise Centrum
Jan Willem Frisolaan 3
2517 JS 's-Gravenhage
The Netherlands

Introduction

An old joke says, "No one has contributed so much to history as historians have". The mission of historians is to create history by constructing and analysing of what has happened. Sometimes the reconstructions seem to deviate from the historical facts.
The advance of information and communication technology (ICT) has brought something new. Electronic documents and records will be unreadable within a few decades. A major task of computer scientists will be to preserve history by constructing and analysing an enormous amount of old electronic records and documents. Within a few years, no one will have contributed so much to history as computer scientists have...
ICT has made document handling easier. One can simply move, copy, cut and paste electronic document information, and construct a new document by using available document parts. It is difficult to believe that digital longevity could be a problem. It is an oxymoron that just under these circumstances document information seems to taint and wither under one's hands.
The digital longevity problem was noticed quite a few years ago. The Dutch Bureau of Digital Longevity was founded in 1991, the book 'Preserving the Present' was published in 1993 and Rothenberg's article appeared in the Scientific American in 1995. Recently, many organisations needed to convert electronic documents or had even to face their loss. But in spite of these, still today many people find it difficult to appraise the relevance of the questions. Proposals of answers are even more difficult to appraise.

Key factors

If we want to guarantee that an electronic document will be readable in 50 years, how can we attain that? On what data carrier can it be preserved? If it is migrated to an other carrier and to an other platform, can its provenance be proved? What will its legal status be? How to recognise electronic annotations to it? On a higher level, what key factors are pertinent to the development of a policy and strategies to preserve archival electronic records? Could an archive issue a Service Level Agreement (SLA) to promise to society that certain documents can still be accessed and serviced in 50 years' time?
An analysis of vital questions leads to the following types of problems.

- Can different levels of authenticity be identified, and if so, what will the use be of them? What roles could exist for electronic records with different levels of authenticity? What requirements must be set to provenance?
- What requirements to hardware and software should be set?

- What responsibilities will be put upon electronic archive management? How can appraisal of electronic records be realised in practice? How can quality requirements be developed that are applicable to control and management of electronic archives?
- What types of costs are made? Can benefits be realised? How do the costs and benefits of electronic record management relate to those of paper record management and how to deal with costs of the ICT infrastructure? What models of cost accounting are applicable for electronic archives?
- What standards are applicable to electronic documents and electronic records and what about longevity of standards?

Standards are vital

Document sharing proves to be a new and strong trend. Open organisations exchange lots of documents with their environments (and closed organisations may hardly be viable much longer, one more consequence of ICT). The sheer reason of many documents' existence is their serving as a means of communication between organisational units. Records witness transactions between organisations, stem from CSCW, or originate from outside the organisation that keeps the records. Document sharing depends literally on standardisation. The design of interorganisational document information systems depends on standardisation of electronic document functionality.

The importance of standards for preservation has, however, been a matter of dispute.

Firstly, it has been stated that standards cannot solve the problem of digital longevity. Still, in our opinion, standardisation is one of the key factors.

Secondly, it has been stated that standards are subject to change. There is a point. Technical developments and improvements continue to lead to new possibilities and hence to new standards. So it must be expected that standards will age.

Nevertheless: standards must answer longevity requirements, else they are void. This leads to the question, how standards and their substrates (documents) can be carried through time. For a first answer to that question, let us browse through some pertinent standards and comment on the longevity aspect. We limit ourselves to standards on (provision of) document information.

Families of standards

There are quite a few important groups of standards for electronic records. They are given here.

1. Document structuring.

- Standard Generalised Mark-up Language SGML (ISO 8879) for the logical document structure,
- Document Style Specification and Semantics Language, DSSSL (ISO 10179) for the document lay-out structure,
- HyperText Mark-up Language, HTML (ISO DIS 15445) for the mark-up and hyperlink facilities of Internet documents,
- Cascading Style Sheets, CSS (no ISO standard), viz. the DSSSL for HTML,
- eXtensible Mark-up Language, XML (not yet an ISO standard), a less complicated SGML,
- HyTime for Hypermedia and Time-based documents (ISO 10744),
- XHTML, standards for multimedia documents, XSL, Xpointer etc. for XML, ...

In 1986 ISO accepted both SGML and one more standard for the document structure: Office Document Architecture, ODA (ISO 8613). Its purpose was comparable to SGML and DSSSL

together. Its implementation was expensive and ODA was hardly used at all. This standard came too early.
SGML needs a 100% DTD modelling. In practice, that is a lot of work. That is why SGML was not widely accepted. It is accepted now mainly in the fields of manuals of complex equipment and publishers. But the need for document structuring is persistent and now XML evolves. XML is a much simplified and slightly enriched (hyperlinking) SGML.

2. Printer language standards.
The printer language standard is Portable Document Format, PDF. It is based upon the older PostScript. PDF is not an official standard, but is proprietary to Adobe®.
PDF is an example of a de facto standard. The ISO printer language standard is SPDL, ISO 10180. SPDL does not seem to have many advantages over PDF. SPDL is hardly supported. Everyone seems to have an Acrobat reader for PDF, but more software is necessary to use all features of PDF. The longevity of this standard is possibly proportional to the longevity of the owner organisation.

3. Interoperability of documents.
- ODMA (Open Document Management API) and
- DMA (Document Management Alliance) are standards aiming at interoperability: access to documents in a distributed environment, open to different document management software and different applications. Of ODMA, a one-to-many solution, version 2.0 is getting accepted in the market. Of DMA, the many-to-many option, version 1.0 seems to lag behind. These standards are owned by AIIM.
- WebDAV, a distributed authoring protocol for Internet documents on which much work has been done in 1997-98, is 'owned' by the W3C.

At this moment there is not really a standard for document interoperability that is stable enough and has sufficient mass. The support may be limited if the supplier vanishes.

4. Meta data elements.
A lot of work has been done on definition of meta data since about 1970. Examples of sets are the following.
- MARC set,
- ISAD (G) set
- Pittsburgh set,
- Dublin core.

MARC is well known in the library. It is too limited for record management. UNESCO set up ISAD (G) in 1994; despite the high patronage there is only a very limited installed base of ISAD (G). The Pittsburgh set and Dublin core (for electronic library materials) have not yet crystallised out. This standardisation process is converging only very slowly.

5. Content access for information retrieval.
Content access will probably be defined by ISO (at the moment a Draft International Standard, DIS) 23950. This is a standard that supersedes ANSI Z39.50 and narrower terms (there are related ANSI standards) and some of the ISO standards in the range of ISO 10160 – 10166.

The ISO 10160 up to and including ISO 10166 standards and Z39.50 were widely known and although there were apparently differences in implementation, their content seems to have been respected. The ISO 23950 standard could be a basis for a long-time standard.

6. Information content structures.
The information content structures have been standardised by the thesaurus. There are two of these: the monolingual, ISO 2788, and the multilingual, ISO 5964. ISO 2788 is very widely known and many packages conform to that standard.

7. Character sets.
The list of standardised character sets contains ASCII (ISO 646), ISO 8859 (the Latin series), ISO 6937 (Teletext and Videotex) and finally UNICODE (ISO 10646). ISO 8859 and ISO 6937 include ISO 646; ISO 10646 includes the other standards. Apparently, there is backward compatibility to the older standards (that are in use). That standard could last the whole coming millennium.

8. Others.
A kind of catalogue for document information standards that are not very relevant from the viewpoint of ICT can be found at http://www.iso.ch/liste/TC46SC9.html, http://www.iso.ch/liste/TC46SC4.html and http://www2.echo.lu/oii/en/archives.html.
ICT standards that have less importance for document information are http, CORBA, COM / DCOM, OLE, DLL, ISO 9660 for CD-ROM and ISO 11560 for WORM, the Workflow Management Coalition standards etc. All those standards are out of scope for our purpose.

"The nice thing about standards is that there are so many different ones to choose from". This common quote proves to be wholly unjustified. There are only few standards for preservation of electronic records that we would dare to advise now for digital longevity. A SLA for electronic records can, alas, not be issued at this moment.
For that kind of agreements and for the very goal of preservation of digital documents and records, existing and new standards in this area should explicitly meet approval, acceptance and implementation for digital longevity. They may have to be explicitly anchored to digital longevity purposes. Once standards are suited, there is a form of healthcare needed for standards, e.g. by insisting on backward compatibility of new developments to existing and useful standards, although other tools could be thought of, too.

Some notions concerning the evaluation of standards
In order to evaluate the operation of some families of standards from the digital longevity point of view, the following parts are distinguished concerning digital documents:
- a documents content which is the dominant aspect in standards that enable document structuring;
- a documents description on which meta data standards focus;
- the accessibility of documents, the central issues in standards concerning the interoperability (e.g. remote access) and standardised thesauruses.

It must be determined to what extent a stored document should correspond to its original. In order to ensure to future generations that documents are accessible in its original format (implying complete reuse) one has to store not only a document, but also its creating environment (not only

MS-Word). The cost of this demand is expected to be very high. Perhaps, in most cases sooner or later one has to accept some loss of information. The issue here is not to search for the 'super' standard on digital longevity, but to enable future computer-scientists-and-historians to reconstruct a document like historians today may reconstruct a document and the information therein from the Middle ages.

If one drops the demand regarding a document's original format but insists on reusability, one ends up with a standard from the XML-family. If one drops the reusability a PDF or TIFF-like formats satisfy.
The less complex a standards is, the more information is lost when it is used to store documents (but the lower the cost to convert a digital document from one version to a more recent version may be). Archives relate this to the appraisal question: which documents are so important (from an organisational, cultural or evidential point of view) that they must be preserved, and how will they have to be preserved, for what purpose will they be used.
But the basic trade off is between the current cost of preventing loss of information and the cost of reconstructing (e.g. converting) a document.

Until so far the evaluation focussed on the documents content and possible loss of information. This line of reasoning is also assumed to be applicable to a document description, e.g. the meta data. Although meta data describes the context of a document, that is needed to understand and interpret a document, (partial) loss of meta data does not by definition makes a document obsolete. Compare it to the loss of information of paper documents from medieval times, where lots of meta data were lost and have been reconstructed by historians. Consequently, one has to determine which kinds of documents belong to the 'no loss of information' category and which kinds of documents can be stored with very little meta data (e.g. system generated meta data). Related questions pop up: what are the cost of adding relevant meta data compared to future recovering of lost meta data and which kinds of documents are so important that loss of meta data is cannot be allowed.

Regarding the accessibility of documents a new phenomenon comes up: a certain knowledge area structured in a thesaurus is by definition cultural and time based. Consequently, today's thesaurus may be of importance to future generations concerning the way a certain knowledge area was structured, but will not probably not satisfy their demands concerning accessibility. Future historians play an important role: they will know their field of study in order to gain accessibility to the document of today. Does that mean that preserving thesauruses to access today's documents for future generations is in fact a waste of resources? The same applies to interoperability: is it effective to preserve today's interoperability in order to ensure future interoperability?

Conclusion
Digital longevity of documents and records is a major problem. Standardisation is necessary to ensure that electronic records can be preserved in time. An overview is given of types of standards. Their behaviour in time is commented upon. It is concluded that they are not explicitly meant for longevity purposes. A conclusion is drawn with respect to their use for digital longevity: only few standards can be recommended now. However, the choice for a standard does not solve the problem of digital longevity permanently. It is expected that standards evolve during time and sooner or later one has to convert ones documents to a contemporary standard,

probably leading to some lost of information. Regaining a document in its original state, appeals on the reconstruction capabilities of future historians. Consequently a trade off has to be made between the cost of keeping a document in its original format versus the cost of reconstructing is. The basic question is what to do now, to enable future historians to easily reconstruct our times. Choosing the right standard is only one thing to do. Work is in progress aiming at the common area of longevity and standards.

References

T.K. Bikson and E.J. Frinking: Preserving the Present / Het Heden Onthouden. SDU Publishers, 's-Gravenhage, 1993.

D.C. Blair and M.E. Maron: An evaluation of retrieval effectiveness for a full-text document-retrieval system. Commun. of the ACM 28, (1985), 289-299.

P. le Cerf, L. de Bremme and R. Schockaert: Standards for electronic document management. In: Proceedings of the DLM-forum, Brussels, December 1996. Luxembourg: Office for Official Publications of the EC, 1997. Pp. 217-222.

Consultative Committee for Space Data Systems: Reference Model for an Open Archival Information System. CCSDS 650.0-W-5.0 White book. April 21, 1999.

http://www.aiim.org/dma

M. Hedstrom: Research issues in migration and long-term preservation. Archives and Museum Informatics 11 (3-4), (1997), 287-291.

P. Horsman: Digital Longevity: policies on electronic records in the Netherlands. Archives and Museum Informatics 11, (3-4), (1997), 235-240.

B. Kasdorf: SGML and PDF – Why we need both. J. Electronic Publ. 3(4), (1998).

KPMG IT-Trends institute: De IT-antenne: Volgen van ontwikkelingen in een dynamische markt (The IT antenna: Developments in a dynamic market). KPMG, Washington DC, March 1999 (in Dutch).

Y. Marcoux and M. Svigny: Why SGML? Why now? J. Am. Soc. Info. Sci. 48(7), (1997), 584-592.

K. van der Meer: Documentaire informatiesystemen. 3e gewijzigde druk. NBLC, 's-Gravenhage. 1998. (in Dutch).

L. Moscato: Australian approaches to policy development and resulting research issues. Archives and Museum Informatics 11(3-4), (1997), 241-250.

http://www.activedoc.com/odma

P. Over, W.E. Moen, R. Denenberg and L. Stovel: Z39.50 Implementation experiences. NIST Special Publication 500-229, US Department of Commerce, NIST, Gaithersburg, 1995.

J. Rothenberg: Ensuring the longevity of digital documents. Scientific American 272 (1), (1995), 24-29.

J. Rothenberg: Avoiding technological quicksand: finding a viable technical foundation for digital preservation. A report to the Council of Library & Information Resources (CLIR), January 1999. (http://www.clir.org/pubs/reports/rothenberg/pub77.pdf).

G. O'Shea: Research issues in Australian approaches to policy development. Archives and Museum Informatics 11 (3-4), (1997), 251-257.

K. Tombs: Governmental, industry and user perspectives of achieving standard storage mechanisms for long-term archival activities. In: Proceedings of the DLM-forum, Brussels, December 1996. Luxembourg: Office for Official Publications of the EC, 1997. Pp. 210-216.

E.J. Whitehead: Collaborative authoring on the Web: Introducing WebDAV. Bulletin of the Am. Soc. Info. Sci. Oct/Nov 1998, 25-29.

Het evalueren van gebruik en ontwerp van een web interface voor thesaurus ondersteund zoeken.

R. E. de Vries
NIWI, Amsterdam (www.niwi.knaw.nl, repke.de.vries@niwi.knaw.nl)

Samenvatting

Om gebruikers van NIWI databanken zo efficiënt mogelijk te laten zoeken en een zo goed mogelijk inzicht te bieden in het wetenschappelijk terrein bestreken door een databank, is sinds juni 1999 thesaurus ondersteund zoeken toegevoegd aan een tweetal databanken die met deze thesauri al langer ook ontsloten werden. Er is een uniform web interface ontwikkeld om in de thesaurus te laten zoeken en browsen. In tegenstelling tot zoekinterfaces op het web, blijkt er nog weinig standaardisatie te zijn in ontwerpen voor thesaurus toegang. Ook is er weinig bekend over waardering door gebruikers en hun feitelijk omgaan met thesaurus hulp, waar in een web omgeving gebruikersgroepen anoniem en van zeer verschillende achtergrond zijn. Bovendien hebben web gebruikers eerder ervaring met simpele interfaces en vrije tekst zoeken dan uitgebreidere voorzieningen en het benutten van gecontroleerde termen. Summier zal enige achtergrond geschetst worden, waarna eerste evaluatie resultaten volgen van het gebruik van het door NIWI ingevoerde thesaurus ondersteund zoeken. De oproep wordt gedaan thesaurus web interfaces meer aandacht en uniformering te geven en aanbieders ervan, ruwe log file data ter beschikking te laten stellen voor evaluering door derden vanuit verschillende invalshoeken. De Werkgemeenschap Informatiewetenschap zou in die "log data sharing" een rol kunnen spelen.

Achtergrond

Met de komst van het web heeft het ontsluiten en doorzoeken van online databanken grote veranderingen te zien gegeven:

- het ontstaan van vrije tekst zoeken op ongestructureerde en niet verder ontsloten informatie (zoals met de zogenaamde "search engines")
- het door eindgebruikers zelf direkt toegang krijgen tot databanken in plaats van door intermediairs (waardoor de mate van deskundigheid onvoorspelbaar is)
- het bijgevolg ontwerpen van sterk vereenvoudige zoekinterfaces die voor iedereen bruikbaar zouden moeten zijn (gespecialiseerde zoektechnieken zijn optioneel)
- het buiten kontrole van gebruikers toepassen van computer technieken (zoals toevoegen van synoniemen aan gekozen zoektermen en "relevance ranking" van resultaten) om de kans op bruikbare zoekresultaten te verhogen

Zowel door de aandacht voor ongestructureerde en voor moeilijk of niet systematisch te ontsluiten informatie als door de hoge kosten verbonden aan handmatig ontsluiten, is het op het web toepassen van thesaurus ondersteund zoeken een minder ontgonnen terrein. Toch kan dit juist een onervaren gebruiker voordelen bieden: het geeft inzicht in aard en samenhang van het kennisdomein dat door een databank bestreken wordt en het reikt termen aan waarmee gericht gezocht kan worden.
Ook onderzoek naar het omgaan met web interfaces tot databanken is schaarser dan de klassieke studies naar zoeken in "on-line databases" door eindgebruikers. En in

web georienteerd onderzoek is het aspekt van thesaurus gebruik ondervertegenwoordigd [1].
Nauw aansluitend bij de NIWI situatie is werk van Hertzberg en Rudner. Zij beschrijven eerder uitgevoerd onderzoek naar omgaan met web interfaces tot databanken, waarna hun eigen onderzoek volgt. Dit is ingegeven door de zorg of de met een thesaurus ontsloten ERIC database vervolgens ook met hulp van de online thesaurus doorzocht wordt (1999) [2] . In het literatuur studie gedeelte naar "end-user searching" hanteren zij een zestal kategorieën, waaronder "use of a thesaurus", "time spent searching" en "number of queries".
Publikaties over ontwerp overwegingen voor web interfaces tot thesauri zijn er weinig [3], hoewel te zelfdertijd de opkomst van Digital Libraries wel de bouw en het toepassen van nieuwe online thesauri als zodanig aanjaagt: bijvoorbeeld de ACM DL '98 workshop "Application of terminology and classification tools for digital collection development and network-based search" [4] en het Networked Knowledge Organisation Systems/Services (NKOS) initiatief begonnen in 1997 in ACM Digital Libraries verband [5].

Het NIWI web interface voor thesaurus ondersteund zoeken

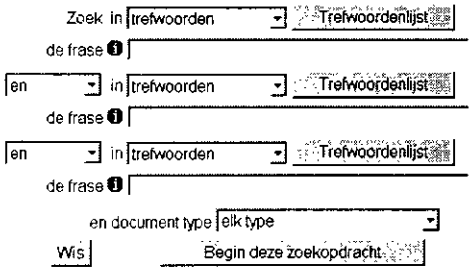
De toepassing is uniform van karakter en geschikt voor thesauri in het algemeen maar is ingezet voor een tweetal NIWI databanken waarbij ook ontsloten is met termen uit respektievelijk een thesaurus in opbouw door NIWI op het terrein van de maatschappijwetenschappen en één op het gebied van milieuvraagstukken. De milieu databank en thesaurus is een produkt van het Ministerie van VROM.

Het ontwerp beoogt:

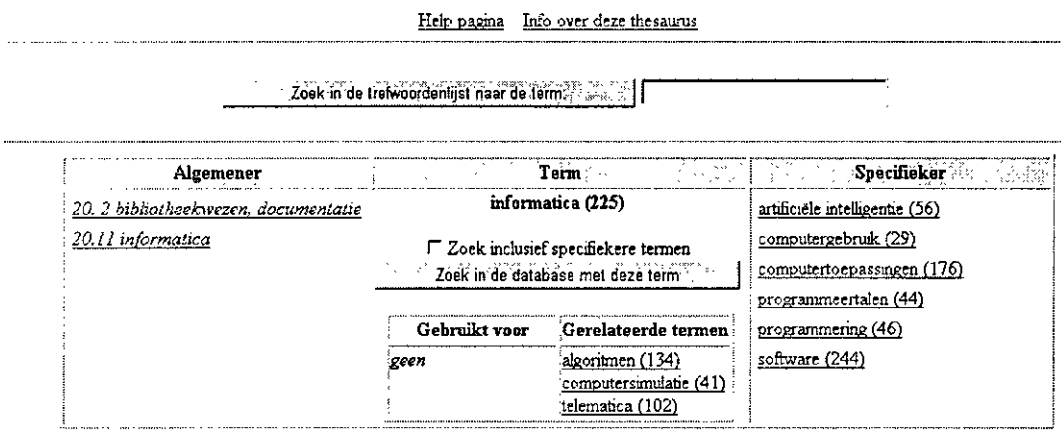
- gebruikers gericht een databank te laten raadplegen door inzicht te bieden in een wetenschapsgebied aan de hand van een thesaurus en door thesaurus termen te laten selekteren voor het zoeken naar informatie
- een gebruiker duidelijk te laten blijken of in de databank zelf gezocht wordt of dat gebrowsed of gezocht wordt in de korresponderende thesaurus ; reden om twee aparte interfaces te maken
- de navigatie tussen zoeken en thesaurus raadplegen zo eenvoudig mogelijk te maken, evenals het overnemen in het zoekscherm van in de thesaurus gevonden termen
- het interface tot de thesaurus zo uniform mogelijk te maken en tot één scherm te beperken ; in dit scherm het volgende samen te brengen: presentatie van de relatie tussen de termen en van het aantal malen dat met een term ontsloten is, de interaktie om te browsen dan wel te zoeken en een help voorziening

Illustratie van zoekscherm en thesaurus interface:

Maatschappijwetenschappen
Bibliografie (SWL)



Bladeren in de thesaurus maatschappijwetenschappen NIWI (swl)



Vergelijking met andere ontwerpen van thesaurus toegang op het web.

Om een vergelijking mogelijk te maken met andere gerealiseerde zoeksystemen met thesaurus ondersteuning, worden in de eindnoot de URL's gegeven van enkele EG gefinancierde projekten (LAURIN, TRANSLIB en LIRN) en van SOSIG, HASSET en tenslote de ERIC databank benadering uit het onderzoek van Hertzberg en Rudner. [6]
Opvallend zijn de grote verschillen in het ontwerp van het thesaurus interface zelf, in het al dan niet scheiden van thesaurus raadpleging en zoeken in de databank en in de keuze die gebruikers gelaten wordt om de thesaurus ondersteuning al dan niet te benutten. Verdere normering, zoals al wel gebeurd is met web zoekinterfaces, zou gebruikers door herkenning en gewenning helpen om met vertrouwen thesaurus ondersteund zoeken, toe te passen.

Het evalueren van gebruik en ontwerp van het web interface voor thesaurus ondersteund zoeken bij NIWI.

In eerste aanzet is gekozen voor evaluatie aan de hand van anoniem gebruik en anonieme gebruikers, door middel van web server log file analyse. Daarbij is gebruik gemaakt van het programma FOLLOW [7] en van standaard UNIX gereedschappen. FOLLOW is ontwikkeld voor usability onderzoek aan web sites en maakt het ondermeer mogelijk achteraf individueel gebruik van een site na te lopen, overzichten te krijgen van de patronen van navigeren en interakties en om per web site onderdeel

inzicht te krijgen wat er aan voorafgaande en er op volgende gebruikers akties. Met het UNIX GREP en WC kunnen uit de log file relevante entries gezocht en geteld worden. Hoewel het met FOLLOW mogelijk is individueel gebruik van de databank en de thesaurus "terug te spelen" is hier om een aantal redenen van af gezien: het programma laat in de presentatie van zogenaamde "cgi-bin" aanroepen niet de variabele gegevens zien, die juist essentiëel zijn bij generieke cgi-bin oplossingen om de specifieke toepassing en betekenis van een aanroep te begrijpen. Een tweede reden is de nog steeds geldende anonimiteit van de web bezoeker: de waarom vraag naar het op een bepaalde manier navigeren en databank raadplegen, kan niet beantwoord worden. Het beperkt in kaart brengen van cgi-bin gebruik door FOLLOW is tevens de reden voor aanvulling van de analyse met tellingen op specifieke log entries, bijvoorbeeld het aantal feitelijke databank raadplegingen waarbij zoektermen gebruikt worden verkregen uit de thesaurus. Waar FOLLOW en de GREP – WC kombinatie ten dele overlappende analyse mogelijkheden hebben, is het goed op die manier een extra kontrole te hebben. Voor het overige vullen de twee methoden elkaar aan: FOLLOW biedt het "wat daarvoor" en "wat daarna", de andere methode opereert op losse log entries zonder context maar kan zeer specifieke gebeurtenissen isoleren.
De mogelijkheden tot konklusies met deze vorm van evaluatie, zijn beperkt: het kan aangeven of de vrijwillige thesaurus ondersteuning herkend wordt in het design van het zoekinterface en of er uiteindelijk gebruik van gemaakt wordt – maar niet of de kwaliteit van de zoekresultaten verbetert of dat de populariteit van de databank er door toeneemt.
Vergeleken met de besproken ERIC databank waar een "wizard" de gebruiker helpt de zoekvraag te formuleren en thesaurus ondersteuning zeer nadrukkelijk aangeboden wordt, gaat het NIWI ontwerp van een voor het web min of meer standaard zoekscherm uit dat zelf ingevuld moet worden en waar zelf de mogelijkheid van thesaurus ondersteund zoeken, herkend moet worden. In deze open situatie mag een tendens dat de thesaurus ingeschakeld wordt en er met gecontroleerde termen gezocht wordt, als eerste bevestiging gelden dat web gebruikers wel degelijk interesse hebben in alternatieven voor het zo in zwang geraakte vrije-tekst zoeken.

In een aantal tabellen zal het ruwe materiaal van de eerste evaluatie gepresenteerd worden en van kommentaar voorzien, gevolgd door konklusies en vaststelling van verder werk.

- Om hoeveel databank gebruik gaat het ?

Het thesaurus ondersteund zoeken voor de databanken MLB en SWL is loop juni 1999 toegevoegd aan de zoek faciliteiten. Behalve één aktie knop voor het inschakelen zijn er verder geen wijzigingen geweest aan voorafgaande navigatie of het zoekscherm zelf. Als indikatie van databank gebruik is gekozen voor het aantal malen dat vanaf elders op de NIWI site of rechtstreeks vanaf buiten (en in beide gevallen na de kontrole op toegang omdat er een abonnement geldt), daadwerkelijk de zoek faciliteit opgeroepen wordt. Het is geen gebruikerstelling: het aantal kan betrekking hebben op enkele zwaargebruikers of een reeks kleine, éénmalige gebruikers en alle varianten daar tussenin. Onderstaand worden het gebruik voor de maanden mei, augustus en een deel van oktober (1 tm 19) weergegeven.

Tabel 1 ; bron: FOLLOW	SWL	MLB
Mei (vóór invoering)	390	206

Augustus (ná invoering)	307	174
Oktober (1-19)	342 (maandschatting: 680)	115 (maandschatting: 330)

Noot : na toevoeging van het thesaurus ondersteund zoeken in juni, is tot in september nog een wijziging in het zoekscherm doorgevoerd geweest maar eveneens in september ongedaan gemaakt: het veld waar standaard op gezocht wordt was bij voorbaat al ingesteld op "trefwoorden" in plaats van "elk willekeurig veld". Dit hinderde de gebruikers gewend aan vrij zoeken. Wel wordt nu automatisch omgezet op trefwoord veld als eerst de thesaurus gebruikt is om zoektermen te bepalen.

- Wordt de thesaurus ondersteuning gebruikt bij het databank zoeken ?

Deze vraag valt in twee deelvragen uiteen:

- het herkennen in het zoekscherm van de mogelijkheid om een thesaurus te benutten (label "trefwoorden" – zie eerdere illustratie) en daar gebruik van willen maken
- het kunnen omgaan met het web interface tot de thesaurus, daar termen kunnen vinden die aansluiten bij het gezochte onderwerp en tenslotte het werkelijk uitvoeren van een zoekopdracht met dergelijke termen (label "zoek in de database met deze term" - zie eerdere illustratie)

De log file analyse geeft hier de volgende aanwijzingen op:

De eerste deelvraag: herkenning en aanroep van de thesaurus ondersteuning.
Zowel het absolute aantal aanroepen vanaf het zoekscherm is in de tabel opgenomen, als als percentage van het eerder gegeven (Tabel 1) aantal malen "oproepen van zoeken in de databank". Bijvoorbeeld voor SWL en augustus: 134 / 307 * 100.

Tabel 2 ; bron: FOLLOW	SWL	MLB
Augustus	Abs. 134 ; 44 %	64 ; 37 %
Oktober (1-19)	123 ; 36 %	38 ; 33 %

De tweede deelvraag: om gaan en om kunnen gaan met het thesaurus web interface. Vergelijkbaar met het gebruik van de data bank zelf, is als indicator van het "omgaan met" gekozen voor het aantal malen aanroepen van de thesaurus faciliteit. Deze biedt zowel browsen als vrij zoeken. De twee vormen van gebruik zijn in de log file niet te onderscheiden. Elk van beide akties, gevolgd door opnieuw presenteren bij welke term men is in de thesaurus hiërarchie, verhoogt het aantal aanroepen. Minimaal komt de telling uit op het aantal aanroepen van de thesaurus ondersteuning als zodanig: bv. (zie Tabel 2) voor augustus en voor SWL: 134. Grotere aantallen betekenen meer gebruik – bijvoorbeeld afdalen in de hiërarchie. In de tabel is tussen haken het totaal aan oproep van de thesaurus faciliteit, ter vergelijking opgenomen.

Tabel 3 ; bron: FOLLOW	SWL	MLB
Augustus	255 (134)	96 (64)
Oktober (1-19)	337 (123)	70 (38)

Kommentaar: het eigenlijke gebruik is kennelijk laag. Dit kan duiden op het eerder zoeken dan browsen in de thesaurus. Het kan vervolgens een aanwijzing zijn dat het

ontwerp niet gemakkelijk genoeg is of dat de gevonden termen niet aansluiten bij wat men voor ogen heeft. Wat betreft het laatste zou het achter de schermen geraadpleegd worden van de thesaurus in een mSQL database, uitgebreid moeten worden met een logging van zoeken en browsen, inclusief de eigen terminologie die gebruikers blijkbaar verwachten aan te treffen en zich kan lenen voor uitbreiding van de thesaurus met synoniemen. Wat betreft het eerste is in ieder geval het teruggrijpen op verdere tekst en uitleg vanuit het interface, minimaal. Er worden er twee aangeboden: "Help pagina" (SWL: elf maal in augustus, vier in oktober ; MLB: elk periode éénmaal) en "Info over deze thesaurus" (SWL: elf en zes ; MLB: twee en niet). Ook de gemiddeld besteedde tijd aan het interface is gering: de "average page time" volgens FOLLOW ligt telkens rond de twintig seconden (met een uitschieter naar 50 sec. voor SWL in augustus): in die gemiddelde tijd wordt de presentatie van de thesaurus verwerkt en geïnterpreteerd en tot een vervolgaktie besloten. Een eerste konklusie lijkt te zijn dat het interface zelf weinig problemen geeft maar dat vermoedelijk niet vlug genoeg aansprekende termen worden gevonden door zoeken of browsen: terminologie of uitgebreidheid van de thesaurus sluiten wellicht niet aan. Men lijkt het dan snel op te geven, gezien het lage aantal aanroepen van de thesaurus faciliteit. Een en ander sluit aan bij het slotonderdeel van de tweede deel vraag:

Vervolg tweede deelvraag: het werkelijk uitvoeren van een zoekopdracht in de databank met termen gevonden in de thesaurus.
Als indikator is gekozen voor het aantal malen dat vanuit het thesaurus interface teruggegaan wordt naar het zoekscherm. Er zijn twee mogelijkheden: de in het ontwerp bedoelde weg terug ("Zoek in de database met deze term") of het teruggaan met de Back-toets van de browser. Theoretisch is het mogelijk dat gebruikers zó Back-gegaan, alsnog op het veld "trefwoord" zoeken en een gevonden thesaurus term zelf invullen. Waarschijnlijker lijkt het dat het aantal Back-met- browser, duidt op een voortijdig en zonder succes willen verlaten van de thesaurus faciliteit. De weg via de ingebouwde link wordt aangeduid met "bedoeld". Zowel het absolute aantal van teruggaan naar het zoekscherm vanuit de thesaurus faciliteit is in de tabel weergegeven, als als percentage van het eerder gegeven (Tabel 2) aantal aanroepen van de thesaurus ondersteuning vanaf het zoekscherm. Bijvoorbeeld voor SWL en augustus: 41 / 134 * 100.

Tabel 4 ; bron: FOLLOW	SWL	MLB
Augustus	Bedoeld: abs. 41 ; 31 % Back v d browser: abs. 58 ; 43 %	Bedoeld: 12 ; 19 % Back v d browser: 46 ; 72 %
Oktober (1 – 19)	Bedoeld: 62 ; 50 % Back v d browser: 54 ; 44 %	Bedoeld: 12 ; 32 % Back v d browser: 23 ; 61 %

Kommentaar: slechts in een beperkt aantal gevallen op het geheel van oproepen van zoeken in de databank SWL of MLB, bleek ook de thesaurus hulp ingeroepen te worden (Tabel 2) ; nu blijkt vervolgens dat op het totaal van inschakelen van die hulp, uiteindelijk maar in beperkte mate ook met gecontroleerde termen gezocht wordt: het percentage "bedoeld". Het percentage dat de Back knop van de browser hanteert, is hoog: dit gegeven lijkt aan te sluiten op de eerdere voorlopige konklusie dat wellicht niet geboden wordt wat de gebruiker verwacht.

- Wat zijn de zoekresultaten ?

Omdat de databanken ook ontsloten zijn met de thesaurus die bij het zoeken ingezet wordt, kan een verbetering van de zoekresultaten verwacht worden. Verbetering zou dan gedefinieerd kunnen worden als het vaker opvragen van volledige bibliografische gegevens vanuit de zoekresultaat lijst met summiere samenvattingen ; en vervolgens als het vaker doen van leenaanvragen of fotokopie bestellingen: een en ander omdat het zoeken gerichter gedaan kan worden vanwege het inzicht dat de thesaurus biedt in het voor de databank onderhavige vakgebied. Om dit effekt te isoleren zouden in FOLLOW individuele web sessies geanalyseerd moeten worden: een andere manier is er niet om te keten te achterhalen van oproepen van databank zoeken, gevolgd door thesaurus hulp, gevolgd door zoeken met gevonden thesaurus termen, gevolgd door opvragen van volledige bibliografische gegevens, gevolgd door leenaanvraag of bestelling. Voor de eerste evaluatie is dit ondervangen en nog beperkt uitgevoerd door met GREP en WC de specifieke aanroepen te bepalen in de log file, die volledige gegevens van boek of artikel opvragen vanuit voorafgaand (succesvol) zoeken op het trefwoorden veld. De aanname is dat de daarbij gebruikte trefwoorden in meerderheid afkomstig zijn uit ingeroepen hebben van de thesaurus hulp. Om enige vergelijking te hebben is dezelfde bepaling gedaan voor de maand mei (vóór invoering).

De tabel geeft het aantal malen volledig opvragen van bibliografische gegevens vanuit zoekresultaten verkregen door zoeken op het trefwoorden veld (tref) – waaronder vanaf juni mede afkomstig van thesaurus hulp en vóór juni uitsluitend met termen afkomstig van gebruikers. Ter vergelijking is tevens het zelfde gegeven opgenomen maar na zoeken op "elk willekeurig veld" (elk).

Tabel 5 ; bron: GREP + WC	SWL	MLB
Mei (vóór invoering)	Tref: abs. 107 ; elk: abs.956	Tref: 95 elk: 542
Oktober (1 – 19)	Tref: 195 ; elk: 1065	Tref: 21 elk: 255

Kommentaar: de GREP aantallen blijken slecht vergelijkbaar met gegevens uit FOLLOW (volledig opvragen zonder nader onderscheid op welk veld gezocht was). Reden om Tabel 5 met grote terughoudendheid te interpreteren. De verwachting dat verhoudingsgewijs het aantal malen volledig opvragen na voorafgaand met trefwoorden (mede afkomstig uit de thesaurus) gezocht te hebben, zou toenemen ten opzichte van het opvragen na vrij zoeken op elk willekeurig veld – is voor SWL met moeite waarneembaar maar werkt voor de MLB tegenovergesteld uit.

Konklusie en verder werk

De percentages aanroep van de thesaurus ondersteuning vanuit het zoekscherm (Tabel 2) zijn een duidelijke aanwijzing dat de faciliteit herkend wordt en betrokken bij het zoeken. De percentages "Thesaurus use" gerapporteerd voor de ERIC database in het onderzoek van Hertzberg en Rudner (Table 2 "Searching Characteristics for Select User Groups", [1]) komen vergelijkbaar uit. Uit Tabel 4 blijkt vervolgens echter dat nog te lage percentages ook werkelijk termen hebben kunnen vinden waar het zoeken in de databank mee uitgevoerd kon worden. Vervolgonderzoek met usability technieken moet uitwijzen of het ontwerp van het thesaurus interface deugdelijk is,

ook al is er weinig dat wijst op problemen in die richting. Uitbreiding van het raadplegen van de thesaurus met logging, zal verder onderzoek mogelijk maken naar de verhouding tussen zoeken en browsen in de thesaurus en aanwijzing moeten geven of gebruikers inhoudelijk aansluiting kunnen vinden bij de nu opgebouwde en aangeboden thesauri.

Tenslotte realiseert NIWI zich dat er weinig algemeen toegankelijke databanken zijn gekoppeld aan online thesauri, om onderzoek te doen naar gebruik en toepasbaarheid van thesaurus web interfaces en thesaurus ondersteund zoeken. NIWI wil daarom de ruwe log file data (na anonimisering van IP informatie) voor dergelijk onderzoek beschikbaar stellen. Een rol voor organisatie en protocolaire begeleiding van dergelijke "log data sharing" lijkt bij uitstek weggelegd voor de Werkgemeenschap Informatiewetenschap.

Noten

1. Hertzberg, Rudner ; "The Quality of Researchers' Searches of the ERIC Database" ; Education Policy Analysis Archives Volume 7 Number 25 August 25, 1999 ; http://epaa.asu.edu/epaa/v7n25.html
2. Zie noot 1.
3. Johnson, Cochrane ; "A hypertextual Interface for a Searcher's Thesaurus" ; ACM DL '95 ; http://www.csdl.tamu.edu/DL95/papers/johncoch/johncoch.html
4. Application of terminology and classification tools for digital collection development and network-based search ; ACM DL '98 ; http://www.alexandria.ucsb.edu/~lhill/dl98_workshop.html
5. Networked Knowledge Organization Systems/Services (NKOS) http://www.alexandria.ucsb.edu/~lhill/nkos/index.html ; met name het funktioneel model dat ontwkkeld wordt
6. LAURIN : http://laurin.uibk.ac.at/ ; LIRN: http://lirn.viscount.org.uk:8887/UI ; TRANSLIB: http://peterpan.uc3m.es/proyectos/translib/HomePage.htm ; SOSIG: http://sosig.esrc.bris.ac.uk/roads/cgi/search.pl ; HASSET: http://155.245.254.46/services/zhasset.html ; ERIC: http://www.ericae.net/scripts/ewiz/amain2.asp
7. FOLLOW, public domain software: http://www.pobox.com/~mnot/follow2

Generating Presentation Constraints from Rhetorical Structure

Lloyd Rutledge, Brian Bailey, Jacco van Ossenbruggen, Lynda Hardman and Joost Geurts*

CWI (Centrum voor Wiskunde en Informatica)
P.O. Box 94079
NL-1090 GB Amsterdam, The Netherlands
Tel: +31 20 592 41 27
E-mail: Firstname{.van}.Lastname@cwi.nl

*Department of Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55455, USA
Tel: +1 612 624-8372
E-mail: bailey@cs.umn.edu

ABSTRACT

Hypermedia structured in terms of the higher-level intent of its author can be adapted to a wider variety of final presentations. Many multimedia systems encode such high-level intent as constraints on either time, spatial layout or navigation. Once specified, these constraints are translated into specific presentations whose timelines, screen displays and navigational structure satisfy these constraints. This ensures that the desired spatial, temporal and navigation properties are maintained no matter how the presentation is adapted to varying circumstances.

Rhetorical structure defines author intent at a still higher level. Authoring at this level requires that rhetorics can be translated to final presentations that properly reflect them. This paper explores how rhetorical structure can be translated into constraints, which are then translated into final presentations. This enables authoring in terms of rhetorics and provides the assurance that the rhetorics will remain properly conveyed in all presentation adaptation.

KEYWORDS: Rhetorics, constraints, presentation generation, meta-structure, authoring

INTRODUCTION

For most of history, the human authoring process was expressed directly in terms of the final presentation. Hand-copied and printed books acted as both data storage and presentation. Hypertext and hypermedia systems are emerging that provide authoring at levels of abstraction higher than final presentation. Authors write specifications at these higher levels, and lower-level presentations are automatically generated that meet these specifications. Examples include the use of CSS and XSL style sheets for presenting HTML and XML documents.

The many types of presentation-independent authoring abstractions include presentation constraints and rhetorical structure. Constraints define the spatial, temporal and navigation limits and relationships that a presentation must have. Rhetorical structure represents the conceptual flow of a document — how the user is lead to understand it. This paper argues that rhetorics are a higher level of abstraction than constraints, and can be used to derive constraints, which are then used to derive the presentation.

Presentation constraints are typically expressed in terms of the timeline, screen layout, or navigation structure. In most constraint systems, only certain aspects of the presentation are adapted to satisfy each constraint. Hypermedia presentation structure can be said to consist of multiple *dimensions*, including primarily *space*, *time* and *navigation*. Current constraint systems typically satisfy constraints of one dimension by generating presentation structure only of that dimension. That is, they will translate spatial constraints into spatial structure, temporal constraints into temporal structure, and navigation constraints into navigation structure. These systems do not provide inter-dimensional constraints — that is, constraints that apply simultaneously to multiple dimensions of presentation structure.

Rhetorics are another means of describing a presentation at a higher level of abstraction [13]. Rhetorics represent the conceptual flow along which a presentation progresses. Rhetorics have been used for analyzing existing documents, and as an instructive aid for writing new documents [14].

This paper describes how a presentation's rhetorical structure can be used to derive the presentation's constraints. Once the constraints are generated, the presentation itself can be automatically derived from these constraints. This means the presentation itself can be encoded in terms of rhetorical structure rather than in terms of less abstract representations such as the presentation's constraints or of the final presentation itself.

This paper first presents background information on automatic presentation generation. Next, it discusses constraints, how they are processed, and what their relationship is with a presentation's structure. Then rhetorical structure is described, as is the means of generating presentation constraints from it. Finally, an implementation of rhetorics and constraints processing for hypermedia presentation generation is presented.

PRESENTATION GENERATION

Abstraction of document presentation, such as that provided by constraints and by rhetorics, provides several benefits. One is that it removes much redundant encoding that the author needs to perform. Another is that it makes the authoring product adaptable to a wider variety of presentation circumstances. Such abstraction also has the effect of changing the authoring process, and the nature of

what is being written. It enables some parts of the human authoring process to become automated. Other parts remain non-computable, but become performed more efficiently by humans in the context of what has become automatically processed. Finally, such abstraction introduces new aspects of the human authoring process that previously did not exist. Much of the code that was previously written by humans is now generated by machine. And what human authors encode themselves is of a different nature than before.

Of course, presentations generated by machines will not have the aesthetics of presentations made by human authors. However, generated presentations are still useful. Some may simply be a quick means of providing selective access to massive amounts of data, such as the pages generated by Web search engines. These need primarily be understandable, and not necessarily pretty. Another use for such a generated presentation is as a first draft, or rough sketch, for a human author. The generated presentation can contain the basic information in a basic presentation structure, and then a human author can edit it into a more pleasing form. Here, the computer makes the job of the human author easier, not unnecessary.

Several prevalent examples of generating useful, but not necessarily pretty, presentations exist currently on the Web. The presentation of HTML is adapted differently by different browsers and different user preferences, and often by different user CSS style sheets. However, the variation in adapted presentation that CSS and browser distinctions provide is small. For example, the specific content, the order of its presentation and its navigational structure are fixed.

More variation is provided with HTML presentations that are generated dynamically. Many commercial sites generate presentations in response to filled-in HTML forms, but these often simply repeat the information provided in the form in a particular structure. News sites often generate presentations from fixed sources based on varying user profiles. These are typically generated from large document information stores and programs that pull components from them for inclusion in specific HTML presentations. But while the content of each generated page varies, it consists of different combinations of fixed components selected for the user and positioned in fixed templates.

The most prevalent examples of generated HTML pages come from Web search engines. Search engines routinely process presentation-independent data into widely varying final presentations. However, although the specific content and the order of presentation can vary greatly with search engines, the structure of presentation is simple and unvarying. Also, the information conveyed is quite simple in structure — typically a list of Web pages with short excerpts.

The current status of presentation generation can be improved upon by increasing the varying complexity of the structure of the presentation and of the information conveyed. The benefits of this include increased adaptation to the user's preferences, language, perceptive abilities and previous knowledge. Constraints and rhetorics are two types of structure that enable wide adaptation. Constraints allow for more complex presentation structure that still adapts to varying presentation circumstances. Rhetorics represent the effect of the information conveyed, and thus understanding their processing provides an important key for increasing its complexity.

A consortium of researchers has developed a model for the automatic generation of hypermedia presentations called the Standard Reference Model for Intelligent Multimedia Presentation Systems (SRM-IMMPSs) [4]. The SRM-IMMPSs describes a general framework for producing final presentations from presentation-independent specifications. It is intended as a model for comparative discussion rather than for implementation, making its application to multimedia presentation generation similar to the Dexter model's application to hypertext [7].

PRESENTATION STRUCTURE AND CONSTRAINTS

This section provides a definition of presentation structure and discusses how constraints on it can be specified and processed. It introduces new types of constraints and constraint resolution that are needed to help the translation of rhetorics to constraints.

Presentation Structure

This paper treats presentation structure as consisting of multiple dimensions, including primarily *space*, *time* and *links*, as shown in Figure 1. Spatial structure defines the layout of visual media items on the screen. Temporal structure is the presentation timeline — the representation of when media items are played. The linking structure is how the user can navigate through the presentation.

Each dimension of hypermedia presentation structure has, in turn, its own sub-structure. Space has two dimensions

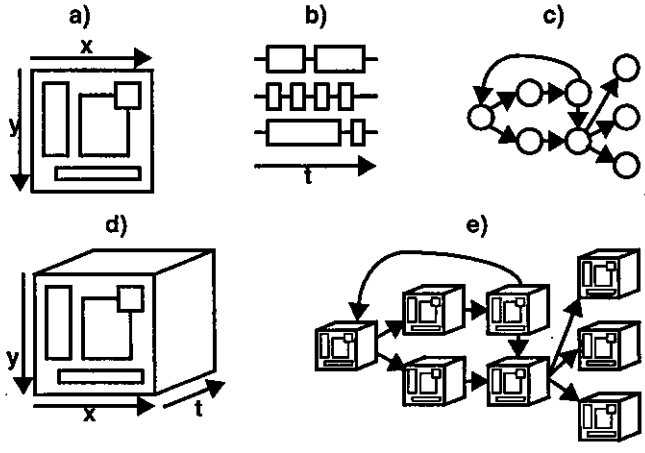


Figure 1: The Hypermedia Presentation Structure of a) Space, b) Time, c) Links, d) Space and Time Combined and e) Space, Time and Links Combined

representing the layout of the screen display. Time has one dimension, along which the playing of the media items is scheduled. The linking structure of a document can be represented as a directed graph, in which each node represents a state of the presentation, and each edge represents a user interaction that moves the presentation from one state to the next. These static states are represented by combinations of space and time structure. They represent how the presentation will progress in the absence of user interaction.

When the structure of the whole presentation is considered, it consists of combinations of these dimensions, also shown in Figure 1. Space and time can be combined into a three-dimensional structure representing how a screen display changes during the progression of a passive presentation. Placing media items along these three dimensions thus represents a state of how a presentation would progress without user interaction. Thus, each such space/time structure can represent a node in the linking directed graph for the presentation. Certain areas in the space-time of each state represent areas of the screen that can be clicked on during certain periods of time to activate the traversal to another state.

Constraint Processing

The details of presentation structure are often rendered from *constraints*. That is, some constraints on what presentation structure is allowed are set, and then a presentation structure that matches these constraints is found. Constraints are just one type of *meta-structure*, or abstraction above final presentation structure, from which hypermedia presentations can be generated. They are, however, perhaps the best-understood hypermedia meta-structures.

PREVISE is a system that resolves spatial constraints through modifications to spatial structure [22]. An example of the use of spatial constraints is shown in Figure 2. An example of spatial constraint code is shown in Figure 3. The Madeus system processes both temporal and spatial constraints, but each type is resolved only with modifications to the presentation structure of the corresponding type [12]. Desired linking structure has been represented with hypertext metrics [5]. The EUCLID project explores the use of spatial constraints in hypertext to maintain desired patterns of user interaction, a conceptual equivalent of linking constraints [20].

In addition to specifying direct relationships between pairs of components, compositional structure has also been used to specify constraints. The W3C multimedia standard SMIL has several types of temporal constraints, including sequential and parallel composites and synchronization arcs [9]. The SMIL editor GRiNS extends SMIL with media selection constraints through the use of channels [6]. Similar temporal composites were also used in the pre-SMIL predecessor to GRiNS, CMIFed [8].

Overflow and Compensation

With most current hypermedia constraint solvers, a constraint on one dimension of presentation structure is typically matched by considering variations of the structure only along that same dimension. This section introduces techniques with the constraints on one dimension can be resolved by affecting any of the dimensions of hypermedia presentation structure. This process starts by considering the presentation structure as a combination, or intersection, of all the dimensions, and then seeking to meet the constraints set by considering variations in the overall structure instead of in just a single dimension.

To illustrate overflow and compensation, a running example was developed that emulates an electronic program guide (EPG) for movies broadcast on television. Specifically, this work enables the user to query a simulated multimedia database containing program guide information, and in response to the query, automatically generate a hypermedia presentation. An EPG screen display is shown in Figure 4.

To define spatial relations between objects, a set of specific-to-general rules are defined. The rules are in the format:

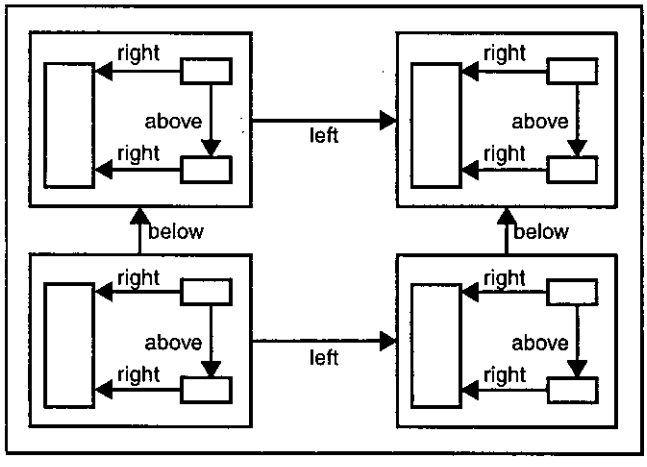


Figure 2: Example of Spatial Constraints

(Ensure that ImageA remains a rectangle)
ImageA.UL.X = ImageA.LL.X
ImageA.UR.X = ImageA.LR.X
ImageA.UL.Y = ImageA.UR.Y
ImageA.LL.Y = ImageA.LR.Y

(Ensure that ImageA maintains its width/height)
ImageA.UR.X - ImageA.UL.X = ImageA.Width
ImageA.LR.Y - ImageA.UR.Y = ImageA.Height

(Ensure that ImageA is left of ImageB)
ImageA.UR.X < ImageB.UL.X
ImageA.UR.Y = ImageB.UR.Y

(ImageA is being constrained to be left of ImageB. Both are costrained to remain rectangles and maintain their dimensions.)

Figure 3: Examples of Spatial Constraint Java Code

spatial_relation(MediaA, ConceptA, Mediab, ConceptB, Rhetoric, position (left, right, above, below)).

For example, a spatial relation stating that the image media object of the movie concept is to be placed underneath the title media object of the movie could be defined as follows:

spatial_relation(image, movie, title, movie, _, below).

However, the same relationship could also be defined on a more generic level by relating the media objects and not the concepts:

spatial_relation(image, _, title, _, _, below).

This states that images are to be placed underneath titles regardless of which concept they belong to.

Figure 5 shows some code from this EPG that specifies overflow and compensation. A sequence of items is to be displayed in the hypermedia presentation. This means that each item is placed within the spatial, temporal and navigational structure of the presentation.

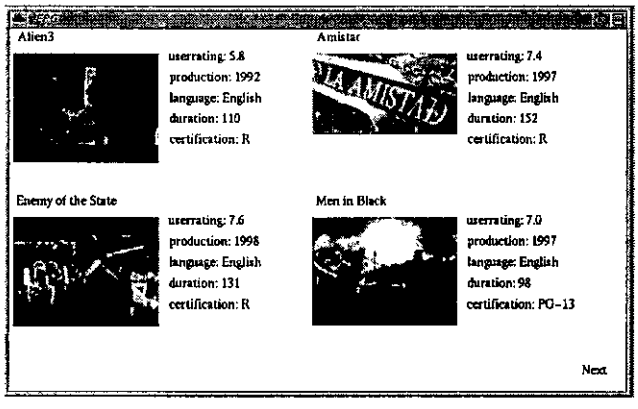


Figure 4: An EPG Screen Display

Sequence(Scene, X1, ..., Xn) :-
XFail=Spatial_Sequence(Scene, X1, ..., Xn, 1)
XFail=Temporal_Sequence(Scene, XFail, ..., Xn)
Link_Sequence(Scene, XFail, ..., Xn).

Spatial_Sequence(Scene, X1, ..., Xn, Column) :-
XFail=Horizontal(Scene, X1, ..., Xn, Column)
Return (Sequence(Scene, XFail + 1, ..., Xn, Column+1))

Temporal_Sequence(Scene1, X1, ..., Xn) :-
NewScene(Scene2)
SequenceScenes(Scene1, Scene2)
Sequence(Scene2, X1, ..., Xn).

Link_Sequence(Scene1, X1, ..., Xn) :-
NewScene(Scene2)
LinkScenes(Scene1, Scene2)
Sequence(Scene2, X1, ..., Xn).

(Xfail represents the element at which the previous goal failed.)

Figure 5: Prolog Code for Generating Spatial, Temporal and Navigational Structure from Sequences

The items are first to be positioned horizontally across the page. Whenever this fails due to the horizontal page boundary being crossed, a new row is allocated and the initial sequence goal is recursively called with the remaining items. Whenever the allocation of a new row fails due to the vertical page boundary being crossed, a new page is created and either temporally sequenced or linked from the previous page. The initial sequence goal is once again recursively called with the remaining items.

Overflow occurs when constraints cannot be meet using only one type of presentation structure, and thus another type of presentation structure must be used as well. In the above example, the items that remain after a single screen display is full are overflow resulting from using only spatial structure to present all the items.

Compensation is using other presentation structure types when the needs of the presentation cannot be met by using only one. Compensation is triggered by overflow. In this example, temporal presentation structure is used if the spatial structure alone is insufficient. That is, if all the items cannot be placed in a single screen display, then they are put in multiple screen displays that are shown one after the other.

A time limit can also be placed on the duration of the full presentation. Such a constraint is useful, for example, in broadcast presentations with which schedules must be adhered to. If the number of screen displays is so large that the resulting presentation would exceed this time limit, then the combination of spatial and temporal structure is not enough to satisfy the constraints put on the presentation. This is another case of overflow, and again compensation is used. This time, the navigational structure is extended as compensation. Links are added to guide the user to particular screen displays instead of having the user passively see all of them. These navigational links would exist as a next, and possibly a previous, button, a menubar, or both.

Intra-dimensional Constraints vs. Inter-dimensional Constraints

Considering compensation results in the distinction between two types of constraints: intra-dimensional constraints and inter-dimensional constraints. *Intra-dimensional constraints* involve only one type of presentation structure. Each intra-dimensional constraint defines a relationship among media objects within a single dimension of space, time, or links without crossing over or including other dimensions. For example, a spatial constraint between two media objects may not also include a temporal constraint within the same specification. However, a single media object itself may separately be involved in both a spatial and temporal constraint. Examples of intra-dimensional constraints include "left of", which involves only spatial structure, and "afterwards", which involves only temporal structure.

The set of intra-dimensional constraints for each type of presentation structure is solved independently of the others. For example, the solution to the set of spatial constraints will have no impact on the solution to the set of temporal constraints, and vice versa. Intra-dimensional constraints do not result directly in compensation. The inability to meet them, however, can cause overflow, which may in turn trigger compensation.

Inter-dimensional constraints define how one type of presentation structure can be altered to allow the meeting of constraints involving another structure type. Inter-dimensional constraints do result directly in compensation. In the current example, an inter-dimensional constraint is used to specify that a group of items is to be played on multiple subsequent screen displays if they cannot all fit in one. The overflow caused by the inability to meet intra-dimensional constraints typically triggers the use of inter-dimensional constraints. Compensation strategies are composed of inter-dimensional constraints and their relationships with intra-dimensional constraints.

Single-pass vs. Order-changing Compensation

The compensation strategy in the current example is simple in that it is linear and one-directional. First only spatial structure is used. When that fails, the temporal structure is used. And when that fails, navigational structure is used. Furthermore, the ordering of the placement of the items in the spatial-temporal structure is fixed. That is, once the first screen display is filled, the remaining items are never candidates for placement in that screen. The list of items is only passed through once and its order is not changed.

One example of a more complex compensation strategy is illustrated in Figure 6. Here, overflow of the spatial layout still results in temporal structure being used. However, the placement of items in the different screen displays is changed to have items of the same media type displayed together. Images are displayed on one screen, and text objects on another. The ordering of the items can be changed to make the presentation look and work better. The input list of items would be passed through multiple times and its order effectively changed for its translation to the final presentation structure. This order-changing compensation strategy is not appropriate when the input order of the item list is significant to the user.

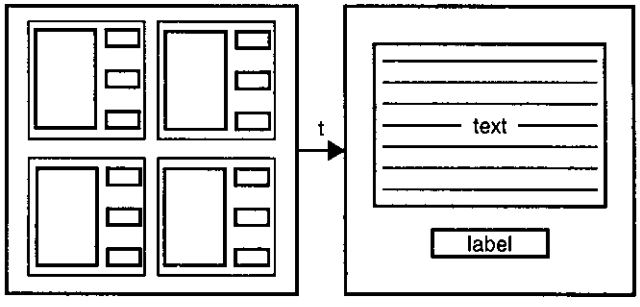


Figure 6: Grouping of Similar Items in Same Temporal

Compensation with Linear vs. Tree-shaped Navigation

Temporal progressions are by nature linear. The navigation structures presented above for handling spatial overflow, that of next and previous buttons and menubars, are also linear. Another alternative when using link structure to handle overflow is to provide tree-shaped instead of linear navigation. Tree-shaped navigation has been long established in hypertext as more efficient than linear. The primary example of tree-shaped navigation is the use of a table of contents, with which sections are selected, and then subsections within them can be selected.

Figure 4 provides a high-level navigational view of the movies selected. The user can click on one movie in this screen display to access another screen display showing more information on that movie. If few enough movies are to be shown, such a detailed screen display may be able to show all movies at once. If overflow happens on this screen display because too many movies are to be shown, a linear navigational compensation could be used to keep the same screen display with next buttons or a menubar. Alternatively, compensation with tree-shaped navigation could be used to generate the display in Figure 4 with links to the more detailed screen displays like that in Figure 7.

RHETORICAL STRUCTURE AND CONSTRAINTS

Presentation constraints were just described as the specification of hypermedia at a level of abstraction higher than that of presentation. There are many other, higher types of hypermedia abstraction as well. They can be used to derive constraints, which in turn can be processed into final presentations. Rhetorical structure is one such higher-level abstraction. This section discusses the generation of constraints from rhetorical structure.

This mapping of rhetorics to constructs will be guided in part by compensation strategies. The compensation strategies introduced earlier in this paper make the mapping of rhetorics to constraints more flexible. They allow a presentation to adapt by altering structure in multiple presentation dimensions and still convey the same rhetorics.

The processing of rhetorical structure into constraints and its context in this paper are shown in Figure 7. This paper

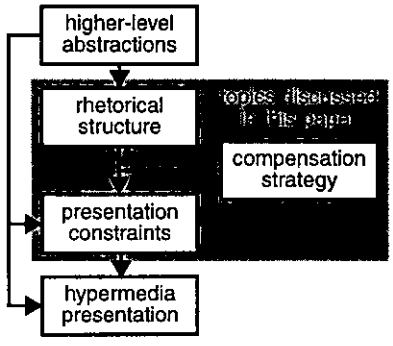


Figure 7: Rhetorical Structure and Constraint Processing

already discussed how presentation constraints are processed into final hypermedia presentations. It also discussed compensation strategies and how they guide this process. This section focuses on how rhetorical structure, as a presentation-independent abstraction above constraints, can be processed into constraints.

Background on Rhetorics and Other Presentation Abstractions

The levels of presentation abstraction that are higher than constraints represent the bases on which constraints are automatically determined. They are equivalent to the considerations human authors may make when defining constraints for the presentations they desire, except that they are modeled in a way that can be automatically processed, either independently of or in conjunction with human authoring. Defining a presentation in terms of higher levels of abstraction enables more varied adaptation. It also gives adaptation a better chance of staying consistent with the author's wishes — that is, it helps the inevitable degradation that comes with adaptation be more graceful.

Rhetorics specify the conceptual structure of how information is conveyed. The study of rhetorics started primarily as a means of analyzing written text. Figure 8 contains a list of rhetoric relations that has been established as complete for most practical purposes of text analysis [13]. It remains to be shown how well these rhetoric structures apply to hypermedia as well as text. This paper does not address this issue, but focuses instead on how rhetorics in general can be used to generate presentations.

Rhetorics, like any presentation-independent abstraction, do not necessarily have a direct correlation with the spatial layouts, timelines, navigational links or media content of the presentations they describe. There can be many different mappings between rhetorics and hypermedia presentation, resulting in many different presentations for the same source rhetorical structure.

The Textnet system provides link types that are often rhetoric structures [19]. This conveys to the browsing user what the rhetoric structure of the information is. In contrast, this paper focuses not on what the structure is but on how varied presentations can be generated from it.

Nucleus-satellite Relations

Evidence	Circumstance	Restatement
Concession	Background	Antithesis
Elaboration	Volitional Cause	Solutionhood
Motivation	Non-volitional Cause	Enablement
Condition	Volitional Result	Purpose
Evaluation	Non-volitional Result	Interpretation
Justify	Otherwise	Summary

Multi-nuclear Relations

Sequence	Constrast	Joint

Figure 8: Some Established Rhetoric Relations [13]

Some research has studied generating multimedia explanations from rhetorical structure [2][14]. This research focuses on explanation as both a particular type of rhetoric and a particular presentation structure. The final generated result of these projects' rhetoric process was a detailed presentation specification, not an adaptive abstraction such as constraints.

The EUCLID project explores the generation of spatial constraints from the structure of reasoned discourse [20]. Reasoned discourse could be considered a type of rhetorical structure. Given this, EUCLID describes constraints from rhetorics with a focus on spatial constraints and on the rhetorics of reasoned discourse.

One type of presentation-independent representation used as input for rendering hypermedia presentations is relational grammars. Relational grammars define domain-specific conceptual relationships between document objects. They have been used as input for deriving spatial constraints of final presentations [21].

The Fiets hypermedia application also explores the mapping from abstract hypermedia representation to presentation structure [17]. Fiets' representation of the city of Amsterdam and its history is represented in terms of time, as years, space, as location, and with various types of relationships. This sense of Amsterdam's time, space and links is contrasted with the time, space and links in the structure of presentations about Amsterdam. For example, the temporal structure of Amsterdam, its history, will not necessarily map directly to the temporal structure of a presentation, its timeline. Fiets demonstrates how the structure of a document can be independent of the structure of its presentation. This property allows multiple presentations that vary widely to be generated from the same document. A screen display of Fiets is shown in Figure 9.

Sequence

Sequence is a rhetorical structure consisting of an ordered list [13]. It is arguably the most important and most frequently used type of rhetorical structure. Most documents consist of sequences of described concepts, each of which is decomposed into smaller sequences or other rhetorical structures.

The sequence rhetorical structure is processed in the EPG application. Its use is illustrated in Figure 5. The abstract concept of a sequence relation does not explicitly specify how it must be realized in terms of presentation format. The goals encoded in Figure 5 define how the rhetorical construct of a sequence of items can be presented in terms of spatial, temporal or navigational structure. One impact that the specification of a rhetoric sequence would have on the resulting constraints is that single-pass compensation would be used, not order-changing compensation. This is because the order of items in the sequence must be maintained in the final presentation.

The sequence rhetorical structure is used in the Fiets application. The higher-level abstractions such as year, address and relation translate directly into sequences. A rhetorical sequence can state that a collection of buildings is to be presented in order of time or, alternatively, in order of street address. Once such a sequence is established, it can be mapped to the presentation structure with a focus on either the spatial, temporal or navigational structure of the final presentation. The Fiets presentation in order of address shown with spatial structure is shown in Figure 9. This would result in, in order, the display of multiple buildings in left-right and top-down order on one screen, a temporal sequence of screen displays that each show one building, or the display of one building at a time with a navigational menu for selecting another building to display.

Strict and Loose Sequences

One variation of rhetorical structure processed by Fiets is that a sequence can be either strict or loose. A *strict* sequence indicated that the items must be presented to the user in the order given, with none skipped. A *loose* sequence, on the other hand, indicates that it is only necessary to indicate to the user that the order of the sequence is important. With a loose sequence, the user has the option to access the items in any order, but what the ordering is will be conveyed.

For example, one Fiets compensation strategy starts with spatial structure and then overflows to navigational structure, showing one page of items at a time, and providing links to other pages. A strict sequence rhetorical structure would be translated into pages that have only a "next" button for accessing other pages. With this, the user can only access the pages in one order. A loose sequence would be translated into pages with a menu bar along the bottom. The menubar has buttons that each indicate and provide access to another page. These buttons are displayed in order, conveying the order of pages to the user, but the user can select any button.

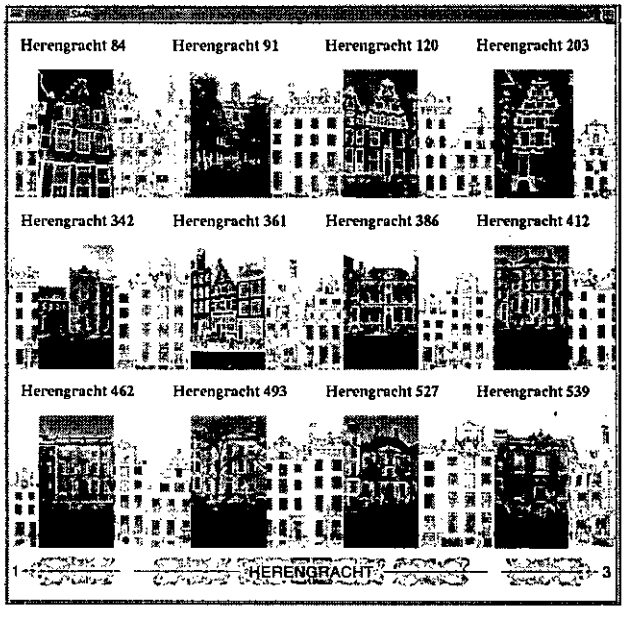


Figure 9: A Fiets Screen Display

Nucleus-satellite Relations

Multi-nuclear rhetorical relations, such as sequence, are lists of equally significant components. Thus, they translate well to list-like presentation structures, as described above. A *nucleus-satellite* relation, on the other hand, associates a piece of knowledge — the *nucleus* — with a related, though less important, piece of knowledge — the *satellite*. These relations are conveyed best with non-list presentation structures, including adjacency spatial relations and directed binary navigational links. A list of multi-nuclear and nucleus-satellite relations is shown in Figure 8.

Adjacency spatial relations convey nucleus-satellite relations well by keeping the satellite concepts visually associated with its nucleus. For example, a piece of text that has a summary relation with an object may best be presented below that object in the final presentation. Translating summary relations as "below" spatial relations will help ensure that this association is visually conveyed to the user in a consistent manner.

Binary navigational links that start with a nucleus object can provide the user with quick access to one or more of its satellites. The information in such a satellite would be considered not essential enough to be shown with the nucleus object in the initial screen display, but significant enough to be readily accessible. This decision may be made in the context of a spatial layout overflow. When all the information cannot be displayed on one screen display, a compensation strategy using tree-shaped navigation would be used, providing links to the less important information. The satellites of certain relations would be assigned in the rhetoric-to-constraint mapping as conveying such less important information.

In the EPG application, each movie may be associated with text making up its summary description by using a summary rhetoric relation. Clicking on a movie image on an EPG screen display like that shown on Figure 4 would bring up a display showing that movie's summary.

IMPLEMENTATION IN BERLAGE

The Berlage environment is a system for authoring and generating hypermedia presentations. It is built in terms of the SRM-IMMPSs [18]. Berlage was developed to demonstrate the generation of presentations from hypermedia meta-structure. With Mix'n'Match, the encoding of how presentations are generated by Berlage was broken up into exchangeable modules [16]. More information about these earlier implementations of the SRM-IMMPSs in Berlage can be obtained from the earlier publications about them. This section describes the components of Berlage and of the SRM-IMMPSs that relate to the translation of rhetorics into constraints.

The primary SRM-IMMPSs components related to constraints and rhetorics are the *design expert*, the *design layer* and the *realization layer*. The design expert stores and provides access to specifications for how various aspects of design, including how rhetorics are translated into constraints, are mapped. The design layer processes design specifications, including those for rhetorics, into constraints. The realization layer processes these constraints into specific mono-medium formats and multimedia formats.

Application Expert

The application expert in the SRM-IMMPSs supplies data relevant to a particular media format or a particular conceptual domain. The EPG relies on a class-instance hierarchy to define a small ontology relevant to the EPG domain, as shown in Figure 10. This ontology is extensible, making the addition of new concepts, attributes, and relations trivial, and the mechanism for searching and retrieving their values already defined. For each class in the ontology, a set of class attributes (in the form of name-value pairs) can be attached. Example class attributes are *width*, *height*, *border*, and *rhetorical relation* to another type. By default, each type inherits the attributes defined for its parent, but the value of an inherited attribute can be overridden by re-defining the same attribute for itself.

Each relational attribute of the movie database is an instance of a class listed in Figure 10. Each instance may also have a set of instance attributes attached that override the class attributes. When an instance attribute value is requested, the value of the first matching attribute starting from the current instance and moving up its class hierarchy is retrieved.

Control Layer

In the SRM-IMMPSs, a goal for the presentation is presented to the control layer. The control layer then processes these goals into sub-goals and communicates them to the content layer. In the Fiets application, there is no focussed goal entered into the control layer. The stored document and the initial settings of the experts, which define the mapping from storage to presentation, are used to provide a default presentation to the user. The EPG application, on the other hand, has as its input a goal. This goal consists of a query entered by the user specifying what programs should be listed in the presentation.

Content Layer

The content layer is responsible for selecting the appropriate media content that meets the information needs of the specified presentation goal. The selection process is non-trivial and must rely on extensive domain knowledge in order for the correct content to be selected. This domain knowledge is provided in part by the application expert. When the content is originally stored in the application expert, semantic information must be attached describing the content items and the relationships between them. Ultimately, the goal is to define semantic information germane to the current domain, but general enough so that the knowledge base does not become a collection of pre-defined presentations.

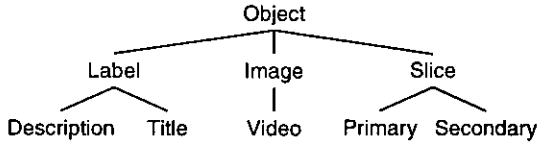


Figure 10: An Ontology for the EPG Domain

In the present system, the content layer relies on the simple concept of a slice, which is adopted from the RMM methodology [11], for semantic information. A slice represents a subset of the attributes of a single relation. Two slices are defined for each table: the primary and secondary slice. When the content layer receives the result of the goal query from the control layer, the set of slices represented by the attributes is retrieved. The query result is then augmented with the additional attribute members of each slice. This result is then passed on to the design layer.

Design Expert

The design expert in the SRM-IMMPSs stores and provides access to different design specifications. Here, a design specification corresponds with the term "stylesheet" used in XML processing. A design could be made by a human designer who is not the document author. Ideally, one design can apply to multiple types of documents, and each document can be rendered with multiple designs.

For example, a multimedia presentation on a particular topic could be generated with a "BBC look" design. Alternatively, the same topic and information could be presented with an "MTV look." Both would be presentations of the same document, with the same topic and information, and perhaps the same media content. The "BBC look" and "MTV look" would correspond with different design specifications provided by the design expert.

In Berlage, specifying the translation of rhetorics to constraints is an aspect of design. This mapping is stored in and provided by the design expert. A rhetoric design expert has been added to Berlage as a component of the design expert. This allows rhetoric processing to be specified separately from other aspects of design, such as font type and visual object color. It is left for future work to explore the ramifications of this distinction.

Design Layer

The design layer in the SRM-IMMPSs is responsible for media generation and overall layout of the media content. It makes the conceptual content fit a given design for how its presentation should look and feel. A general description of how the presentation should progress, specified at least in part by a rhetorical structure, is provided by the content layer. The design is provided by the design expert.

One aspect of design is how rhetorics are translated to constraints. This translation process, described earlier in this

paper, occurs in the design layer. The design layer's output is a set of constraints passed to the realization layer for processing into the final presentation.

The constraints passed from the design to the realization layer currently include both intra-dimensional and inter-dimensional constraints. However, it has not been fully determined whether inter-dimensional constraints are final presentation specification and not design or higher-level presentation abstraction. Future versions of Berlage may thus process inter-dimensional constraints in the design layer and pass only intra-dimensional constraints to the realization layer.

Realization Layer

The realization layer in the SRM-IMMPSs is responsible for creating concrete media objects as well as resolving their spatial, temporal, and linking constraints. The realization layer takes format-independent presentation details from the design layer and generates the presentation in a particular format. The design layer communicates the media objects along with their constraints to the realization layer through a design plan. The goal of the design plan is to be comprehensive enough to capture all of the necessary constraints within the EPG domain, yet be flexible enough to accommodate future requirements. The design plan is represented as an XML document.

To realize the design plan, the realization layer first parses the XML document instance and creates an in-memory representation of the document tree. The layer then separates the different types of relationships such as spatial, temporal, and link relationships. Once separated, both the temporal and spatial constraints are transformed into numerical constraints and subsequently solved using the constraint solver. Links are further processed for quick retrieval during output generation. The solutions from the solver are then passed to the SMIL generator, which maps the elements, attributes, and relationships from the design plan to corresponding SMIL constructs. This entire process is depicted in Figure 11.

IBM's Java *alphaWorks* XML package [10] is used to parse the XML document and manipulate the document tree through its DOM implementation. The *Cassowary* hierarchical constraint solver was developed as an improvement in performance to the DeltaBlue system [3]. The set of specified constraints are then solved using a modified Simplex algorithm. The realization layer transforms each spatial and temporal constraint into one or more linear equalities and/or inequalities and then passes them to Cassowary. The SMIL presentations generated were played using the *GRiNS* SMIL player [15].

SUMMARY

This paper introduces inter-dimensional constraint compensation, which extends this processing with constraints that apply to spatial, temporal and navigational layout simultaneously. It also discusses mappings for translating rhetorical structure into presentation constraints that help ensure that the author's wishes in terms of rhetorics will be meet. The Fiets and EPG applications demonstrated the effectiveness of compensation and rhetoric-to-constraint processing in a small scale for focussed domains. They also provided the first steps for increasing the complexity of compensation and rhetoric-to-constraint translation for hypermedia. Extensions were made to the SRM-IMMPSs-based Berlage hypermedia presentation generation environment to exercise the ideas discussed in this paper, both in terms of process modeling and final implementability.

ACKNOWLEDGEMENTS

Brian Bailey developed the EPG application during a three-month visit to the CWI. The research behind this paper was funded in part by the Multimedia Information Analysis (MIA) project. Many images in Fiets come from the Amsterdam Heritage Website [1].

REFERENCES

1. City of Amsterdam Municipal Department for Preservation and Restoration of Historic Buildings and Sites. Amsterdam Heritage, http://www.amsterdam.nl/bmz/adam/adam_e.html.

2. André, E and Rist, T. "The Design of Illustrated Documents as a Planning Task", in *Intelligent Multimedia Interfaces* (ed. Maybury, M.T.), AAAI Press / The MIT Press, 1993, pp. 94-116.

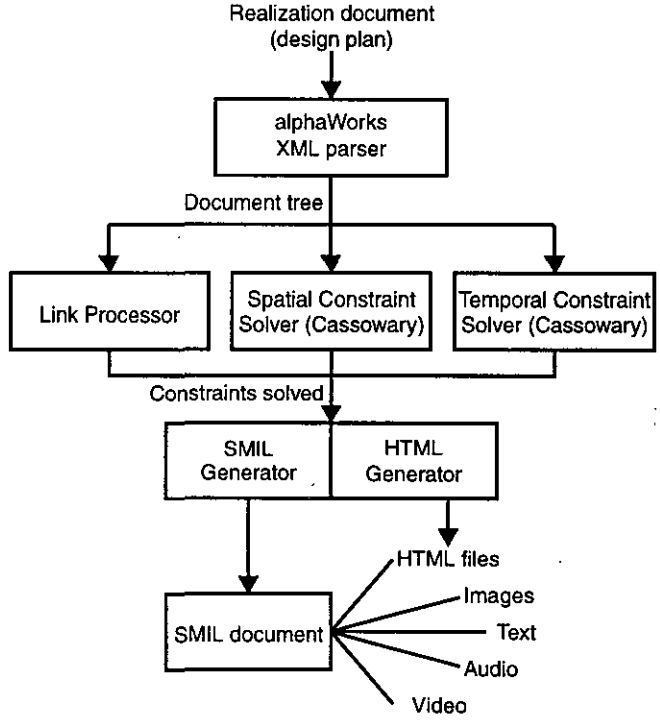


Figure 11: Processing Model of the Realization Layer

3. Badros, G.J. and Borning, A. *Cassowary: A Constraint Solving Toolkit*, http://www.cs.washington.edu/research/constraints/cassowary/.

4. Bordegoni, M., Faconti, G., Feiner S., Maybury, M.T., Rist, T., Ruggieri, S., Trahanias, P. and Wilson, M. A. "Standard Reference Model for Intelligent Multimedia Presentation Systems." *Computer Standards and Interfaces*, 18(6,7) (December 1997), pp. 477-496.

5. Botafogo, R.A., Rivlin, E. and Shneiderman, B, "Structural Analysis of Hypertexts: Identifying Hierarchies and Useful Metrics", *ACM Transactions on Information Systems*, Vol. 10, No. 2, April 1992, pp. 142-180.

6. Bulterman, D.C.A, Hardman, L., Jansen, J. Mullender, K.S. and Rutledge, L. "GRiNS: A GRaphical INterface for Creating and Playing SMIL Documents", *Computer Networks and ISDN systems*, 10, (1998), pp. 519-529.

7. Halasz, F., and Schwartz, M. "The Dexter Hypertext Reference Model". *Communications of the ACM*. Vol. 37, No. 2, February 1994, pp. 30-39.

8. Hardman, L., van Rossum, G. and Bulterman, G.C.A. "Structured Multimedia Authoring", *Proceedings of ACM Multimedia 93*, Anaheim, California, USA, August 1993, pp. 283-289.

9. Hoschka, P. (ed.). *Synchronized Multimedia Integration Language*, World Wide Web Consortium Recommendation. June 1998.

10. International Business Machines Corporation, *IBM alphaWorks*, http://www.alphaworks.ibm.com.

11. Isakowitz, T., Stohr, E.A., and Balasubramanian, P. "RMM: A Methodology for Structured Hypermedia Design". *Communications of the ACM*, 38(8), August 1995, pp. 34-44.

12. Jourdan, M., Layaïda, N., Roisin, C., Sabry-Ismail, L. and Tardif, L., "Madeus, an Authoring Environment for Interactive Multimedia Documents", *Proceedings of ACM Multimedia 98*, Bristol, England, September, 1998, pp. 267-272.

13. Mann, W.C., Mattheissen, C.M.I.M and Thompson, S.A. "Rhetorical Structure Theory and Text Analysis", *Information Sciences Institute Research Report*, ISI/RR-89-242, November 1989.

14. Maybury, M.T. "Planning Multimedia Explanations Using Communicative Acts", in *Intelligent Multimedia Interfaces* (ed. Maybury, M.T.), AAAI Press / The MIT Press, 1993, pp. 60-74.

15. Oratrix Development BV., *GRiNS*, http://www.oratrix.com/GRiNS/.

16. Rutledge, L. Hardman, L., van Ossenbruggen, J. and Bulterman, D.C.A. "Mix'n'Match: Exchangeable Modules of Hypermedia Style", *Proceedings of ACM Hypertext 99*, Darmstadt, Germany, February 1999, pp. 179-188.

17. Rutledge, L. Hardman, L., van Ossenbruggen, J. and Bulterman, D.C.A. "Structural Distinctions Between Hypermedia Storage and Presentation", *Proceedings of ACM Multimedia 98*, Bristol, England, September, 1998, pp. 145-150.

18. Rutledge, L., van Ossenbruggen, J., Hardman, L. and Bulterman, D.C.A. "Practical Application of Existing Hypermedia Standards and Tools", *Proceedings of Digital Libraries 98*, Pittsburgh, USA, June 1998, pp. 191-199.

19. Trigg, R. A *Network-Based Approach to Text Handling for the Online Scientific Community*, Ph.D. Thesis, University of Maryland, November 1983, University of Maryland Technical Report, TR-1346.

20. Smolensky, P., Bell, B., Fox, B., King, R. and Lewis, C. "Constraint-Based Hypertext for Argumentation", *Proceedings of Hypertext 87*, Chapel Hill, North Carolina, USA, November 1987, pp. 215-245.

21. Weitzman, L., and Wittenberg, K. "Automatic Presentation of Multimedia Documents Using Relational Grammars", *Proceedings of ACM Multimedia 94*, San Francisco, USA, September 1994, pp. 443--451.

22. Zhou, X. Michelle. "Visual Planning: A Practical Approach to Automated Presentation Design" *Proceedings of the International Joint Conference on Artificial Intelligence 99 (IJCAI 99)*, pp. 634-641.

Top N MM query optimization

The best of both IR and DB worlds

H.E. Blok, A.P. de Vries, H.M. Blanken
{h.e.blok,arjen,blanken}@cs.utwente.nl
Computer Science Faculty, University of Twente
PO BOX 217, 7500 AE, Enschede
The Netherlands

Abstract

It is commonly recognized that top N queries belong to one of the most important query classes in IR and MM retrieval, or more general, content based retrieval. A lot of work has been done on query optimization in database research but that research has mainly focussed on the area of optimization of databases in a business application environment. In IR research also work has been done on query optimization but this is not directly applicable in a database environment.

Exploiting the integrated content based retrieval technology in the miRRor database system, we intend to look into optimization of top N queries in MM DBMSs. This paper provides a brief overview of current IR and database technology relevant to top N MM query optimization. It also describes our DB environment and ideas on three major sub topics in the light of top N MM query optimization: incorporate known fragmentation techniques to ensure scalability, introduce a new intermediate optimizer layer that supports inter-object optimization, develop a cost based optimizer for real MM querying. This paper will mainly focus on the physical design part since that part has the strongest ties with the IR field.

Keywords: top N, query optimization, content based retrieval, multi media, databases

1 Introduction

Most current MM[1] database systems are not really what the name suggests them to be. First of all they often only operate on one single media type, in most cases 2D images, or sometimes audio. Secondly, these systems are not really database systems but merely large monolithical systems with a lot of data. The consequence of this is that these systems still do not really address *Multi* Media retrieval and lack the properties and accompanying advantages of a *real* database system, such as scalability, extensibility, data independency, etcetera.

In our group research is being done on building a *real* MM DBMS [HVBB98, dV98, dVW99, Doo99]. This research views MM retrieval as a general case of content based retrieval. Commonly known statistical IR techniques have been incorporated in a new type of DBMS that promises to overcome the historical efficiency bounds that usually render IR in a DB environment too slow. This new prototype DBMS is build around an extensible, structured object algebra, called Moa [BWK98], which functions as a layer on top of the Monet [BK95] main memory DBMS developed at CWI.

Utilizing the basics from well known text retrieval systems, several generalizations were build: a small one to experiment with audio retrieval and one to retrieve images using thesauri [Doo99],

[1] MM = multi media.

which both showed promising results, demonstrating that with this database system content based MM retrieval should be very well possible. Since our system is able to handle the meta information about all sorts of distinct media types, such as text, audio and images, within one query algebra, e.g. Moa, it is expected to be much more easy to query those types together than in the current MM database systems which either operate on single media or behave as mediator systems like Garlic does [HKWY97].

However, the database properties, like scalability and extensibility, need to be proven to be available, still. Optimization plays an important role in supporting these properties, since it can compensate the extra executions costs involved in scalability and extensibility. Optimization also becomes important when dealing with *real* MM querying, since it then serves the scheduling of how, when and where to evaluate which media type to get the desired results as quickly as possible.

Due to the limited perception capabilities of the human end user, content based retrieval results are often ranked and only the best answers are returned. The queries that provide such kind of answers are often called top N queries. It also might be very clear that these queries play a very important role in the content based retrieval area and that optimizing this type of queries might be very rewarding.

In this paper we describe our ideas on the optimization of top N queries in a MM DBMS setting.

The rest of this paper is structured as follows. First we introduce the top N query optimization problem a little more in section 2. In section 3 we give an overview of important techniques from the IR and database field relevant to the top N query optimization problem. In section 4 we describe our ideas on what we as three major topics that one has to deal with when one wants to build top N optimization facilities into an extensible MM DBMS. Finally we end this paper with some concluding remarks in section 5.

2 Problem

A typical content based retrieval query is usually evaluated by computing some ranking based on statistics and distances in feature spaces. The returned objects are then sorted by descending relevance relative to the given query. Since users are limited in their capabilities of reviewing all objects in that ranked list only a reasonable top of say N objects is returned.

However, this can turn out to be a quite time consuming process, in particular when done the naive way. The first reason is that the number of objects (i.e. documents) in the DBMS is usually very large (10^6 or even more). From the IR field it is known that usually half of all objects (e.g. documents) in the considered collection contains at least one query term, meaning that even if a system only considers these objects it still has to process a lot of them. The second reason is that ranking certain media types can be very expensive per media object. So the number of objects and the computational effort needed just to evaluate one single object both quickly result in excessive computational costs, unless we start making use of the special characteristics of the data to speed up the querying process.

The problem of top N MM query optimization is to find such techniques that utilize this knowledge, in an effort to:

1. Limit the total set of objects of all types taken into consideration during the ranking process as much and as soon as possible,
2. Limit the set of objects with expensive ranking functions, before ranking them, by using ranking results from objects with less expensive ranking functions, in case more than one media type is involved in the query.

Several techniques are known in literature from both the IR field and the database field to achieve such effects more or less thoroughly. In this document we provide an overview of those techniques

and try to sketch directions in which those concepts might become interesting to be used in the context of MM database research in our group as described previously.

3 State of the Art

3.1 IR Techniques

IR is a quite old research area (compared to MM retrieval research) resulting in a lot of different techniques. A large group of these techniques makes use of certain statistical methods. We mainly concentrate on (optimization) techniques concerning this group of statistics based methods because:

- this group of techniques has been proven to work quite well, so results from this research are more likely to be widely applicable,
- (also because of the previous reason) the techniques used in our group belong to this group as stated in the introduction, thus already providing a suitable platform to build on, making it much more easy to arrange for experimental verification of this research.

This section is structured as follows. First we give an overview of basic technology in the field. Next we provide a brief overview of common optimization techniques that have been developed over the past years in this research area. In the third subsection we briefly touch on some of the most important advantages and disadvantages of the common approach in this field.

3.1.1 Query Processing in IR

This subsection contains two paragraphs, reflecting the two major phases in the life of an IR system. The first paragraph addresses the first phase: indexing. The second paragraph addresses the retrieval phase.

Indexing Phase The basic components of a statistics based information retrieval system, constructed during some kind of indexing process, are[2]:

- a (usually large) set of text documents (denoted by $\{d_j\}$),
- a set of all terms occurring in the documents set (denoted by $\{t_i\}$),
- a set containing the frequency of every term per document, i.e. the number of times each term occurs in each document (denoted $\{tf_{ij}\}$),
- a set containing the in document frequency of every term, i.e. the number of documents a term occurs in (denoted by $\{df_i\}$).

Usually the $\{tf_{ij}\}$ and $\{df_i\}$ are both normalized to eliminate the effects of document respectively collection size. The resulting sets are respectively denoted as $\{ntf_{ij}\}$ and $\{ndf_i\}$.

The parts we describe here are the most important ones. However, a lot of additional techniques are known and used to increase computational efficiency and speed and minimize requirements on system resources like disk, memory and/or CPU. Also some analysis algorithms may be run over the terms like stemming (reducing several terms to one common base form: for instance convert different forms of the same verb to its base).

[2]For a more elaborate overview [Bro95]

Retrieval Phase Given a set of query terms (denoted by $\{q_k\}$) it takes usually the following steps to compute a ranking of the document set $\{d_j\}$:

1. Compute the subset of the terms known by the system and occurring in the query, let us denote this set by $QT := \{t_i\} \cap \{q_k\}$.
2. Compute the subset of the documents known by the system which contain at least one of the terms in QT.
3. Compute the corresponding subsets of $\{ntf_{ij}\}$ and $\{ndf_i\}$, here denoted by $\{ntf_{ij}\}'$ and $\{ndf_i\}'$ respectively.
4. Compute the belief[3] contributions per document per term

$$B := \{bel_{ij} | bel_{ij} = \frac{ntf_{ij}}{ndf_i}, ntf_{ij} \in \{ntf_{ij}\}', ndf_i \in \{ndf_i\}'\}$$

5. Compute the final belief per document by aggregating the belief contributions per document per term.

The quotient $\frac{ntf_{ij}}{ndf_i}$ is usually formulated as the product $ntf_{ij} \cdot nidf_i$, where $nidf_i = \frac{1}{ndf_i}$.

Since a user is limited in his/her perception only a limited set of most relevant (i.e. highest ranking) documents is to be presented to the user. Therefore the result of the last step above is usually followed by a selection and ordering of the highest ranking documents. The number of documents finally returned to the user is in most cases not more than a very small subset of the whole documents set $\{d_j\}$.

3.1.2 Query Optimization in IR

In this subsection we provide a brief overview of the most important optimization techniques in the IR field.

This section is divided in three paragraphs. The first paragraph describes the basic algorithm on which most of the optimization techniques (that we are interested in) are based. The second paragraph shows the possible consequences for the quality of the returned answers when using certain versions of this algorithm. The last paragraph describes some issues for physical design of the data storage which might help increase the profits of the algorithm in the first paragraph.

Effects of a Smarter Algorithm In IR-research a lot of effort has been done on optimizing query processing in IR-systems. Most of the work done in this area is based on some quite simple ideas from [Fag98, Fag99, FM]. In these papers several algorithms are proposed that stop executing as soon as the required answers are computed. Most importantly is that these algorithms and especially the used stop criteria are proven to be correct. This means that using these algorithms under the given constraints will result for sure in the required set of answers or a superset of those.

The basic principle of the algorithms goes as follows (loosely formulated):

Given two sequences of equal length, say $A = [a_i]$ and $B = [b_i]$, of numbers, with $|A| = |B| = n$.

We want to compute a subsequence D of C, with $C = [c_i | c_i = a_i \cdot b_i]$, where $|D| = N$ and $min(D) \geq max(C - D)$, meaning D contains the N highest elements of the product of A and B.

The steps of the algorithm are then (roughly speaking):

[3] We use the term *belief* here because this product is called this way in the system developed at our group.

1. Suppose we order one of the sequences, say A, descending. The resulting sequence A' is therefore a permutation of the original sequence A. Let's call this permutation π. Then:

$$A' = [a'_i | a'_i = a_{\pi(i)}]$$

such that:

$$\forall i \in \{2, \ldots, n-1\} : a'_{i-1} \geq a'_i \geq a'_{i+1}$$

2. Create a sequence B', that has the same ordering as A':

$$B' = [b'_i | b'_i = b_{\pi(i)}]$$

3. Create an empty set E.

4. For each ascending $i \in \{1, \ldots, n\}$:

   (a) Compute $v = a'_i \times b'_i$.

   (b) IF $|E| < N$ OR $v > min(E)$ THEN $E := E \cup \{v\}$.

   (c) IF $|E| > N$ THEN $E := E - \{min(E)\}$.

   (d) IF $max[b'_{i+1}, \ldots, b'_n] \times a'_i < min(E)$ THEN stop looping over i.

5. Create sequence D from set E by ordering the elements ascending. D is the requested top N of the product of A and B.

Both [Bro95] and [CP97] use (modified versions of) this idea to calculate the document belief-contribution per term, also known as the $tf \cdot idf$. Where tf is taken as the A and idf as B or vice versa. In fact, this means that one computes the $tf \cdot idf$ of the most promising tf (or idf) before the $tf \cdot idf$ of the lesser promising tf (or idf), hoping that not all tf (or idf) values will have to be taken into consideration but only a certain, hopefully small, set of 'best' ones. Of course the algorithm takes some extra administrative overhead that will make the algorithm only profitable in cases where the considered part of the ordered sequence A is small enough to compensate this overhead.

Note that tf and idf are not of equal length, so formally these two sequences cannot be used in the mentioned algorithm. By repeating each idf_i value for all tf_{ij} with corresponding i and distinct j this little problem can be fixed. The implementation of IR techniques used in our group uses this trick. Of course, with a slight, more or less trivial, modification of the algorithm, the desired effect can be achieved without repeating values.

Safe and Unsafe methods An extra option is to skip the last step of ordering the final results. In cases of IR this might be very well acceptable for several reasons. Two of these reasons are:

- The top N is the most important issue, the ordering is often less important.
- If the elements in E do not differ much, the ordering of E is disputable since these figures result from statistically based computations of which the error margin might be much greater than that inter-element margin.

This point also brings me to another point in general, noted by [Bro95] as the difference between so-called *safe methods* and *unsafe methods* of optimization:

safe methods This class of methods still return (at least) the required answers in the desired form (i.e. ordering and such).

unsafe methods This class of methods promise to return good answers, but do not guarantee to return all the required answers or in the desired form. Correct answers might be replaced by other, usually just a little less optimal, answers. And, as already addressed above, the answers might not be ordered correctly.

Since IR is by no means exact retrieval, in contradiction to querying a DBMS containing only alphanumerical data, *the* good, unique answer does not exist, or cannot be determined. Only good answers and better answers exist. And even the question which answer is better than the other one is hard to answer solidly. Also, in systems where the querying process is an interactive one with relevance feedback from the user, even the worse elements in an answer set might serve a good purpose of providing the user with an option to tell the system what he/she explicitly does not want.

Therefore a great reduction in query execution time might very well be preferable above a 'better' answer of which the computation takes much more time.

Effects of Physical Design One other important issue is the tuning of the storage system underneath/within the IR-system. By placing the data on disk in the right order, with the right clustering and appended by some efficient access accelerators, a lot of time can be saved, just streamlining disk read IO. [Bro95] uses properties of the IR level to place the document-term statistics efficiently ordered in an inverted file structure. The statistics are stored in descending *idf*. This way he achieves two effects in one:

1. He is able to use an algorithm as described above.
2. He is able to consider a lot of different terms per block read operation. The reason for this is that a large *idf* means that a term occurs in only a few documents. This results in a small data segment in the inverted file for a certain term (per term the list of documents is small). In turn, this means that a lot of those small document-per-term lists can be stacked in one block.

[Bro95] implemented and tested his ideas using INQUERY, and claims interesting performance improvements, and only a minor decrease in precision and recall.

3.1.3 (Dis)advantages of the IR approach

The main advantage of the approach in IR optimization techniques as described previously is the fact that the element-at-a-time iteration while computing the beliefs allows recomputation of the boundaries that form the basis for the stop criterium. As proven by [Fag98, Fag99, FM] these accurate boundaries can result in computation of only a very minimal set of belief candidates which of course is highly computationally efficient.

However, most IR systems tend to be quite monolithic requiring lots of reimplementation for sometimes only minor conceptual model changes. Also the integration with other systems can be difficult or highly inefficient [dVW99].

On the other hand, the theoretical work of [Fag98, Fag99, FM] is clearly very well founded and provides means for very interesting optimizations, which also holds for the work of [Bro95]. Due to the characteristic Zipf distribution of the indexed content data these methods often result in very high profits with often only very limited (when using unsafe methods) or even no (when using safe methods) loss in answer quality. However, these techniques did unfortunately not get implemented in most of the IR systems. The work of [Bro95] was implemented using INQUERY but we are not sure whether it is still used in the current INQUERY implementation.

The main issue that summarizes the negative aspects of IR-systems is that they tend to be quite inflexible to adaptation of new technologies since it requires a lot of code replacements.

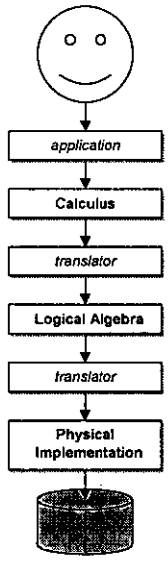


Figure 1: Query processing layers in a typical relational DBMS.

3.2 DB Techniques

DBMSs were originally designed as smart data storage systems which could handle data requests from users on a higher level than the file system of an OS can. In administrative environments relational DBMSs have proven to be quite useful. However, that data usually consisted of just alphanumerical parts of limited (often even fixed) lengths. Due to practical demands most DBMSs nowadays are extensible: new data types and/or operations can be added. Often also other functionality can be added to the DBMS like new index/access structures. Modern DBMSs also often support OO data modeling and access interfaces instead of, or additionally to, the relational way of handling the data, in an attempt to cope with the growing number of different uses and stored types of data.

Note that several types of DBMSs exist. Two of the most important ones are:

- relational DBMSs (or RDBMSs),
- object oriented DBMSs (or OODBMSs).

RDBMSs is by far the oldest of these two types, and also the most widely used one. OODBMSs might seem very interesting due to their OO nature but still are often not much more than persistent object stores which cannot compete in efficiency and scalability with their relational counterparts. Also attempts to build an OO layer on top of an RDBMS resulted in most cases in unrealistically unhandy or slow systems. Since our research has more to do with relational than OO data base technology we will not elaborate on the latter one any more.

For a more complete overview on relational database theory see [Dat95].

3.2.1 Query Processing in a DBMS

Note that relational DBMSs are designed to operate on sets providing operations to retrieve strictly defined subsets or elements given a query. This is in contrast with the basic design of IR systems which is usually referred to as providing inexact retrieval (as opposed to exact retrieval in a relational DBMS).

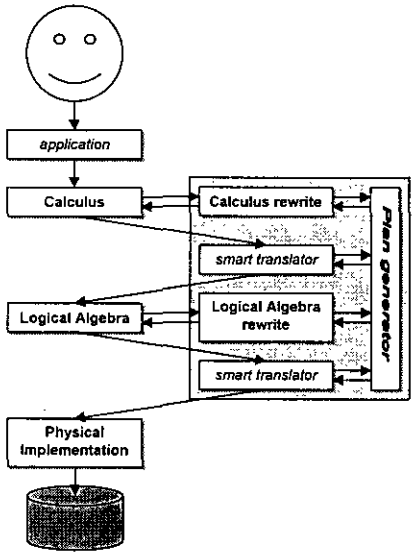


Figure 2: Adding query optimization to the three layered query processing architecture.

Most relational DBMSs are currently build in divided in three, independent, layers, on top of each other:

1. Conceptual level
2. Logical level
3. Physical level

The conceptual level is the most abstract level and is mapped onto the logical level, which in turn is mapped onto the physical level. Queries are formulated in a declarative way at the conceptual level and translated into an algebra at the logical level. The algebra then is translated to methods on the physical level [Figure 1].

The current standard conceptual level query language is SQL[4], with both calculus and algebra elements in it [Ste95]. The algebra at the logical level and the operators at the physical level are system specific whereas the implemented SQL version only shows minor extensions and/or differences, usually concerning constructs not or not clearly defined in the standard. Another importing issue to note is that the conceptual level language (i.e. SQL) usually is tuple oriented whereas the two lower levels usually are set based.

3.2.2 Query Optimization in a DBMS

As described in [Dat95, Ste95, Cha98] and shown in [Figure 2] optimization can take place at almost every level in the system.

Just a naive translation from a calculus expression into an algebra expression will in most cases result in a so called most costly expression [Ste95]. This means that the resulting algebra expression has even worse execution time than nested loop iteration over all elements in the data set processed for the query. Therefore it is not also possible to optimize at all levels but necessary to take advantage of every possibility, too.

The basic conceptual components of a relational query optimizer are [Figure 2]:

[4]SQL = Structured Query Language

Calculus rewriter In this phase it is often difficult to really optimize an expression. In most cases it it used to transform the expression into some normal form thus limiting the search space for lower level optimization processes. However, normalization should be done careful to avoid decreasing execution performance promises.

Calculus to logical algebra translator It is at this level that a lot of performance can be gained by trying to achieve a not too naive translation since otherwise the resulting algebra expression in most cases would provide a very inefficient starting point for any further optimization. However, in literature not much information is available on this important link in the chain of query processing.

Logical algebra rewriter At this level algebra equivalence rules (based on common algebra properties like commutativity, associativity, distributivity, idempotency, etc.) are used to transform the expression into a form that should translate more easily and more efficiently into the physical algebra at the physical layer. Most systems do not use a cost function at this level but just (simple) heuristics that usually produce better expressions than the initial input. Commonly known heuristics are: push select/project down or the more sophisticated predicate move around [HS93, LMS94, Hel98], most restrictive join first.

Logical algebra to physical algebra translator This stage aims at translating the logical algebra expression into the most efficient program at the physical level. It is usually at this stage that a cost model of the physical layer is used to find the cheapest executable program. The costs of a certain program are computed by estimating the costs of each operator given known or estimated data properties such as number of tuples [Dat95], ordering [SSM96, CK97a, CK97b, CK98, DR99], availability of access accelerators, etc. Also for each operator the properties of the results are computed/estimated which in turn can server as input for costs computations of depending operators [Cha98].

Plan generator The plan generator is responsible for keeping track of all possible plans that have been or are taken into consideration. In [Figure 2] we placed it as a global component next to the three other stages. This is just a way of displaying things because at a certain level knowledge of all lower levels might be required. The reason that such knowledge is required at the higher level is that a certain decision at that level might result in better or worse optimizable/executable query plans at subsequent lower levels, thus influencing that high level decision.

3.2.3 (Dis)advantages of the DB approach

The main advantages of any DBMS is its flexibility to store and manipulate a lot of data in a more or less abstract manner. In particular the ANSI/SPARC properties allow changes at a certain level without having to modify large parts in other levels of the system [Dat95]. The price of this layered structure is a lot of inefficiency. Fortunately the fact that many systems use one or more algebras to handle the data allows for a great deal of optimization. The other advantage is that the optimization is usually better than in cases where a user needs to implement the physical operator implementation by himself: the system can incorporate of the shelf optimal implementations which resembles the union of programming skills of several expert programmers together [Dat95]. Also because of the set-based/bulk behavior of, in particular relational systems, IO can often be arranged more efficiently since several types of overhead can be eliminated over larger quantities of data.

One of the few disadvantages of this bulk behavior and/or lack of control of how elements are iterated over finally, is the fact that stop criteria as the ones used in section 3.1.2 often cannot be computed equally well, disallowing such efficient breaks in computing over large sets.

4 Our Approach

In this section we describe our approach concerning top N MM query optimization, given the problems and solutions provided by the IR and DB research areas as described in the previous sections.

First we describe briefly our DB environment. Next we give an overview of our approach including some first results.

4.1 Our DB Environment

In this section we briefly describe the software systems that we have been using and/or developing lately. We start with introducing Monet, a main memory DBMS developed at CWI[5]. Next we will introduce Moa, a structured object algebra, currently build on top of Monet. After introducing both Monet and Moa, we give an introduction to the MM retrieval work that has been going on at that this group and that is being implemented using Moa.

4.1.1 Monet

Monet[6] is the main memory DBMS that is being developed by CWI. Monet is able to cope with modern day DBMS requirements emerging from areas like DSS[7] and MM retrieval. These areas often use wide tables with very many elements/tuples. However, during query processing only a few columns are needed at a time. To exploit this property Monet uses full vertically fragmentation, resulting in a binary data model in which tables consist of only two columns, so called BATs[8]. Relations that need more than two columns can be made by joining several BATs. This vertical fragmentation allows Monet to keep the hot set[9] in main memory, in contrast with most other DBMSs. This way Monet outperforms most other systems when used in situations that wide relations with many elements do exists but only a few fields are part of the hot set, like often occurs in DSS and MM retrieval. Monet has proven to be fast in those cases using several benchmarks like OO7 [vdBvdH96], TPC-D [BWK98] and Wisconsin.

In the scope of this paper it is also important to note that Monet uses a dynamical physical query optimization system, choosing the right physical operators just before the operator needs to be executed, thus ruling out the usual error introducing guess work during optimization. The resulting better optimization decisions, facilitate speeding up query execution even more.

4.1.2 Moa

Moa [BWK98] is an extensible structured object algebra which is mapped on top of a binary physical algebra. Currently Monet is used as the binary algebra layer underneath Moa. Moa is designed as an intermediate language, e.g. logical algebra, on which a high level language like OQL could be mapped. Moa was originally developed for GIS[10] and is being extended to serve other areas, like content based retrieval, as well.

One of the key advantages of Moa over other systems is that the structured object nature of Moa allows for unlimited nesting of structures without loss of performance. Most other systems that allow deep/unlimited nesting usually tend to be very slow since they iterate in a depth-first manner through the nested structures during query processing. In case of a large database with

[5] CWI = Center for Mathematics and Computer Science, Amsterdam, The Netherlands.
[6] See [BK95, KBQ+97, BK] or on the CWI web site http://www.cwi.nl/~monet for more info.
[7] DSS = decision support system.
[8] BAT = Binary Association Table
[9] hot set = parts of the database under consideration at a certain moment during query processing
[10] GIS = geographic information system.

many levels of nesting this element-at-a-time query processing has proven to highly inefficient compared to the set-at-a-time way of query processing common to most relational DBMSs, that do not allow nesting at all. Moa combines the best of both worlds in that it allows nesting but processes queries in a set-based manner.

4.1.3 MM Retrieval

Exploiting the facilities of Moa and Monet a MM DBMS is being built. Currently a basic extensible open distributed architecture has been set up [HVBB98] providing a framework around Monet and Moa. With respect to the content retrieval part currently a text retrieval system has been implemented in the Moa/Monet environment [dV98, dVW99]. The retrieval model originally used resembles the INQUERY inference network [CCH92]. Lately another retrieval model [Hie98], which had proven been to work quite good on TREC, has been implemented in the system as an alternative for the INQUERY model. Our text retrieval DBMS using this new model has is been evaluated at TREC-8 [VH99].

We plan to generalize the text retrieval system to become a MM retrieval system. We have already been experimenting with audio retrieval (only on a very small an primitive scale) and image retrieval [Doo99] using the generalized text retrieval framework.

It is in the context of this MM retrieval setting where our top N MM query optimization research is situated.

4.2 Our Optimization Efforts

When looking at top N MM query optimization, in particular in a setting where we use Moa on top of Monet, the following three main topics are interesting to look into:

- Fragmentation,
- Intra v.s. inter object/ADT optimization,
- Real MM querying.

These three topics can, to some extent, be linked to the three levels in the ANSI/SPARC architecture, often, although not quite correctly, designated by:

- Physical level,
- Logical level,
- Conceptual level.

Furthermore it should be noted, that right from the start, for any of the topics, it is also necessary to define and set a baseline for the performance of the system that will serve as a reasonable reference point to compare an optimized version of the system with. Note that performance in an IR or MM retrieval system does comprise more than execution time alone. Also the precision and recall are part of the performance. Since no combination function for these three components (execution time, precision, recall) exists, performance should be noted as the 3-tuple. Display techniques like a 3D-projection of a plot of the performance points for different situations might provide an easy way for humans to compare the results in a flexible way (i.e. not aggregating the 3-tuple into one number, thus not throwing away a lot of information).

4.2.1 Fragmentation [Physical design]

To facilitate any optimization on a higher level and to make use of the main memory characteristics of Monet we need to (horizontally) fragment the data in Monet. In case of a real world application the size of the data even will be that huge that fragmentation will be necessary to get things running anyway, since otherwise memory will be too limited to do anything at all.

Looking at paragraph 3.1.2, where the ideas of [Bro95] concerning storage of IR data are explained, fragmenting *tf* (or *ntf*) and *idf* (or *nidf*) vectors in Monet in a similar manner based on *idf* (respectively *nidf*) seems to be an interesting first choice.

However, since we are dealing with a database system that preferably operates set based, in contrast with the element-at-a-time approach used in the INQUERY system, the ideas of [Bro95] are note directly applicable in our case. In other words: we have to adapt these ideas to suit a set based way of query processing. Since we want the best of both worlds a way in between seems to very interesting. A possible solution might be exploiting the commonly known Zipf distribution of the IR data. The Zipf law for the English language says that that the distribution of terms over a set of documents is hyperbolically correlated to the number of those documents they occur in. This means for example that 5% of the terms is related to 95% of the total amount of data in the retrieval system. These are the terms that occur in a lot of documents and therefore have a low *idf*. The other 95% of the terms only is responsible 5% of the space.

Further note that the *tf* vectors used to compute the beliefs are in fact inverted lists noting a *tf* for each term per document it occurs in. This means that using the observation about the Zipf distribution of this vector, we get two (horizontal) fragments:

- a fragment containing the *tf* values corresponding with the other 95% of the terms, with higher *idf*, only using 5% of the space.
- a fragment corresponding to 5% of the terms corresponding with the terms with the lowest *idf*, but taking up 95% of the space,

Now, we can start the evaluation of a typical IR document ranking query by ignoring the second fragment that takes 95% of the space. This way we can compute a quite good belief estimate by only having to consider 5% of the original document statistics. Assuming that time complexity is linear in the size of the considered vectors this reduces execution time with 95%. Some first experiments based on this idea show that execution times at least drop with 60%.

As described in paragraph 3.1.2, this technique is unsafe, which is obvious, since not all terms are taken into consideration thus ignoring belief contributions from those terms. And also, although the terms that are ignored have a low *idf* and are therefore unlikely to contribute much to document beliefs, this does not exclude the possibility that some indeed would have contributed significantly in the case when they have a very high *tf* that compensates for the low *idf*. Resulting from the same early experiments that showed the 60% time profit we found that in the case of the above suggested fragmentation precision drops a few percents and recall drops not more than 30%.

Some additional, but still primitive, experiments using different percentages for fragmentation and/or more than two fragments (up to 20) have shown that better precision and recall, approaching the non-optimized versions, or higher executions time gains, over 90%, can be achieved. More thorough experiments need to be done to get a better view on the trade off between execution time and answer quality. We expect to be able to reach an optimization level that still keeps the quality of the answers in terms of precision and recall equal to the non-optimized query processing system but with at least 25% faster execution times. The test were performed using the TREC topics 300 till 350 on the Financial Times sub collection of TREC.

We also are thinking of some other experiments. We did for instance neglect any restrictions that need be obeyed when one wants to really exploit the benefits of main memory execution, as provided by Monet. When running main memory it is important to keep the hot set scaled to fit in

main memory at all time. This brings us to the question what the best fragment size and number of fragments are to keep swapping to disk as limited as possible. In case of too many too small fragments, a lot of administrative overhead will result in a loss of speed. Also small fragments contain fewer information needed at query processing time increasing the probability that several fragments need to processed to reach a certain level of answer quality. In case of too large fragments the chances that other fragments need to be considered due to a lack of information are low. However, processing a single large fragment can take much time, approaching the evaluating time of an unfragmented database or even worse due to administrative overhead. Also, the case that extra fragments need to be considered still is not excluded and when that occurs it increases execution time extensively.

4.2.2 Intra v.s. inter object/ADT optimization [Logical design]

Although the techniques used in the fragmentation approach described in the previous paragraph are mostly realized on the physical level, in our case Monet, it certainly has several consequences for the logical level. First of all it is obvious that given the different physical lay out the translation of logical algebra expressions to physical algebra expressions need to be adapted accordingly. Secondly, the new translations allow the easy introduction of some special top N operators [Bro95, CK98] on the logical level, that can exploit the new opportunities in a better way than was possible before. This can of course be seen as giving up some of the independency between the two layers but it is also likely to be interesting from an optimization point of view since it allows more efficient expressions at the logical level when dealing with top N queries.

However, since the best mapping of logical algebra expressions on physical expressions is often dependent on the application area, most systems provide extensibility, as does Moa. This allows for adding new structures and/or operators to the algebra to suit specific needs, thus adding the best possible translation(s) for that case. Extensibility of systems is quiet common but keeping optimization at an acceptable level is still a very large problem.

In the PREDATOR project [SP97] a partial solution for this problem is given. Their system architecture does allows for new structures and operators to be added using, what they call, E-ADTs, or Enhanced ADTs. E-ADTs are ADTs but also provide optimization algorithms for optimizing operations in that ADT. The global logical query optimizer delegates optimization to the E-part of the E-ADT if it finds itself unable to optimize the ADT operators in a query expression.

This is, as we said, only a partial solution. E-ADTs are only aware of themselves so they are only able to optimize when two or more operations on that particular E-ADT are involved [Example 1] but fail when operators of distinct ADTs interact [Example 2].

Example 1 (E-ADT optimization succeeds [SP97]) *Consider an Image ADT with two operators: clip and rotate.*

The clip operator takes as operands an image and the x and y coordinate of the upper left and lower right corner of a rectangle within the bounds of the given image.

Suppose we have an image myimage of 40 pixels wide and 60 pixels high.

An example use of the clip operator would be:

$$clip(myimage, 0, 0, 9, 19)$$

which produces a sub image of 10 pixels wide and 20 pixels high of myimage containing its upper left corner (image x coordinates increase in the normal way to the right but y coordinates increase usually downwards instead of upwards).

The rotate operator takes as operands an image and the angle to rotate over in degrees, counter clockwise.

An example use of the rotate operator would be:

$$rotate(myimage, 90)$$

which produces the a copy of myimage placed on its left side.

The following query, combining these two operators, is also valid:

$$clip(rotate(myimage, 90), 0, 0, 9, 19)$$

which produces the upper left corner area of 10 pixels wide by 20 pixels high of the rotated version of the image.

An alternative expression with exactly the same answer is:

$$rotate(clip(myimage, 20, 0, 9, 39), 90)$$

Since the rotate in the second expression has to rotate a smaller images than in first expression, whereas the clip region stays of the same size, the second expression will be executed faster. The E-ADT system now provides the means to the optimizer to translate the first expression, when encountered, into the second one.

Example 2 (E-ADT optimization fails) *Let's look a a system that has a LIST and a BAG structure, both provided by separate ADTs. Also, let's assume that both have a select operator that behaves in the usual way, and takes an upper and lower bound to select a range of values. For example*

$$select([1, 2, 3, 4, 4, 5], 2, 4)$$

selects all elements from the list $[1, 2, 3, 4, 4, 5]$ *with values ranging from* 2 *up to and including* 4. *This will result in the list*

$$[2, 3, 4, 4]$$

Besides the select, LIST also provides a projecttobag operator which produces a BAG containing all the elements of the LIST the operator acts on. For example

$$projecttobag([1, 2, 3, 4, 4, 5])$$

results in the bag

$$\{1, 2, 3, 4, 4, 5\}$$

Now, consider the following expression

$$select(projecttobag([1, 2, 3, 4, 4, 5]), 2, 4)$$

Current optimizer technology, including the E-ADT system of PREDATOR, cannot optimize this expression. However, anybody can see that

$$projecttobag(select([1, 2, 3, 4, 4, 5], 2, 4))$$

produces exactly the same answer but can be executed more efficient than the original expression. The second expression can be evaluated even more efficient when the system is aware of the ordering of the elements, which in case of a list is well defined, but formally does not exist for a bag.

Now note that a top N operator is in fact a special select operator on a list structure. Furthermore, more sophisticated versions of expressions like in [Example 2] often occur in top N queries (it goes to far to explain those expression here). This means that it would be preferable to have some optimizer system that can cope with such expressions incorporating operators from different ADTs. We propose to introduce an extra optimizer layer, between the global logical optimizer and the optimizer parts within the (E)ADTs, to fill the gap. We call this an inter ADT (or in generalized case: inter object) optimizer, whereas we call the E part of the E-ADT system an intra ADT (or intra object in the generalized case) optimizer.

4.2.3 Real MM querying [Conceptual design]

Our final challenge is to make our top N query optimizer MM aware. Since we plan to generalize our DBMS to act as a *real* MM DBMS that is able to provide facilities to query different media types in an integrated manner in one query expression, we also get a system that provides new opportunities to optimize. Consider for instance the case in which we deal with both alphanumerical data and images. The alphanumerical data might contain classification information or who the author is of a certain image, so we might be interested in asking for images with certain content and require they are made by a given author. The big question is now, in what order do we process the different query parts: are we going to select on the author field first and then look at the content or do we select on image content prior to selecting the ones with the right author. In this case one would tend to prefer the first option since selecting on alphanumerical attributes usually is much cheaper than processing images on content. However, when dealing with a query that incorporates two equally *expensive* media types this order is not so obvious. To solve this problem one again needs inter ADT/object optimization facilities, in this case augmented with cost information. It is only at the inter ADT level that a system can decide properly, having an idea of the execution costs of the involved ADT operators given their operands, what operator should be executed first or second. This type of cost based query optimization at the logical level and in a MM context is entirely new and we see it as a very interesting and promising area to look into. Note that the conceptual level now having to deal with more than one media type is the key reason for the necessity of the introduction of this new cost model.

5 Concluding Remarks

By providing an overview of existing IR and database optimization techniques and describing three important research topics we have tried to demonstrate that still a lot of work has to be done when one wants to optimize top N queries in a MM database environment.

The three topics as described can be seen as relatively separate topics. On the other hand, it also might be clear that they facilitate and use each other to a certain extent. We are also aware of the fact that the mentioned topics might very well be much larger than depicted here and therefore might not all be solved in the near future due to time limitations. For this reason we have chosen to just start at the ground level, which fragmentation. Since this is more a physical issue, whereas our interest in mainly in the logical level of Moa, we will set up just a basic fragmentation and do not plan to investigate it more than necessary to suit the use Monet environment. That way we can proceed on the, in our opinion, more interesting part of inter ADT optimization. The *real* MM query optimization will probably to far fetched for the near future and is left for the long term.

Our goal for the next few years is to set up a good physical basis, as said, and build a prototype optimizer for Moa, including an inter ADT layer. If all works out fine this will provide a new kind of query optimizer that, in contrast with the current query optimizers, is suited to optimize modern day DBMS in a MM environment.

References

[ACM97] ACM SIGMOD, *Proceedings of the 1997 ACM SIGMOD International Conference on the Management of Data*, Sigmod Record, ACM, 1997.

[ACM98] ACM SIGMOD, *Proceedings of the 1998 ACM SIGMOD International Conference on Principles of Database Systems, Seattle, USA*, Sigmod Record, ACM, 1998.

[BK] Peter A. Boncz and Martin L. Kersten, *MIL Primitives For Querying A Fragmented World*, VLDB Journal, accepted.

[BK95] Peter A. Boncz and Martin L. Kersten, *Monet: An Impressionist Sketch of an Advanced Database System*, Basque International Workshop on Information Technology, Data Management Systems, San Sebastian, Spain, July 19-21, 1995 (BIWIT'95), IEEE Computer Society Press, jul 1995.

[Bro95] Eric W. Brown, *Execution Performance Issues in Full-Text Information Retrieval*, Ph.D. Thesis/Technical Report 95-81, University of Massachusetts, Amherst, okt 1995.

[BWK98] Peter Boncz, Annita N. Wilschut, and Martin L. Kersten, *Flattening an Object Algebra to Provide Performance*, 14th International Conference on Data Engineering, February 23-27, 1998, Orlando, Florida, IEEE Transactions on Knowledge and Data Engineering, IEEE Computer Society, feb 1998.

[CCH92] J.P. Callan, W.B. Croft, and S.M. Harding, *The INQUERY Retrieval System*, 3rd Internationl Conference on Database and Expert Systems Applications, 1992, pp. 78–83.

[Cha98] Surajit Chaudhuri, *An Overview of Query Optimization in Relational Systems*, In *Proceedings of the 1998 ACM SIGMOD International Conference on Principles of Database Systems, Seattle, USA* [ACM98], pp. 34–42.

[CK97a] Michael J. Carey and Donald Kossmann, *On Saying "Enough Already!" in SQL*, In *Proceedings of the 1997 ACM SIGMOD International Conference on the Management of Data* [ACM97], pp. 219–230.

[CK97b] Michael J. Carey and Donald Kossmann, *Processing Top and Bottom N Queries*, IEEE Bulletin of the Technical Commitee on Data Engineering **20** (1997), no. 3, 12–19.

[CK98] Michael J. Carey and Donald Kossmann, *Reducing the Braking Distance of an SQL Query Engine*, 24th VLDB Conference, New York, USA, 1998, VLDB, 1998, pp. 158–169.

[CP97] Douglass R. Cutting and Jan O. Pedersen, *Space Optimizations for Total Ranking*, Proceedings of RAIO'97, Computer-Assisted Information Searching on Internet, Quebec, Canada, June 1997, jun 1997, pp. 401–412.

[Dat95] C.J. Date, *An Introduction to Database Systems*, 6th. ed., 1995, ISBN 0-201-54329-X.

[Doo99] M.G.L.M. van Doorn, *Thesauri and the Mirror retrieval model: A cognitive approach to intelligent multimedia information retrieval*, Master's thesis, Faculty of Computer Science, University of Twente, Enschede, The Netherlands, jul 1999.

[DR99] Donko Donjerkovic and Raghu Ramakrishnan, *Probabilistic Optimization of Top N Queries*, Technical Report CR-TR-99-1395, Department of Computer Sciences, University of Wisconsin-Madison, 1999.

[dV98] Arjen P. de Vries, **miЯRor**: *Multimedia Query Processing in Extensible Databases*, 14th Twente Workshop on Language Technology, Language Technology in Multimedia Information Retrieval (Enschede, The Netherlands), University of Twente, dec 1998, pp. 37–47.

[dVW99] Arjen P. de Vries and Annita N. Wilschut, *On the Integration of IR and Databases*, 8th IFIP 2.6 Working Conference on Data Semantics 8, 1999.

[Fag98] Ronald Fagin, *Fuzzy Queries in Multimedia Database Systems*, In *Proceedings of the 1998 ACM SIGMOD International Conference on Principles of Database Systems, Seattle, USA* [ACM98], pp. 1–10.

[Fag99] Ronald Fagin, *Combining fuzzy information from multiple systems*, Journal on Computer and System Sciences **58** (1999), no. 1, 83–99, Special issue for selected papers from the 1996 ACM SIGMOD PODS Conference.

[FM] Ronald Fagin and Yoëlle S. Maarek, *Allowing users to weight search terms*, Retrieved from authors website.

[Hel98] Joseph M. Hellerstein, *Optimization Techniques for Queries with Expensive Methods*, ACM Transactions on Database Systems **23** (1998), no. 2, 113–157.

[Hie98] Djoerd Hiemstra, *A Linguistically Motivated Probabilistic Model of Information Retrieval*, Proceeding of the second European Conference on Research and Advanced Technology for Digital Libraries, ECDL'98 (Christos Nicolaou and Constantine Stephanidis, eds.), Springer-Verlag, 1998, pp. 569–584.

[HKWY97] Laura M. Haas, Donald Kossmann, Edward L. Wimmers, and Jun Yang, *Optimizing Queries across Diverse Data Sources*, 23th VLDB Conference, Athens, Greece, 1997, VLDB, 1997.

[HS93] Joseph M. Hellerstein and Michael Stonebreaker, *Predicate Migration: Optimizing Queries with Expensive Predicates*, Proceedings of the 1993 ACM SIGMOD International Conference on the Management of Data, Washington DC, Sigmod Record, ACM SIGMOD, ACM, may 1993, pp. 267–276.

[HVBB98] E. van het Hof, A.P. de Vries, H.E. Blok, and H.M. Blanken, *Een architectuur voor multimedia databases [Eng.: An Architecture for Multimedia Databases]*, Conferentie Informatiewetenschap 1998 [Eng.: Information Science Conference 1998] (E. de Smet, ed.), Werkgemeenschap Informatiewetenschap, dec 1998, In Dutch, pp. 89–106.

[KBQ+97] M.L. Kersten, P. Boncz, W. Quak, N. Nes, and J. Karlsson, *The Monet System, version 4.0 (BETA)*, Tech. report, CWI and University of Amsterdam, Amsterdam, oct 1997.

[LMS94] Alon Y. Levy, Inderpal Singh Mumick, and Yehoshua Sagiv, *Query Optimization by Predicate Move-Around*, 20th VLDB Conference, Santiago, Chile, 1994, VLDB, 1994.

[SP97] Praveen Seshradri and Mark Paskin, *PREDATOR: An OR-DBMS with Enhanced Data Types*, In *Proceedings of the 1997 ACM SIGMOD International Conference on the Management of Data* [ACM97], pp. 568–571.

[SSM96] David Simmen, Eugene Shekita, and Timothy Malkemus, *Fundamental Techniques for Order Optimization*, Proceedings of the 1996 ACM SIGMOD International Conference on the Management of Data, Montreal, Canada, Sigmod Record, ACM SIGMOD, ACM, 1996, pp. 57–67.

[Ste95] Hennie Steenhagen, *Optimization of Object Query Languages*, Ph.D. thesis, University of Twente, Enschede, The Netherlands, oct 1995.

[vdBvdH96] C.A. van den Berg and A. van der Hoeven, *Monet meets OO7*, Proceedings of OODS'96, jan 1996.

[VH99] A.P. de Vries and D. Hiemstra, *The* miЯRor *DBMS at TREC*, Proceedings of the Seventh Text Retrieval Conference TREC-8 (Gaithersburg, Maryland), nov 1999, To appear.

AUTOMATIC CATEGORIZATION OF MAGAZINE ARTICLES

MARIE-FRANCINE MOENS and JOS DUMORTIER

Katholieke Universiteit Leuven, Belgium
Interdisciplinary Centre for Law & IT (ICRI)
Tiensestraat 41
B-3000 Leuven
Belgium
Tel: xx/32/16/325383
Fax: xx/32/16/325438
e-mail: {marie-france.moens,jos.dumortier}@law.kuleuven.ac.be

Abstract

Automatic text categorization is an important research area and has a potential for many text-based applications including text routing and filtering. Typical text classifiers learn from example texts that are manually categorized. In this paper we discuss the categorization of magazine articles with broad subject descriptors. We especially focus upon the following aspects of text classification: effective selection of feature words and proper names that reflect the main topics of the text, and training of text classifiers. The χ^2 test, which is sometimes used for selecting terms that are highly related to a text class, is applied in a novel way when constructing a category weight vector. Despite a limited number of training examples, combining an effective feature selection with the χ^2 learning algorithm for training the text classifier results in a satisfactory categorization of new magazine articles.

1. Introduction

An important Belgian publisher provides his magazine articles on the Internet for on-line purchase. Controlled subject descriptors are assigned to the articles. The descriptors are mainly used to route the articles to magazine subscribers who are interested in electronic articles that treat specific topics. A fast routing of articles immediately after their publication is important, hence the interest in automating the descriptor assignment or text categorization.

Automatic text categorization is an important research area and has a potential for many text-based applications including text routing and filtering. The purpose of our research is to develop adequate techniques for classifying a variety of articles of different magazines and columns and to test the techniques upon a large corpus of articles. The *descriptors* regard the broad subjects of the stories (e.g., music, film, investments). Our research regards experiments with different text categorization algorithms. The algorithms learn the classification patterns from example texts that are manually classified. One algorithm uses the χ^2 test for pattern recognition with satisfying results. We put a strong focus upon effective selection of content terms and proper name phrases. A detailed overview of the research is given in Moens and Dumortier (in press).

2. Text corpus and output of the demonstrator

The articles of the text corpus were published in 1998 in magazines such as "Knack", "Weekend Knack", "Trends", and "Cash!". They are written in Dutch and are very heterogeneous in content and structure. The articles cover a variety of subjects in domains, such as politics, economy, finance, life style, arts, sports, and

many others, and often interweave different subject domains. The articles belong to different columns of the magazines. This is reflected in their structure. Most of them follow the schema of the written news story consisting of a headline, lead paragraph, attribution section, and body of the story. Other schemata are also present, such as a list of film titles with explanatory sentences, or a simple address of a restaurant. A few articles are so-called satellite articles: They are small texts that elaborate on a subtopic of another large article. The article texts are of varying length ranging from a few paragraphs to multiple pages (most articles fall in the range of 6000 to 25000 bytes, some very short ones of less than 1000 bytes exceptionally occur).

The articles have descriptors attached that were assigned by the professional indexers of the publisher. Usually, one, sometimes two, and in very rare cases three subject descriptors are ascribed per article. These broad subject descriptors are used in matching article profiles with users' profiles in a routing task.

We used articles of the text classes CAR, INVESTMENTS, STOCK MARKET, CULINARY, FILM, COMPUTER SCIENCE, INTERNATIONAL, LITERATURE, MARKETING, MUSIC, POLITICS, SPORTS, TOURISM, and REAL ESTATE. The publisher has defined other classes, but the number of their members in the text corpus is too small to consider them in the experiments. For each class, a sample of articles was manually analyzed, yielding the following characteristics of the texts of the classes.

1. The texts in the class CAR often describe new car and motor models and give their technical characteristics. They often exhibit a technical vocabulary. Sometimes, a text has the form of an index card that contains the technical details of the car.
2. The texts of the class INVESTMENTS bear upon different forms of investments (e.g., bonds, stocks, art, and real estate). They often overlap with texts in the classes STOCK MARKET and REAL ESTATE.
3. The texts of the class STOCK MARKET describe stock exchanges. They sometimes describe products of companies that offer stocks, which might result in a rich vocabulary.
4. CULINARY describes culinary books, recipes, wines, restaurants, and cafés. The texts often exhibit a rich vocabulary due to descriptions of historical settings and locations of the places or to the variety of the ingredients of recipes.
5. The texts of the class FILM describe new films. Part of this description is rather technical, but another part of it gives a summary of the story of the film. The stories enrich the vocabulary of this class. Sometimes, an article contains a list of film titles and a short description of each film.
6. Texts of the class COMPUTER SCIENCE are technical in nature. They contain descriptions of companies and products.
7. The texts of the class INTERNATIONAL cover events outside Belgium. The main linguistic expressions that cue this class are often the names of foreign countries and important foreign personalities. The vocabulary in this text class is very rich. There is an overlap with the class POLITICS.
8. The texts of the class LITERATURE are commonly reviews of newly published books. As in the class FILM, the content is shortly described. But, there are some specific, technical cues such as references to the type of work, ISBN number, number of pages, and publisher.
9. The class MARKETING mostly contains detailed descriptions of products that are marketed. The products can be almost anything (e.g., animal food, insurance, and clothing). Sometimes, an article discusses multiple products. The vocabulary in these texts is very heterogeneous.
10. The class MUSIC contains texts about classical and modern music. The texts contain technical details and references to known composers.
11. The texts of the class POLITICS contain political events. They are rich in names of political personalities, parties, and organizations.
12. The texts of the class SPORTS describe sport events. Often, the technical vocabulary of a specific sport is present.
13. The class TOURISM contains travel stories and promotions of foreign places, cultures, and hotels. The texts are usually long with a rich vocabulary. However, an article in the form of an information scheme is possible.
14. The texts of the class REAL ESTATE describe the location, area, rent, price, and other characteristics of the real properties.

A corpus of more than 2650 articles was split: two thirds of the articles were used for training and one third for testing. A classifier was learned for the 14 classes. The training corpus contains the following distribution of classes. There are about 300 examples of the class MARKETING and about 200 examples of the class CULINARY. There are about 150 examples of the classes INVESTMENTS, STOCK MARKET, TOURISM, and SPORTS, about 100 examples of the classes FILM, CAR, COMPUTER SCIENCE, MUSIC, and

LITERATURE, and about 50 examples of the classes INTERNATIONAL, POLITICS, and REAL ESTATE. In the test corpus, the classes are present with about equal proportions.

A demonstrator was built in the programming language C on a Sun™ SPARC station 5 under Solaris® 2.5.1. It learns a text classifier from the training set and automatically assigns descriptors to new article texts with the learned classifier (Figure 1). It was agreed with the publisher that the system must simulate the manual process of assigning one or two descriptors to the articles, which reflect the main topics of the article.

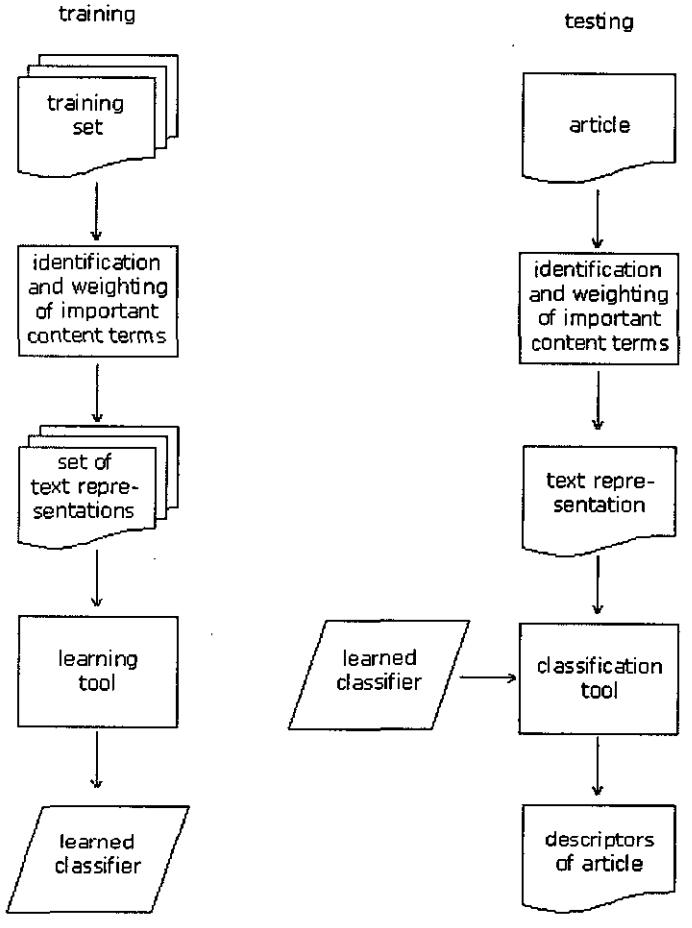


Figure 1: Main components of the demonstrator.

3. Methods

Humans reliably identify relevant texts for a certain subject or classification domain by skimming the texts for specific word patterns and their contexts. *Knowledge bases* that describe the word patterns and their relation with the text classes have been successfully applied in automatic text categorization (e.g., Hayes, 1992; Moens & Uyttendaele, 1997). The text is skimmed for cue patterns defined in a rule or frame base, possibly followed by an evaluation of the logical constraints or of a minimum frequency of occurrence in the text imposed on the patterns. This indicates that surface text features can be identified that successfully discriminate the subject and classification codes linked to a text. Direct representation of this knowledge is a time and effort consuming task, which is only justified when the knowledge is restricted. In other circumstances, machine learning methods provide an interesting alternative. Techniques of *supervised learning* are common in text categorization (for an overview of automatic text categorization: see Moens, in press, p. 101 ff.). In general, they involve the construction of a classification function from a large set of example texts for which the true classes are known. The function agrees with the training instances, i.e., for a given class it classifies the positive examples as members of the class and discards the negative examples, and is hopefully predictable to classify new, previously unseen texts.

Because of the large and heterogeneous subject domain, we use machine learning techniques to acquire the textual patterns that imply the text classes from an example or training set. An example text is represented as a set of features (words and phrases). A new text is equally represented as a set of features. The methods for classifying the magazine articles comprise an initial feature selection to identify important content terms in the texts, learning algorithms, and assignment of subject descriptors.

Words and proper names are the salient features involved in classifying magazine articles. The number of different features in a corpus of magazine articles is enormous. Because the text classes regard the main topics of the texts, it is important to identify content terms that relate to the main topics and to discard terms that do not bear upon content or treat only marginal topics in training and test corpus. Proper names are identified with heuristic rules that take into account patterns of capitalized words and reoccurrence of the names in the texts. Other content words are selected after elimination of stopwords. A stoplist of 879 non-content words is built based upon their syntactic classes. The stoplist contains function words such as articles, prepositions, auxiliary verbs, and others. Numbers are not accounted for. Currently, no form of stemming is used, except for the use of conjugated forms of auxiliary verbs in the stoplist. After removal of stopwords and numbers, we consider two different approaches for selecting important topic terms. In a first approach, words and proper names with a high weight are selected. Terms are weighted by their frequency of occurrence in the text divided by the maximum frequency a content term occurs in the text (length normalization factor). In a second approach, words and proper names are selected from the beginning of the article, which usually includes the discourse segments of the headline, lead, and the attribution of the article.

For recognizing the classification patterns and assignment of subject descriptors, we implemented several statistical algorithms.

In *Bayesian independence classification* (Maron, 1961; Fuhr, 1989; Lewis, 1995) the posterior probability that a new, previously unseen text belongs to a certain text class given its features (here words and proper names) is computed based on the probabilities that the individual features are related to the class. Probability estimates of the individual features are based on the co-occurrence of the text class and selected features in the training corpus, and on the assumption of their linkage. The computation of the probability that a document text belongs to a specific class given the probability of its features is simplified by using the theorem of Bayes, which assumes that the probabilities of the features are independent.

The *Rocchio* (Rocchio, 1971; Lewis, Schapire, Callan, & Papka, 1996) and the χ^2 *algorithms* generalize the positive and negative examples of each class into a category weight vector. The components of this vector are the text features (words and proper names) of the example texts. The weights indicate the strength of their relationship with the subject class.

For a new article to be classified by the Bayesian classifier, the probability of class membership is computed for each subject descriptor. The most probable *descriptor* is *assigned.* When a new article is classified with the Rocchio or χ^2 classifier, a scoring function computes the similarity between the feature vector of the new text to be classified and the weight vector of each class or category. We use the inner product of the vectors for computing this similarity (Jones & Furnas, 1987). The subject descriptor of the category weight vector with highest similarity is assigned to the new article. In a variant implementation, a second descriptor is assigned when the probability with the second best class or the similarity with the vector of this second best class is less than 10% lower than the probability or similarity of the best class.

The Bayesian and Rocchio classifiers are classical tools for pattern recognition in texts and are useful for comparisons. The Rocchio differs from the χ^2 classifier in the computation of the category weight vector.

The Rocchio algorithm developed for relevance feedback in information retrieval learns a better weight for each term of the query based upon the average weight of the term in the set of relevant and non-relevant texts. In text categorization the algorithm is used in a like manner. The weight of a feature (word or proper name) in a category weight vector is computed as the weighted difference of the mean weight of the feature in positive and negative training examples of the text class.

The χ^2 test computes how closely an observed probability distribution corresponds to an expected probability distribution. In our task, the observed probability distribution is formed by the observed frequencies of the number of texts relevant or non-relevant for the text class that contain the text feature (word or proper name) or

not contain the feature. A useful expected probability distribution is that all expected frequencies of the presence of the feature (or of the absence of the feature) will be equal in texts relevant for the class and texts non-relevant for the class. The hypothesis is tested whether the observed and the expected frequencies are close enough to conclude that they come from the same probability distribution (*goodness-of-fit test*). When the χ^2 variable of a term feature is low, the fit of the observed and expected frequencies is good and hence the feature has no influence upon the text class. When the value is high, there is an association between the feature and the class. In text categorization this association is used to select features that are highly related to the text class (Schütze, Hull, & Pedersen, 1995).

We use the χ^2 variable in a different way. The relationship of a feature with a class is computed by applying the χ^2 test. Instead of selecting features with a high χ^2 value, the raw χ^2 values are used in the category weight vector. The contingency table of relevant and non-relevant texts containing or not containing the text feature has 1 degree of freedom. Using the raw χ^2 values in similarity computation with the feature vector of a new text implicates that a term of the new text (word or proper name) that is related to the text class with a probability of 68% or more based on the training corpus has a positive effect upon class assignment (χ^2 value of 1 or more in the category weight vector used in the inner product). High χ^2 values of 9 or more indicate a probability of close to 100% that the term is related to the text class.

4. Results

We conducted a number of experiments aiming at comparing the initial feature selection methods and at comparing the different algorithms for text categorization.

The methods were tested upon a set of more than 930 new, previously unseen magazine articles. The effectiveness of automatic assignment of subject descriptors to the articles is obtained by comparing the results with descriptor assignment to these texts by the indexers of the publisher and by computing recall, precision, and F-measure values, which are common metrics for evaluating text categorization (Lewis, 1995). Text categorization is seen as a binary decision. An article belongs or does not belong to a specific class or category. *Recall* is the proportion of class members that the system assigns to the class. *Precision* is the proportion of members assigned to the class that really are class members. The *F-measure* combines recall and precision in one single measure. Recall, precision, and F-measure are ideally close to one.

A magazine article often contains many marginal *topics* that are of no interest when learning the class concepts or when assigning subject descriptors to new texts. Term weighting by considering the term frequency divided by the maximum frequency that a term occurs in the text and selecting terms with a high weight proved to be effective. The approach is useful to detect important topic terms and to undo the effect of long texts that are the result of verbosity. This form of term weighting proved also to be practical for identifying important proper names in the articles. Categorization results are usually better when feature selection is based on the term frequency with length normalization with an elimination of low term weights than when features are selected only from the begin section of the article. When content terms are selected from topically important segments of the article such as the heading, lead and attribution part, the set of terms still contains a set of noise terms. Discourse structure can be useful when selecting content terms. But, the segments, in which important content terms are located, may differ from one text class to another or from one text type to another. For instance, selecting content terms from the begin segments was advantageous for identifying features of the class MARKETING. In this class, the noisy terms of product descriptions usually appear further in the texts.

The χ^2 *classifier* scored much better than the other training methods and especially better than the Rocchio algorithm under the same circumstances. The classifier resulted in an average recall of 73%, average precision of 64%, and an average F-measure of 66%, when one or two descriptors were assigned (Table 1). When one descriptor was assigned, the classifier produced an average recall of 69%, average precision of 68% and an average F-measure of 66%. The percentages are the result of a initial feature selection by considering terms with a high frequency after length normalization. The average F-measure was 6% and 12% higher than the resulting F-measure when applying the Bayesian independence classifier and the Rocchio classifier respectively under the same circumstances (Table 2). The χ^2 classifier is less sensitive to noise terms. In the category weight vectors, noise terms have a very low weight compared to good cue terms. The χ^2 test that measures the fit between the observed and the expected frequencies of the content terms in the training corpus is effective for identifying

terms that are related to the class. We use the raw χ^2 values in the category weight vector. The benefit of this approach is explained and proved by our experiments. The χ^2 values strongly distinguish terms that are highly related to a class from the ones that are related to a lesser degree. Good results are obtained despite a limited number of positive training examples. A limited number of positive examples is common in routing tasks. At any time new topics may be introduced in the document stream.

Table 1. Results of categorization with the χ^2 algorithm. Features are selected by considering the term frequency with length normalization and elimination of terms with low weights. 1 or 2 descriptors are assigned.

Category	Recall	Precision	F-measure
CAR	0.827586	0.558140	0.666667
INVESTMENTS	0.789773	0.785311	0.787535
STOCK MARKET	0.481203	0.646465	0.551724
CULINARY	0.805970	0.613636	0.696774
FILM	1. 000000	0.576471	0.731343
COMPUTER SCIENCE	0.673913	0.620000	0.645833
INTERNATIONAL	0.444444	0.689655	0.540541
LITERATURE	0.909091	0.526316	0.666667
MARKETING	0.457895	0.906250	0.608392
MUSIC	0.93750	0.759494	0.839161
POLITICS	0.607143	0.377778	0.465753
SPORTS	0.888889	0.623377	0.732824
TOURISM	0.743590	0.707317	0.725000
REAL ESTATE	0.653846	0.500000	0.566667
Average	0.730060	0.635015	0.658920

Tabel 2. Average values of recall, precision, and F-measure when assigning 1 or 2 descriptors to the articles. Features are selected by considering the term frequency with length normalization and elimination of terms with low weights.

	Recall	Precision	F-measure
Bayesian independence	62%	62%	60%
Rocchio	63%	57%	54%
χ^2	73%	64%	66%

In appendix we include four examples of articles that were categorized with the χ^2 *classifier* (see below). Examples 1, 2, and 3 were correctly categorized by the system respectively as LITERATURE, REAL ESTATE, and FILM. The system correctly assigned the descriptor INTERNATIONAL in example 4, but ignored the second descriptor POLITICS. The texts of the examples are the original texts written in Dutch.

The Belgian publisher has recently implemented the techniques of selection of content terms and of training a χ^2 classifier as part of his document management system. Initial results with a partly different training and test corpus and partly different text classes (e.g., economy, taxes, etc.) show very good average F-measures and confirm the usefulness of our approach.

5. Conclusions

We can conclude the following. Successful systems that classify texts and assign subject or classification codes to texts rely upon the words and phrase patterns that signal the text class. In many text categorization situations their number is large to acquire manually. In this case, the classifier is trained upon example texts. Because the subject descriptors regard the broad text topics, an initial feature selection that identifies important content terms and proper names based upon the term frequency that is normalized by the maximum number a content term occurs in the text is effective. Adding knowledge of the discourse structure in the term selection process is useful for certain text classes. Given the limited number of positive examples and the high number of text features in the articles that belong to a variety of magazines, columns, and subject domains, the results of training a text classifier with the χ^2 algorithm are very satisfying.

References

Fuhr, N. (1989). Models for retrieval with probabilistic indexing. *Information Processing & Management*, 25 (1), 55-72.

Hayes, P.J. (1992). Intelligent high-volume text processing using shallow, domain-specific techniques. In P.S. Jacobs (Ed.), *Text-Based Intelligent Systems: Current Research and Practice in Information Extraction and Retrieval* (pp. 227-241). Hillsdale: Lawrence Erlbaum.

Jones, W.P., & Furnas, G.W. (1987). Pictures of relevance: a geometric analysis of similarity measures. *Journal of the American Society for Information Science*, 38 (6), 420-442.

Lewis, D.D. (1995). Evaluating and optimizing autonomous text classification systems. In E. A. Fox, P. Ingwersen, & R. Fidel (Eds.), *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 246-254). New York: ACM.

Lewis, D.D., Schapire, R.E., Callan, J.P., & Papka, R. (1996). Training algorithms for linear text classifiers. In H.-P. Frei, D. Harman, P. Schäuble, & R. Wilkinson (Eds.), *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 298-306). New York: ACM.

Maron, M. (1961). Automatic indexing: an experimental inquiry. *Journal of the ACM*, 8, 404-417.

Moens, M.-F. (in press). *Automatic Indexing and Abstracting of Document Texts* (270 pp.). Boston: Kluwer Academic Publishers.

Moens, M.-F., & Dumortier, J. (in press). Automatic categorization of magazine articles. *Information Processing & Management.*

Moens, M.-F., & Uyttendaele, C. (1997). Automatic structuring and categorization as a first step in summarizing legal cases. *Information Processing & Management*, 33(6), 727-737.

Rocchio, J. J. (1971). Relevance feedback in information retrieval. In G. Salton (Ed.), *The SMART retrieval system: Experiments in automatic document processing* (pp. 313-323). Englewood Cliffs, NJ: Prentice Hall.

Schütze, H., Hull, D. A., & Pedersen, J. O. (1995). A comparison of classifiers and document representations for the routing problem. In E. A. Fox, P. Ingwersen, & R. Fidel (Eds.), *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 229-237). New York: ACM.

APPENDIX

Example 1

HET GROTE PLAN

Als conservatieve factor kan het Ierse katholicisme tellen. Aanvaard je lot, is de algemene teneur, en wanneer je er onderdoor dreigt te gaan, kun je altijd troost vinden in de gedachte dat alle lijden in Gods grote plan een beter doel dient. Ook al ben je dan te kortzichtig om dat te snappen. In *Niall Williams'* debuutroman *"Vier liefdesbrieven"* spelen dit befaamde lot en de eruit resulterende schuld de hoofdrollen.

Nicholas Coughlan is twaalf wanneer zijn vader William van God de opdracht krijgt om zijn ambtenarenjob te laten voor wat hij is en kunstschilder te worden. Dat hij zijn gezin daarmee de armoede en zijn vrouw de dood injaagt, is voor William slechts een irrelevante opmerking. Wanneer Nicholas volwassen is, vecht hij een innerlijke strijd uit: zijn vader kan niet gelogen hebben over zijn roeping en er bestaat dus wel degelijk een groot goddelijk plan waarvan ook hij deel uitmaakt. Maar waarom blijft de openbaring zo lang uit?

In een tweede verhaallijn voert Williams ons naar een eiland buiten de Ierse westkust. Daar, in een bijna sprookjesachtig Keltisch aura van dichters en feeën, leven de kinderen Isabel en Sean Gore. Wanneer Sean tijdens het spelen verongelukt, neemt zus Isabel de morele schuld op haar schouders, waarna ze zichzelf als boetedoening opoffert in een hels huwelijk. Ook zij is dus getekend. De twee verhalen komen samen wanneer de zoektocht naar het laatste schilderij van zijn vader, Nicholas op het eiland brengt en het goddelijke plan, anders dan hij verwachtte, hem duidelijk wordt.

Williams heeft zijn boek in een licht magisch-realistische stijl geschreven, wat vooral in de eilandpassages goed aansluit bij de Ierse setting. Verwacht dus geen *Roddy Doyle* of *John Banville*, maar wel een hedendaagse prozaversie van *William Butler Yeats*.

Niall Williams, "Vier liefdesbrieven", Contact, Amsterdam, 336 blz., 995 fr.

Marnix Verplancke

Example 2

Lintbewinkeling

Hoe zijn baanwinkels in ons land gegroeid, wie zijn de eigenaars en welke producten worden er aangeboden? Een studie van makelaar Healey & Baker brengt de Belgische baanwinkelsituatie in kaart.

Dat het winkelaanbod in België op zijn zachtst gezegd 'versnipperd' is, mag voor een groot deel worden toegeschreven aan de baanwinkels, winkels die door hun omvang en activiteit de periferie van de steden opzoeken. Het bekendste voorbeeld is ongetwijfeld de 'Boomsesteenweg', op het grondgebied van Aartselaar bij Antwerpen.

Een studie van makelaar *Healey & Baker* schetst de geschiedenis van de baanwinkels in België en trekt enkele verrassende conclusies uit de samenstelling en de oppervlakte van het baanwinkelgebeuren.

De wildgroei in het winkelaanbod in perifere zones leidde in 1975 tot de wet op de handelsvestigingen. Het land werd ingedeeld in zones. Buiten de centra mochten nog winkels worden gebouwd zonder socio-economische vergunning met een netto-oppervlakte van 1500 vierkante meter (bruto 3000 vierkante meter), binnen de kernen van 750 vierkante meter netto (1000 vierkante meter bruto). Voor grotere verkoopoppervlakten was een dergelijke vergunning wel verplicht. Maar de tijdrovende procedures om ze te verkrijgen (in totaal mag op een wachttijd worden gerekend van zes tot acht maanden) en de subjectieve criteria die worden gehanteerd, zetten winkelbouwers aan om achterpoortjes te zoeken. De oplossing bleek panden te bouwen met een netto-oppervlakte van iets minder dan de limiet (1499 vierkante meter buiten en 749 vierkante vierkante meter binnen de kernen).

De reactie van de overheid liet lang op zich wachten, maar kwam er toch in 1994 met een nieuwe wet op de handelsvestigingen. Die beperkte de netto-oppervlakte van winkels zonder socio-economische vergunning tot 1000 vierkante meter netto buiten en 400 vierkante meter netto binnen bewoningskernen.

De studie van Healey & Baker stelt de vraag hoezeer het baanwinkelaanbod versnipperd is, welke eigenaars dit marktsegment

beheersen en welke producten en diensten er vooral worden aangeboden. Van 'baanwinkelketens' spreekt de studie zodra een onderneming meer dan vijf vestigingen heeft in ons land (met uitzondering van *Ikea* met vier vestigingen, maar een belangrijke verkoopoppervlakte) en twee derde van de vestigingen zich in de periferie bevinden.

KLEDING EN SCHOENEN

101 ketens voldeden aan deze voorwaarde (voedingswinkels kwamen niet in aanmerking), goed voor 1975 vestigingen en 2.100.000 vierkante meter brutoverkoopoppervlakte. Een groot gedeelte ervan (39 van de 101 ketens om precies te zijn) heeft minder dan tien vestigingen. Slechts drie hebben er meer dan honderd. Van versnippering gesproken. Uitgedrukt in vierkante meter verkoopoppervlakte beschikken twee ketens over meer dan 100.000 vierkante meter (*Brico* en *Euroshoe*). De helft bezet minder dan 10.000 vierkante meter.

Opvallend is het overzicht van de winkelactiviteiten van de baanwinkels. Meer dan 600 (27% van de oppervlakte) van de 1975 vestigingen bieden schoenen en kleding aan. Nochtans zijn dat precies de activiteiten die men via de vestigingswet in de stadskernen wilde houden. De wet van 1975 bereikte met andere woorden niet het gewenste doel.

Van de 101 ketens zijn 63 Belgisch van nationaliteit (goed voor 1370 van de 1975 vestigingen). Verder tellen we 14 Franse (145 winkels) en 19 Nederlandse (411 winkels) ketens (en vijf 'andere' waaronder Ikea en*Toys 'R' Us*).

In tegenstelling tot de situatie in het buitenland - waar baanwinkels gegroepeerd liggen in daartoe bestemde zones - gaat het in België om veel kleine parken. Meer dan de helft van de baanwinkellocaties telt minder dan tien winkels. Slechts in vier gevallen groepeerden meer dan dertig winkels zich op eenzelfde plaats.

Het rendement van baanwinkels daalt intussen fors. Voor enkele jaren moest een dergelijke zaak nog een rendement bieden dat 3,5 tot 4% boven dat van een A1-winkel (in een winkelstraat) lag. Nu vragen investeerders nog slechts een premie (het verschil in rendement) van goed 1%. 62% van de baanwinkels behoort toe aan privé-investeerders, 14% is in handen van de ketens zelf, 18% is eigendom van ontwikkelaars en 6% van institutionelen (maar het aandeel van deze laatste stijgt snel). De gemiddelde huurprijzen voor baanwinkels stegen volgens de studie van Healey & Baker met 60% tussen 1987 en nu. En de experts van het huis verwachten een verdere opmars met nog eens 40% tegen het jaar 2001.

GW

Example 3

De dromen waren bitter

De Hollywoodstudio Warner Brothers is vijfenzeventig jaar oud. Een terugblik op het imperium van de ruziënde broers Warner.

Bij het kijken naar oude Hollywoodprenten begint het plezier al bij het logo van de producerende studio: de brullende leeuw van *MGM*, de berg met wolkenaureool van *Paramount*, de rondtollende wereldbol van *Universal*, de radiomast van *RKO*, de zwaailichten rond de monumentale *20th Century Fox* letterblokken, de dame met toorts van *Columbia Pictures*. Maar geen bedrijfssymbool is de filmfan liever dan het robuuste zilveren schild waarin de twee eerste letters van *Warner Brothers* gebeiteld lijken.

Dit vertrouwd embleem, tijdens de gouden jaren van Hollywood steevast begeleid door de bruisende fanfare van *Max Steiner*, roept meteen een wereld op van taaie films, liefst in zwart-wit, vol misdaad, vervaarlijk avontuur en maatschappijkritische ondertonen. "The Maltese Falcon" en "Casablanca" met *Humphrey Bogart*, "Jezebel" en "The Letter" met *Bette Davis*, "The Roaring Twenties" en "White Heat" met *James Cagney*, "Mildred Pierce" met *Joan Crawford* - om maar enkele van de honderden prenten te noemen.

Meer dan bij welke andere studio is de historiek van Warner Bros. ook het verhaal van een aantal kleur rijke pioniers: de vier gebroeders Warner, dankzij wie het begrip nepotisme onlosmakelijk met Hollywood verbonden is. *Jack L.* (1995-1981), de jongste en bekendste van de vier, superviseerde samen met de vroeggestorven *Sam* (1888-1927) de productie; Jack aan de westkust en Sam in New York. *Harry* (1882-1958) en *Albert* (1884-1967) bestuurden de financiële en commerciële afdeling. Het viertal stamde uit een kroostrijk gezin van Pools-joodse afkomst. Hoe zij Hollywood veroverden, lijkt wel een filmscenario over de *American Dream*.

Vader *Benjamin* had in de vroege jaren 1880 zijn gezin achtergelaten in Kraznashiltz in Polen, om verwanten te volgen die in Amerika hun geluk gingen beproeven. Na twee jaar schoenen lappen in Baltimore had hij genoeg gespaard om zijn familie te laten overkomen, die in het beloofde land gestaag bleef groeien.

De vier broers probeerden aan de bak te komen in de meest diverse beroepen en handel (schoenen, slachterij, roomijs, kermis,

zeep, fietsen) en verzeilden per toeval in de nieuwe industrie die moeizaam uit de grond werd gestampt. Met hun spaarcenten kochten ze samen een kapotte filmprojector die ze oplapten en waarmee ze enige tijd (1905-1907) een bioscoop in Pennsylvania exploiteerden. Daarna stapten ze over op filmdistributie - een weinig gereglementeerde activiteit die nog in zijn kinderschoenen stond - en gingen vervolgens hun eigen films produceren. Hun eerste succes was minder een kwestie van talent dan van doorzettingsvermogen. In 1923 richtten ze hun eigen filmmaatschappij Warner Brothers op.

EEN PROVOCATEUR

De reputatie van hun studio werd gemaakt doordat Warner als eerste de stap naar de geluidsfilm waagde met "The Jazz Singer", een mijlpaal. De eerste experimenten met geluid waren vooral uitgevoerd door Sam, die echter de triomf van deze revolutionaire techniek zelf niet zou meemaken. De ironie van het lot wil dat hij in 1927 aan de vooravond van de première van "The Jazz Singer" overleed aan de gevolgen van een slecht verzorgd sinusabces. Voortaan had Jack de leiding over de filmproductie. Meer dan dertig jaar stond hij aan het hoofd van Warner Brothers en kwam daarbij voortdurend in conflict met zijn oudste broer. Jack en Harry, de twee pijlers van het bedrijf, konden elkaar niet uitstaan. Hun vijandschap was zo intens dat ze het vertikten om samen in het studiorestaurant te eten; tegen het eind van hun leven spraken ze gewoon niet meer tegen elkaar.

Jack, een snelpratende provocateur, gedroeg zich als een gefrustreerdeBroadwaykomiek. Volgens getuigen deed hij niets liever dan met luide stem vulgaire moppen vertellen. Toen *Albert Einstein* de studio in Burbank bezocht, zou hij tegen de grondlegger van de relativiteitstheorie gezegd hebben: *"You know, I have a theory of relatives, too - don't hire them."* Jack was onbeschoft, vulgair, opzichtig gekleed en hield ervan de anderen in verlegenheid te brengen. Vooral dan zijn broer Harry, in alles zijn tegengestelde.

De sobere en conservatieve Harry maakte weinig indruk op zijn omgeving. Hij was een toegewijde echtgenoot en vader. Net als rivaliserende "moguls", *Adolph Zukor* van Paramount en *Louis B. Mayer* van MGM, was hij een strenge moralist. Van zijn vader, een devote jood, had hij geleerd raciale en religieuze verdraagzaamheid te propageren.

Harry wist hoe hij de bankiers in Wall Street moest aanpakken en dankzij geweldige kapitaalsinvesteringen kon hij Warner ombouwen tot de eerste grote geluidsstudio. In de jaren dertig nam WB ook honderden filmzalen over, samen met platenfirma's en radiostations. Ook financierde hij Broadwayshows. In volle economische crisis had alleen MGM evenveel troeven in handen, waardoor beide studio's het best de moeilijke tijden doorspartelden.

Terwijl de studio uitgroeide tot een van de best uitgeruste filmproductiefabrieken, met naast hangars voor de geluidsstudio's en administratiegebouwen ook een aantal permanente decors (het archetypische westernstadje, straten in New York), werden de Warners ook getroffen door persoonlijke tragedies en verdeeldheid. Tijdens een bezoek aan Cuba liep Harry's 22-jarige zoon *Lewis* een fatale bloedvergiftiging op. De studio was daarmee een monarchie zonder kroonprins. Na de dood van Lewis groeide de tweedracht tussen Harry en Jack. Die werd nog op de spits gedreven toen Jack verliefd werd op *would-be* actrice *Ann Page Alvarado* en met zijn maîtresse ging samenwonen nog voor hij wettelijk was gescheiden. Nu ook hun vader Benjamin was overleden, zag Harry het als zijn heilige taak om de eendracht binnen de familie te bewaren. *"As long as you stand together, you will be strong"*, had de oude Benjamin nog gewaarschuwd.

BOERENKINKELS

Zelfs naar Hollywoodnormen werden de Warner broers als onderontwikkelde boerenkinkels beschouwd. Geen van de Warners was gecultiveerd. Ze lazen bijvoorbeeld nooit een boek, zelfs niet een gevierde roman die in aanmerking kwam om door hun studio te worden verfilmd. Toen regisseur *Mervyn LeRoy* tijdens zijn wittebroodsweken (na zijn huwelijk met Harry's dochter) merkte dat iedereen "Anthony Adverse" las, stuurde hij Jack een telegram met de aanmaning om het boek te lezen. *"Read it?"* telegrafeerde Jack terug, *"I can't even lift it."* En toch was het inzicht van deze cultuurbarbaar, alsook zijn instinctief aanvoelen van wat het publiek bezighoudt, van kapitaal belang bij het leiden van dit strak georganiseerde productiesysteem.

In zijn grondig gedocumenteerde studie over de joodse gemeenschap in Hollywood, *"An Empire of Their Own - How the Jews Invented Hollywood"*, beschrijft *Neal Gabler* een werkdag van Jack in de jaren dertig. Elke ochtend stond hij om 9 uur op, greep naar de telefoon om met zijn productiemanager zijn dagtaak te bespreken. Daarna nam hij met zijn assistente de post en de Hollywoodvakbladen door, waarin de passages die hem aanbelangden al waren aangestreept en samengevat. Bij het ontbijt las hij, met het oog op een mogelijke verfilming, synopsissen van scenario's en boeken. Daarna douchte hij. Gewoonlijk arriveerde hij pas rond de middag op het studioterrein waar hij nogmaals checkte met de productiemanager en occasioneel ook met de juridische dienst aangaande een deal voor een ster of een boek.

Rond één uur dertig ging hij lunchen in de directie-eetzaal waar hij een Zwitserse chef en een Duitse hoofdkelner had aangesteld. Tijdens de lunch praatte hij alleen maar over koetjes en kalfjes - gewoonlijk roddels en tips voor de paardenrace, zijn grote passie. Na de lunch ging Jack Warner in een van de filmzaaltjes de nog niet gemonteerde opnamen van de dag bekijken - *dailies* zoals ze in het jargon worden genoemd. Dit nam het grootste deel van de namiddag in beslag, twee tot drie uur. Terug in zijn kantoor ontving hij bezoekers en wisselde hij informatie uit met zijn productiechef, wiens bureau aan het zijne paalde.

Daarna was het tijd voor zijn dagelijkse scheerbeurt bij de studiobarbier, gevolgd door een bezoekje aan de studiosauna, waarna hij met hernieuwde krachten aan nog meer vergaderingen en conferenties begon. Toen de avond viel, ging hij nog niet naar huis, maar woonde hij samen met de andere top executives *previews* bij van nieuwe Warner-films, gewoonlijk in de buitenwijken van

Los Angeles, soms meer dan een uur rijden, maar altijd op een "geheime" locatie. Af en toe nam hij zijn dochtertje *Barbara* mee (*"Climbing into those black cars, we were like gangsters going to rob a bank"*, herinnert ze zich).

Tijdens de *screening* zat Jack altijd naast de *cutter*, aan wie hij zijn instructies doorgaf voor het aanbrengen van wijzigingen bij de montage. Wat daarbij opviel, was zijn feilloos geheugen voor *dailies* die hij drie tot vier maanden tevoren had gezien. Hij had altijd het laatste woord. Na die proefvertoningen keerde hij laat terug naar huis.

Alle studiobazen volgden in grote lijnen dezelfde routine. Hun leven speelde zich af in en rond de studio die hen in staat stelde een compleet fictieve wereld te scheppen waarover zij heer en meester waren.

AGRESSIEF EN GEDURFD

Tijdens de hoogdagen van het Amerikaanse studiosysteem had elke *major* zijn persoonlijke stijl - iets wat door heel wat factoren werd bepaald. Maar vooral de eigenaar van de studio drukte zijn stempel op de pellicule. Zo straalde de persoonlijkheid van Jack af op de producten die de Warner fabriek uitrolden, hoe groot ook de inbreng mag geweest zijn van de twee uitzonderlijk begaafde productieleiders die de studio inhuurde (*Darryl Zanuck* van 1929 tot 1933, *Hal Wallis* van 1933 tot 1944). Zoals zijn baas, was WB niet meteen een studio die je associeerde met klasse en prestige: het was integendeel de meest agressieve, realistische en gedurfde studio. MGM zweerde bij glamour en luxe, waardoor alle films er altijd onecht uitzagen; gepolijste musicals waren het paradepaardje van de studio. Paramount, waar voornamelijk emigranten werkten, was gespecialiseerd in gesofistikeerd, "continentaal" amusement. Bij Columbia heerste dankzij de populistische fabels van huisregisseur *Frank Capra* een spirit van sociaal optimisme.

Warner Brothers stond bekend voor zijn schraapzucht. Voor hij naar huis ging, liep Harry nog even langs de toiletten om alle lichten te doven. De studio zat voortdurend in de schulden en Harry snoeide dan altijd drastisch in de budgetten. *"Luister eens, een film is niet meer dan een dure droom"*, zei hij tegen een journalist. *"Het is even makkelijk om voor 700.000 dollar te dromen dan voor 1.500.000 dollar."*

Terwijl er bij MGM met geld werd gesmeten, werd bij Warner op alle mogelijke manieren bespaard. Die zuinigheid ging uiteindelijk de stijl van de films bepalen. De Warner-films waren bot, taai en snelden in grote vaart over het doek. Een stijl die perfect paste bij de actuele onderwerpen die de taal spraken van de grote stad, een wreed, onverschillig en antagonistisch *environment*, vaak in zware schaduwen gehuld. De verhalen waren energiek en kordaat verteld, maar werden overheerst door gevoelens van woede, bitterheid en doem. Denk maar aan klassiekers als "I am a Fugitive From a Chain Gang", "Wild Boys of the Road" en "High Sierra". De typische Warner-musical "Forty-Second Street" vertelde het vreugdeloos verhaal van een gevallen impresario, die een laatste wanhopige gok doet naar het succes.

MGM moest het van zijn sterren hebben (*"More stars than in heaven!"* pochte de studio). Warner beschikte ook over zijn sterrencontingent, alleen beantwoordden hun acteurs niet aan het traditionele Hollywoodideaal van glamour en zeemzoete romantiek. De mannen - Bogey, Cagney, *Edward G.Robinson*, *Paul Muni*, *John Garfield* - waren ruw, ongeschoren en klein van gestalte; de dames - Bette Davis, *Joan Blondell* - waren hard, leep en niet op hun mondje gevallen. Zelfs de cartoonkarakters, zoals *Bugs Bunny*, waren cynisch, bliksemsnel en onsentimenteel.

ONTEVREDENHEID EN MUITERIJ

Warner projecteerde in zijn sterren een ideaalbeeld van zichzelf: de rebel die van leer trekt tegen het establishment waartoe hij nooit zou behoren.

Jack Warner stond bekend als een keihard zakenman en een slavendrijver met zijn personeel. *"Het lijkt wel alsof de Warner-bazen hun acteurs verwarren met hun renpaarden"*, sneerde Cagney. Niet alleen de acteurs werden afgebeuld, ook voor de regisseurs was het arbeidsritme hels: *Michael Curtiz*, het werkpaard van de studio, regisseerde in de jaren dertig niet minder dan 44 films.

Precies omdat de studio teerde op koppige, compromisloze individuen, heerste er grote ontevredenheid en hing er voortdurend muiterij in de lucht. De bekendste sterren kwamen in opstand tegen hun langetermijncontracten, zowel Bette Davis als *Olivia de Havilland* sleepten hun werkgever voor de rechter en verloren het proces.

Volgens Gabler liet die vijandigheid ook zijn sporen na op de films, die vaak een opstandig en anti-autoritair toontje bezaten. Zelfs in de piratenprenten met huisster *Errol Flynn* zat klassenhaat verwerkt. Alle verworpenen van de Grote Depressie defileerden op het doek: werklozen, beroepsboksers, vuistvechters, vleesverwerkers, mijnwerkers, broodkaarters, oplichters en detectives. De studio werd dan ook het favoriete doelwit van zedenprekers die beweerden dat de studio niet alleen anti-sociaal gedrag afbeeldde, maar ook vergoelijkte.

De kritiek van moraalridders ten spijt, konden de Warner-films zeker niet antisociaal genoemd worden, alleen toonden ze een veel grotere ambivalentie jegens de traditionele Amerikaanse waarden dan de concurrerende productie.

Niet dat de Warner-broers zelf racidalen waren. Zoals alle andere studiobazen stemden ze republikeins. Behalve in 1932, toen machtige industriëlen en lobbyisten de hulp van de Warners inriepen om voor *Franklin Roosevelt* de kandidatuur en het

presidentschap te winnen. Eens aan de macht, ontbood Roosevelt, Jack met het verzoek of hij een film wilde maken waarin de Russen - toen bondgenoten van de Amerikanen - in een goed daglicht werden gesteld. Het resultaat, "Mission to Moscow" (1943), zou Jack zuur opbreken tijdens de daaropvolgende heksenjacht op communisten.

Toen het *House Un-American Activities Committee* (HUAC) op het einde van de jaren veertig naar Hollywood trok om te onderzoeken of de filmindustrie zich bezondigde aan communistische propaganda, was de rabiate anticommunist Jack Warner er als de kippen bij om namen te noemen van verdachte radicalen. Hij en zijn broers zouden maar al te graag alle communisten naar Rusland verbannen en geld inzamelen voor een pesticide om dit ongedierte uit te roeien. Zijn hatelijke tirade was deels ook een reactie tegen de brutale staking die de studio's had lamgelegd.

VADER EN ZOON

Het einde van het oude Hollywood liep parallel met het definitieve uiteenvallen van de Warner-dynastie. Jack raakte vervreemd van zijn zoon *Jack Jr.*, die zelfs verbannen werd naar het kantoor in Londen. Het was opvallend en pijnlijk in welke mate de relatie tussen Jack en zijn broer Harry, zijn vaderfiguur, later zou worden weerspiegeld in de relatie tussen Jack en zijn zoon. Vanaf de late jaren veertig werd het voortbestaan van het oude studiosysteem langs alle kanten bedreigd: door het toepassen van de antitrustwetten (studio's zagen zich genoodzaakt hun bioscoopketens af te stoten), de competitie van de commerciële televisie en de opkomst van de onafhankelijke productie. Geleidelijk geraakten de studio's opgezogen in de nieuwe Hollywoodconglomeraten. In de jaren vijftig verkochten de broers hun aandelen in Warner Brothers aan de *First National Bank* van Boston.

Achter de rug van zijn broer Harry, sloot Jack echter nog een deal om zijn titel te behouden. *"Dit verraad werd Harry fataal"*, zou diens schoonzoon later zeggen. Als in een Warner Bros-melodrama kreeg Harry kort na de verkoop een beroerte waarvan hij nooit volledig herstelde. Toen hij twee jaar later in 1958 stierf, keerde Jack voor de begrafenis niet eens terug uit Frankrijk. Zelf gereduceerd tot een levend anachronisme, verkocht de laatste oorspronkelijke oprichter van Warner Brothers in 1966 voor 32 miljoen dollar zijn eigen aandelen aan de holdingmaatschappij *Seven Arts Limited*. Na een mislukte fin de carrière als Broadwayproducer hield hij zich onledig met gokken en tennis. Tijdens een van die partijtjes tennis maakte hij een kwalijke val waarvan hij nooit helemaal herstelde. Vier jaar later, in 1978, stierf Jack net als Harry aan een beroerte. Zijn zoon die hij geweigerd had aan zijn ziekbed, werd gedoogd op de begrafenis. Het leven van de broers Warner werd nooit verfilmd, wat even jammer als onbegrijpelijk is.

Patrick Duynslaegher

Example 4

VINGER OP DE WOND

België reageert "ontzet" op het wapenincident in Congo, maar geeft hierdoor feitelijk (nogmaals) te kennen dat onze ambassade in Kinshasa geen terreinkennis heeft. Niemand betwist het ridicule van de Congolese aantijgingen, noch dat het gaat om een groteske manipulatie door anti-Belgische fracties in de *Kabila*-regering. Wie Congo/Zaïre (of een ander ontwikkelingsland) kent, geeft echter geen aanleiding tot dit soort voorspelbare incidenten. De ambassade trapte nodeloos in een val - de Belgische bedrijven in Congo dreigen eens te meer de rekening te betalen.

Buitenlandse Zaken zal dit tegenspreken, maar bevestigt hiermee onze stelling. Het is immers een kwestie van elementaire basiskennis dat Belgische diplomaten niet met een krat wapens door Congo zeulen, indien zo'n transport niet wordt gewaarborgd door een vrijbrief van de president in hoogsteigen persoon (het kleinste *expat*-kind in Congo weet dat het niet met een waterpistool over straat loopt).

Het handvol wapens waarover het gaat, waren al maanden in Lubumbashi - er was niet de minste hoogdringendheid om deze hoognodig naar België te repatriëren. Vooral niet in een sfeer van onderkoelde relaties. Buitenlandse Zaken verwijst zelf naar een toenemende spanning, onder meer na de binnenlandse verbanning van de vroegere held van de Belgische media, oppositieleider *Etienne Tshisekedi*, en de opheffing door het Kabila-bewind van de mensenrechtenorganisatie *Azadho*. Ook in dit geval reikte België de stok aan waarmee het geslagen wordt, want kan men zich voorstellen (en dulden) dat Congo de *Witte Comités* in België zou financieren?

Dat Brussel dit soort gevoeligheden niet onderkent, geeft aan dat het niet over de diplomatieke bekwaamheden beschikt om op een constructieve manier met moeilijke regimes om te gaan. In die omstandigheden is zelfs "stille diplomatie" tot mislukken gedoemd.

MESH: an Object-Oriented Approach to Hypermedia Modeling and Navigation

Dr. Wilfried Lemahieu

Author information:

Dr. Wilfried Lemahieu

Department of Applied Economic Sciences
Katholieke Universiteit Leuven
Naamsestraat 69 B-3000 Leuven
Belgium
tel: + 32 16 32 68 86
fax: + 32 16 32 67 32
e-mail: wilfried.lemahieu@econ.kuleuven.ac.be

MESH: an Object-Oriented Approach to Hypermedia Modeling and Navigation

Dr. Wilfried Lemahieu

Abstract

This paper introduces the *MESH* approach to hypermedia modeling and navigation, which aims at relieving the typical drawbacks of poor maintainability and user disorientation. The framework builds upon two fundamental concepts. The *data model* combines established entity-relationship and object-oriented abstractions with proprietary concepts into a *formal hypermedia data model*. Uniform layout and link typing specifications can be attributed and inherited in a *static* node typing hierarchy, whereas both nodes and links can be submitted *dynamically* to multiple complementary classifications. In the *context-based navigation paradigm*, conventional navigation along static links is complemented by run-time generated *guided tours*, which are derived dynamically from the context of a user's information requirements. The result is a two-dimensional navigation paradigm, which reconciles complete navigational freedom and flexibility with a measure of linear guidance. These specifications are captured in a high-level, platform independent *implementation framework*.

1 Introduction

The hypermedia paradigm looks upon data as a network of *nodes*, interconnected by *links*. Whereas each node symbolizes a *concept*, a link not only stands for a *relation* between two items, but also explicitly assumes the semantics of a navigation path, hence the quintessential property of *navigational data access*. Their inherent flexibility and freedom of navigation raises hypermedia systems as utterly suitable to support user-driven exploration and learning. Therefore, hypermedia data retrieval embraces a notion of *location*. Data accessibility depends on a user's position in the network, denoted as the *current node* [Lucarella, 1990]. Manipulation of this position gradually reveals links to related information.

Unfortunately, due to inadequacy of the underlying data models, most hypermedia technologies suffer from severely limited maintainability. Moreover, the explorative, non-linear nature of hypermedia navigation imposes a heavy processing load upon the end user, referred to as *cognitive overhead* [Conklin, 1987]. The stringent problem of cognitive overhead effecting into user disorientation and losing one's chain of thought is known as the 'lost in hyperspace' phenomenon [Nielsen, 1990]; [Hammond, 1993]. Disorientation is further increased by the sense of *fragmentation* that is induced by scattering information over numerous separate nodes [Thüring et al., 1995].

This paper overviews the *MESH* hypermedia framework as deployed in [Lemahieu, 1999], which proposes a structured approach to both data modeling and navigation, so as to overcome said maintainability and user disorientation problems. *MESH* is an acronym for *Maintainable, End user friendly, Structured Hypermedia*. The text is partitioned according to *MESH*'s fundamental concepts. To start with, the *object-oriented hypermedia data model* is portrayed. The next section is dedicated to the *context-sensitive navigation paradigm*. A subsequent section translates these blueprints into a high-level *implementation framework*, specified in an abstract and platform independent manner. A last section makes comparisons to related work and formulates conclusions.

2 An object-oriented hypermedia data model

2.1 Introduction

The benefits of data modeling abstractions to both orientation and maintainability were already acknowledged in [Halasz, 1988]. They yield richer domain knowledge specifications and more expressive querying. Typed nodes and links offer increased consistency in both node layout and link structure [Thüring et al., 1991]; [Knopik & Bapat, 1994]. Higher-order information units and perceivable equivalencies (both on a conceptual and a layout level) greatly improve orientation [Thüring et al., 1995]; [Ginige et al., 1995]. Semantic constraints and consistency can be enforced [Garzotto et al., 1995]; [Ashman et al., 1997], tool-based development is facilitated and reuse is encouraged [Nanard & Nanard, 1995].

The first conceptual hypermedia modeling approaches such as *HDM* [Garzotto et al., 1993] and *RMM* [Isakowitz et al., 1995]; [Isakowitz et al., 1998] were based on the entity-relationship paradigm. Object-oriented techniques were mainly applied in *hypermedia engines*, to model functional behavior of an application's *components*. Along with the *Tower model* [De Bra et al., 1992], *EORM* [Lange, 1994] and *OOHDM* [Schwabe et al., 1996]; [Schwabe & Rossi, 1998a]; [Schwabe & Rossi, 1998b], *MESH* is the first approach where modeling of the *application domain* is fully accomplished through the object-oriented paradigm. Its data model is based on concepts and experiences in the related field of database modeling, taking into account the particularities inherent to the hypermedia approach to data storage and retrieval. Established object-oriented modeling abstractions [Rumbaugh et al., 1991]; [Jacobson et al., 1992]; [Meyer, 1997]; [Snoeck et al., 1999] are coupled to proprietary concepts to provide for a *formal hypermedia data model*.

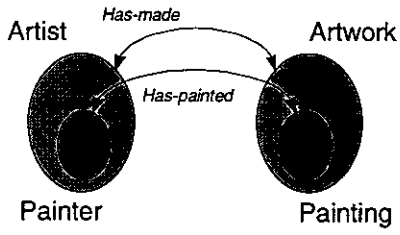
2.2 The basic concepts: node types, layout templates and link types

On a conceptual level, a *node* is considered a black box, which communicates with the outside world by means of its *links*. External references are always made to the node *as a whole*. True to the O.O. *information-hiding* concept, no direct calls can be made to its multimedia content. However, internally, a node may encode the intelligence to adapt its visualization to the *navigation context*, as discussed in section 4. Nodes are assorted in an inheritance hierarchy of *node types*. Each child node type should be compliant with its parent's definition, but may fine-tune inherited features and add new ones. These features comprise both node layout and node interrelations, abstracted in *layout templates* and *link types* respectively.

A *layout template* is associated with each level in the node typing hierarchy, every template being a refinement of its predecessor. Its exact specifications depend upon the implementation environment, e.g. as to the Web it may be *HTML* or *XML* based. Node typing as a basis for layout design allows for uniform behavior, onscreen appearance and link anchors for nodes representing similar real world objects.

A *link* represents a one-to-one association between two nodes, with both a semantic and a navigational connotation. A directed link offers an access path from its *source* to its *destination node*. Links representing similar semantic relationships are assembled into *types*. Link types are attributed to node types and can be inherited and refined throughout the hierarchy. Link type properties such as *domain*, *cardinalities* and *destination/inverse*[1] allow for enforcing constraints upon their instances. These properties can be overridden to provide for stronger restrictions upon inheritance. E.g. whereas an **artist** node can be linked to any **artwork** through a ***has-made*** link type, an instance of the child node type **painter** can only be linked to a **painting**, by means of the more specific child link type ***has-painted***.

[1] As discussed in detail in [Lemahieu, 1999], a link type's *destination* is a derived property, defined as the *inverse link type's domain*.



2.3 The use of aspect descriptors and aspect types to overcome limitations of a rigid node typing structure

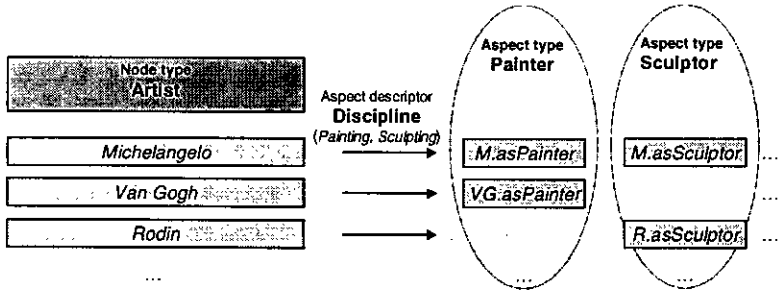
The above model is based on a node typing strategy where node classification is total, disjoint and constant. The aspect construct allows for defining *additional* classification criteria, which are not necessarily subject to these restrictions. Apart from a single "most specific node type", they allow a node to take part in other secondary classifications that are permitted to change over time.

An *aspect descriptor* is defined as an attribute whose (discrete) values classify nodes of a given type into respective additional subclasses. In contrast to a node's "main" subtyping criterion, such aspect descriptor should not necessarily be *single-valued* nor *constant over time*. Aspect descriptor properties denote whether the classification is *optional/mandatory*, *overlapping/disjoint* and *temporary/ permanent*.

Each *aspect type* is associated with a single value of an aspect descriptor. An aspect type defines the properties that are attributed to the class of nodes that carry the corresponding aspect descriptor value. An aspect type's instances, *aspects*, implement these type-level specifications. Each aspect is inextricably associated with a single node, adding characteristics that describe a specific "aspect" of that node.

A node instance may carry multiple aspects and can be described by as many aspect descriptors as there are additional classifications for its node type. If multiple classifications exist, each aspect descriptor has as many values as there are subclasses to the corresponding specialization. Its cardinalities determine whether the classification is total and/or disjoint. As opposed to node types, aspects are allowed to be volatile. Hence, dynamic classification can be accomplished by manipulating aspect descriptor values, thus adding or removing aspects at run-time. Aspect types attribute the same properties as nodes: *link types* and *layout*. However, their instances differ from nodes in that they are not directly referable. An aspect represents the *same real-world object* as its associated node and can only be visualized as a subordinate of the latter.

E.g. to model an **artist** that can be skilled in multiple disciplines, a non-disjoint aspect descriptor *discipline* defines the **painter** and **sculptor** aspect types. Discipline-specific node properties are modeled in these aspect types, such that e.g. the **Michelangelo** node features the combined properties of its **Michelangelo.asPainter** and **Michelangelo.asSculptor** aspects.



Node type properties (i.e. layout and link types) can be *delegated* to aspect descriptors, such that they can be inherited and overridden in each aspect type that is associated with one of the descriptor's values. The inheritance/overriding mechanism is similar to the mechanism for supertypes/subtypes,

but because an aspect descriptor can be multi-valued, particular care was taken so as to preclude any inconsistencies[1].

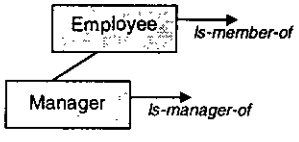
2.4 Link typing and subtyping

In common data modeling literature, subtyping is invariably applied to *objects*, never to *object interrelations*. If additional classification of a relationship type is called for, it is *instantiated* to become an object type, which can of course be the subject of specialization. However, as for a hypermedia environment, node types and link types are two separate components of the data model with very different purposes. It would not be useful to instantiate a link type into a node type, since such nodes would have *no content* to go along with them and thus each instance would become an 'empty' stop during navigation. This section demonstrates how specialization semantics can be enforced not only upon node types, but also upon the *link types*. A sub link type will model a type whose set of instances constitutes a subset of its parent's, and which models a relation that is more specific than the one modeled by the parent.

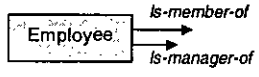
A link instance is defined as a source node - destination node tuple (n_s, n_d). Tuples for which this association represents a similar semantic meaning are grouped into link types. A link type defines instances that comply with the properties of the type and is constrained by its *domain*, its *cardinalities* and its *inverse link type*. The *domain* of the link type is the data type to which the link type is attributed. This can be either a node type or an aspect type.

If L_c is a sub link type resulting from a specialization over L_p, the set of (n_s, n_d) tuples defined by L_c is a subset of the one defined by L_p. Such specialization is called *vertical* if it is the consequence of a parallel classification over the link types' *domain*, denoting that the sub link type is attributed at a 'lower', more specific level in the node typing hierarchy than its parent. If L_c and L_p share the same domain, L_c can still define a subtype of L_p in the case where L_c models a more restricted, more specific kind of relationship than L_p, independently of any node specialization. Both parent and child link type are attributed at the same level in the node type hierarchy, hence the term *horizontal* specialization.

E.g.



Vertical link specialization



Horizontal link specialization

Apart from the domain, a link type's cardinalities and inverse can be overridden as well upon specialization. *MESH* presents a formal overriding mechanism, wherein particular care is taken so as not to violate the parent's constraints, particularly in case of a non-disjoint classification.

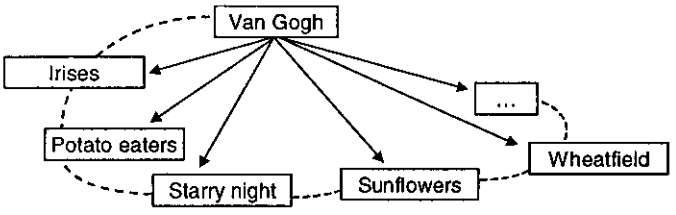
3 The context-based navigation paradigm

3.1 Linearity and guided tours

To highlight the advantages of hypermedia navigation, comparisons are often made to books. Books are said to be *linear* information systems; their pages are organized uni-dimensionally, in a fixed order. Hypertext offers the possibility to break through this linear constraint and organize data in more complex structures, to be accessed following different possible paths, depending on the user's preferences and interests. Cognitive overhead, however, is significantly lower in a linear structure, be it at the cost of navigational freedom. Linearity provides a leading thread that facilitates orientation

[1] We deliberately opted for a single inheritance structure, however, aspects can provide an elegant solution in many situations that would otherwise call for multiple inheritance, much like *interfaces* in the *Java* language. See [Lemahieu, 1999] for further details.

and prevents the reader from getting lost [Jonassen, 1990]. The latter is acknowledged in [Trigg, 1988]; [Nielsen, 1990], where linearity is re-introduced in so-called *guided tours*, chaining together all nodes pertaining to a common subject with *forward/backward* links. E.g. the typical hypermedia links (represented as arrows) between **Van Gogh** and each of his **paintings** can be complemented by a *guided tour* (represented as dotted lines) along these **paintings**.



Unfortunately, such hard-coded guided tours have proven to be inflexible and difficult to maintain. Moreover, they introduce a measure of redundancy into the hyperbase, as a guided tour typically reflects a communal property among its participating nodes. However, the property of '*being painted by the same artist*' is already established within the respective links from each **painting** to its **artist**. Thus, it would be possible to infer this knowledge and generate such guided tour *at run-time*, without burdening hyperbase maintainability.

MESH builds upon its data model to reconcile navigational freedom with the ease of linear navigation. Its intended navigation mechanism is that of an "intelligent book", which is to provide a disoriented end user with a sequential path as a guidance. Such guided tour is not static, but is adapted dynamically to the *navigation context*. In addition, a node is able to tune its *visualization* to the context in which it is accessed, hence providing the user with the most relevant subset of its embedded multimedia objects.

3.2 A guided tour as derived from the current context

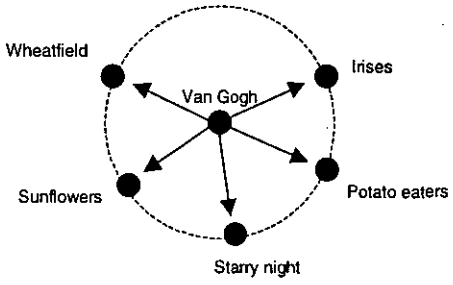
In conventional hypermedia applications, the *current node* is the only variable that determines which information is accessible at a given moment; navigation is only possible to nodes that are linked to this current node. Its value changes with each navigation step as it represents the immediate focus of the user's attention. *MESH* introduces the *current context* as a second, longer-term variable that 'glues' the various visited nodes together and provides a background about which common theme is being explored. The current context is defined as the combination of a *context node* and a *context link type*. The context node represents the subject around which the user's broader information requirements 'circle'. The nature of the relationship involved is depicted by the context link type.

A guided tour derives from the current context. Therefore, *MESH* discriminates between *direct* and *indirect* links. A direct link represents a lasting relation between two nodes. Direct links are typed and reflect the underlying conceptual data model. Because they are permanent and context-independent, they are stored explicitly into the hyperbase and are always valid. E.g. the node **Sunflowers** is directly linked to the **Van Gogh** node.

An *indirect* link between two nodes indicates that they share relevancy to a common third node. The latter denotes the *context* within which the indirect link is valid. As indirect links not only reflect the data model, but also depend on a run-time variable, the *current context*, they cannot be stored within the hyperbase. They are to be created *dynamically* at run-time, as inferred from a particular context. E.g. an indirect link between **Sunflowers** and **Wheatfield** is only relevant when exploring information related to **Van Gogh**.

A *guided tour* is defined as a path of *indirect* links along all nodes relevant to the current context. These nodes are directly linked to the context node (through instances of the context link type) and indirectly to their predecessor and successor in the tour. As they are chained into a linear structure, a logical order should be devised in which the subsequent tour nodes can be presented to the user. The

most obvious criterion is in alphabetical order of a *node descriptor* field. More powerful alternatives are discussed in [Lemahieu, 1999]. E.g. the context **Van Gogh.*has-painted*** yields a guided tour among the nodes {**Irises, Potato eaters, Starry night, Sunflowers, Wheatfield,** ...} with **Van Gogh** as the *context node* and ***has-painted*** as the *context link type*.

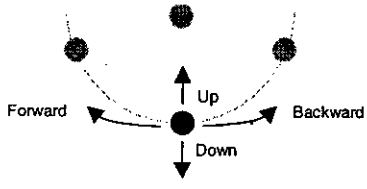


Current context: **Van Gogh.*has-painted***

Note that the discrepancy between *guided tour* and *context* can be compared to the traditional duality in representing a circle either through the points on its *circumference*, or through its *center* and a *radius*. Guided tours are not stored within the hyperbase as *an enumeration of participating nodes*, but are calculated at run-time from the *current context*. Although sequential by nature, such tours do not restrict the user's navigational freedom, as long as sufficient flexibility is offered in choosing which tour to follow. The linearity lies in 'following' the tour. The freedom lies in starting one.

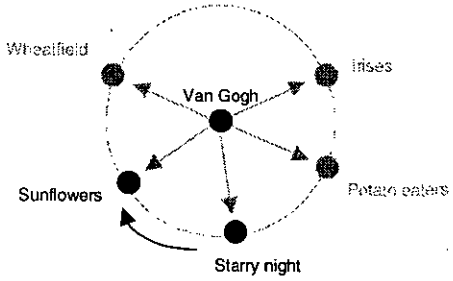
3.3 Navigational actions

Navigational actions can be classified according to two dimensions. First, there is moving *forward* and *backward* within the current tour, along indirect links. Second, and orthogonal to this, there is the option of moving *up* or *down* along direct links, closer to or further away from the session's starting point. Additionally, one can distinguish between actions that change the current context and actions that only influence the current node.



3.3.1 Moving forward/backward within the current tour

Moving forward or backward in a guided tour along indirect links, results in the node following/preceding the current node being accessed to become the new current node. The current context is unaffected.



Current context: **Van Gogh.*has-painted***
Current node: **Starry night ⇒ Sunflowers**

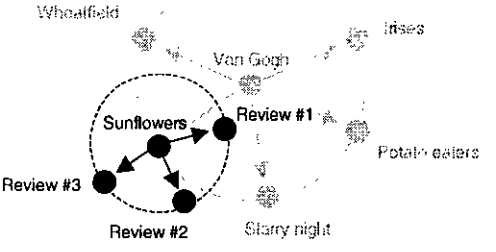
3.3.2 Moving up/down

Moving down implies an action of 'digging deeper' into the subject matter, moving away from the starting point. This is accomplished through selection from the current node of either a direct link type or link instance. In the case of a *unique* destination node, the result is the latter node being accessed. In the case of a *set* of destination nodes, the outcome is a new 'nested' tour being started.

In complete analogy to traditional hypermedia navigation, selection of a link instance l from a given source node n_s results in its unique destination node n_d being accessed: $n_s.l := \{n_d \mid l = (n_s, n_d)\}$. E.g. selection of the link (**Sunflowers, National Gallery**) from the current node **Sunflowers**, induces an access to the node **National Gallery**.

However, *MESH* aggregating single *link instances* into *link types*, yields the opportunity of anchoring and consequently selecting a *complete link type* from a given source node. Selection of a link type L from a source node n_s yields a set of all destination nodes n_d of tuples representing link instances of L with n_s as the source node, i.e. all nodes that are linked to the current node by the selected link type: $n_s.L := \{n_d \mid (n_s, n_d) \in L\}$. Depending on maximum cardinality of the link type, the resulting set may contain multiple destination nodes. E.g. with **Sunflowers** as the current node, selection of the link type ***reviews*** generates a *collection* of nodes to-be-accessed: **Sunflowers.*reviews*** := {**review#1, review#2, review#3**, ...}.

The result of such action is a *context change*: a new context emanates, resulting in new indirect links. A new tour is generated, nested within the former, according to this new context. The current node **Sunflowers** is denoted as the new context node. The non-unique link type ***reviews*** defines the context link type, which yields a new *nested* tour: **Sunflowers.*reviews***. The first review is accessed to become the new current node. Such *context change* reflects the user's decision to concentrate on the current node as a new topic of interest. All indirect links are destroyed and redefined around this new context.



Current context: **Van Gogh.*has-painted*⇒ Sunflowers.*reviews***
Current node: **Sunflowers ⇒ Review#1**

Hence contexts, and consequently guided tours, can exist in layers. As such, it is possible to 'delve' into a subject and have multiple *open* tours, nested within one another, where the context node of one tour is the current node of the tour it is nested in. Navigation along indirect links is invariably carried out within the "deepest", i.e. most recently started tour. Continuing a tour on a higher level is only possible if all tours on a lower level have been either completed or disbanded. This is accomplished by *moving up*, which reverses the latest *move down* action. If the latter involved a context change, the move up action results in the reestablishment of the previous context and the cancellation of the tour generated through this most recent link type selection. The previous context's context node and indirect links are restored. The most recent context node (**Sunflowers** in the example) again becomes the current node.

The practice of node and link typing allows for casting navigational actions to a whole *class* of nodes, regardless of the actual instance they are applied to. Hereby, selections of link *types* that exist at a sufficiently high level of abstraction can be imposed upon every single node belonging to a tour. E.g. in the context of **Van Gogh.*has-painted***, a **painting#x.*reviews*** selection can be issued once on *tour* level, with additional (nested) tours being generated automatically for each node participating in the **Van Gogh.*has-painted*** tour. If these tours in their turn include navigational actions on type level, a complex navigation pattern results, which can be several levels deep. Again, *forward* and *backward*

links always apply to the current tour, i.e. to the open tour at the 'deepest' level. In addition, the abstract navigational actions and tour definitions sustain the generation of very compact tree-shaped overviews and maps of complete navigation sessions[1]. In this respect, the *move up* and *move down* actions indeed correspond to moving up or down in the graph. The represented information can also be *bookmarked*, i.e. bookmarks not just refer to a single node but to a complete navigational situation, which can be resumed at a later time.

4 A generic application framework

The *information content* and *navigation structure* of the nodes are separated and stored independently. The resulting system consists of three types of components: the *nodes*, the *linkbase/repository* and the *hyperbase engine*. In [Lemahieu, 1999], a platform-independent implementation framework was provided, but all subsequent prototyping is explicitly targeted at a *Web* environment.

A node can be defined as a static page or a dynamic object, using e.g. *HTML* or *XML*. Its internal content is shielded from the outside world by the indirection of link types playing the role of a node's *interface*. Optionally, it can be endowed with the intelligence to tune its reaction to the *context* in which it is accessed by integrating the node type's set of attributed link types as a parameter in its layout template's presentation routines, hence the so-called *context-sensitive visualization* principle. This allows for different views to be defined over the same information, much like the *city* construct in the *Tower* model.

Since a node is not specified as a necessarily searchable object, linkage information cannot be embedded in a node's body. Links, as well as meta data about node types, link types, aspect descriptors and aspects are captured within a searchable *linkbase/repository* to provide the necessary information pertaining to the underlying hypermedia model, both at design time and at run-time. This repository is implemented in a relational database environment. Only here, references to physical node addresses are stored, these are never to be embedded in a node's body. All external references are to be made through location independent *node ID*'s.

The *hyperbase engine* is conceived as a server-side script that accepts link (type) selections from the current node, retrieves the correct destination node, keeps track of session information and provides facilities for generating maps and overviews. Since all relevant linkage and meta information is stored in the relational DBMS, the hyperbase engine can access this information by means of simple, pre-defined and parameterized *database queries*, i.e. without the need for searching through *node content*.

5 Evaluation and conclusions

5.1 Data modeling and authoring

Ohter hypermedia approaches such as *EORM*, *RMM*, *HDM* and *OOHDM* are also based on conceptual modeling abstractions, either through E.R. or O.O techniques. Among these, *OOHDM* is the only methodology to incorporate a subtyping and inheritance/overriding mechanism. However, subtyping modalities are not explicitly stipulated. Rather, they are borrowed from *OMT* [Rumbaugh et al., 1991], a general-purpose object-oriented design methodology.

MESH deploys a proprietary approach, specifically tailored to hypermedia modeling, where *structure* and *relationships* prevail over *behavior* as important modeling factors. Its full O.O. based data modeling paradigm should allow for hypermedia *maintenance* capabilities equaling their database counterpart; with unique object identifiers, monitoring of integrity, consistency and completeness checking, efficient querying and a clean separation between authoring *content* and *physical hyperbase*

[1] See [Lemahieu, 1999] for further details.

maintenance. *MESH* is the only approach to formulate specific rules for inheriting and overriding layout and link type properties, taking into account the added complexity of plural (possibly overlapping and/or temporal) node classifications. Links are treated as first-class objects, with link types being able to be subject to multiple specializations themselves, not necessarily in parallel with node subtyping. Authoring is greatly facilitated by O.O. features such as inheritance and overriding, class properties and layout templates that allow for high-level specification and lower-level refinement of node properties. Links can be anchored on *type* level, independently of actual node and link instances. Semantics attributed within the data model permit the automated type checking and integrity constraints we have grown accustomed to in a database environment. Dangling links and inconsistent link attributions can already be detected during the design phase. Finally, it is clear that a model-based approach in general facilitates information sharing, reuse, development in parallel, etc.

5.2 Navigation and orientation

Apart from the obvious benefit of a well-maintained hyperbase, *typed links* should permit a better comprehension of the semantic relations between information objects. The use of *higher-order* information units and the representation of collections of nodes as (source node, link type) combinations, induces a stronger sense of structure. A *node typing hierarchy* with consistent layout and user interface features, reflecting similarities between nodes, is to reduce both cognitive overhead and the impression of fragmentation. The *context* concept, as a representation of what various nodes have in common, will diminish fragmentation, but is to remedy cognitive overhead as well, as the linear guided tours improve a user's sense of position and his ability to ascertain his navigational options. Through the specification of navigational actions *on tour level*, complex navigation patterns can be applied to all nodes in a tour without additional effort. A further decrease in fragmentation and cognitive overhead is obtained by making node *visualization* dependent upon the context in which a node is accessed. The abundance of meta-information as node, aspect and link types allows for enriching *maps* and *overviews* with concepts of varying granularity. A final benefit is the ability to *bookmark a complete navigational situation* in an utterly compact manner, with the possibility of it being resumed later on, from the exact point where it was left.

EORM, *RMM*, *HDM* and *OOHDM* also feature specific topologies such as *guided tours*, *indexes* etc. A fundamental difference is that these are conceived as explicit *design components*, requiring author input for query definitions, node collections and forward/backward links. In *MESH*, neither guided tours nor indexes require any maintenance nor design effort, as the author is not even engaged in their realization. They are generated at run-time upon user request, not to restrict his freedom, but merely to facilitate navigation and support orientation.

References

[Ashman et al., 1997] H. Ashman, A. Garrido and H. Oinas-Kukkonen, Hand-made and Computed Links, Precomputed and Dynamic Links, *Proceedings of Hypertext - Information Retrieval - Multimedia (HIM '97)*, Dortmund (Sep. 1997)

[Conklin, 1987] J. Conklin, Hypertext: An Introduction and Survey, *IEEE Computer Vol. 20*, No. 9 (Sep. 1987)

[De Bra et al., 1992] P. De Bra, G. Houben and Y. Kornatzky, An Extensible Data Model for Hyperdocuments, *Proceedings of the fourth ACM European Conference on Hypermedia Technology (ECHT '92)*, Milan (Dec. 1992)

[Garzotto et al., 1993] F. Garzotto, P. Paolini, and D. Schwabe, HDM - A Model-Based Approach to Hypertext Application Design, *ACM Trans. Inf. Syst. Vol. 11*, No. 1 (Jan. 1993)

[Garzotto et al., 1995] F. Garzotto, L. Mainetti and P. Paolini, Hypermedia Design, Analysis, and Evaluation Issues, *Commun. ACM Vol. 38*, No. 8 (Aug. 1995)

[Ginige et al., 1995] A. Ginige, D. Lowe and J. Robertson, Hypermedia Authoring, *IEEE Multimedia Vol. 2*, No. 4 (Winter 1995)

[Halasz, 1988] F. Halasz, Reflections on NoteCards: Seven Issues for Next Generation Hypermedia Systems, *Commun. ACM Vol. 31*, No. 7 (Jul. 1988)

[Hammond, 1993] N. Hammond, Learning with Hypertext: Problems, principles and Prospects, *HYPERTEXT a psychological perspective*, C. McKnight, A. Dillon and J. Richardson Eds., *Ellis Horwood*, New York (1993)

[Isakowitz et al., 1995] T. Isakowitz, E. Stohr and P. Balasubramanian, RMM, A methodology for structured hypermedia design, *Commun. ACM Vol. 38*, No. 8 (Aug. 1995)

[Isakowitz et al., 1998] T. Isakowitz, A. Kamis and M. Koufaris, The Extended RMM Methodology for Web Publishing, *Working Paper IS-98-18, Center for Research on Information Systems*, 1998 (Currently under review at ACM Trans. Inf. Syst.)

[Jacobson et al., 1992] I. Jacobson, M. Christerson, P. Jonsson and G. Övergaard, Object-Oriented Software Engineering, *Addison-Wesley*, New York (1992)

[Jonassen, 1990] D. Jonassen, Semantic net elicitation: tools for structuring hypertext, *Hypertext: State of the Art*, R. McAleese and C. Green Eds., *Intellect*, Oxford (1990)

[Knopik & Bapat, 1994] T. Knopik and A. Bapat, The Role of Node and Link Types in Open Hypermedia Systems, *Proceedings of the sixth ACM European Conference on Hypermedia Technology (ECHT '94)*, Edinburgh (Sep. 1994)

[Lange, 1994] D. Lange, An Object-Oriented design method for hypermedia information systems, *Proceedings of the twenty-seventh Hawaii International Conference on System Sciences (HICSS-27)*, Hawaii (Jan. 1994)

[Lemahieu, 1999] W. Lemahieu, Improved Navigation and Maintenance through an Object-Oriented Approach to Hypermedia Modelling, *Doctoral dissertation (unpublished)*, Leuven (Jul. 1999)

[Lucarella, 1990] D. Lucarella, A Model For Hypertext-Based Information Retrieval, *Proceedings of the European Conference on Hypertext*, Versailles (Nov. 1990)

[Meyer, 1997] B. Meyer, Object-Oriented Software Construction, Second Edition, *Prentice Hall Professional Technical Reference*, Santa Barbara (1997)

[Nanard & Nanard, 1995] J. Nanard and M. Nanard, Hypertext Design Environments and the Hypertext Design Process, *Commun. ACM Vol. 38*, No. 8 (Aug. 1995)

[Nielsen, 1990] J. Nielsen, The Art of Navigating Through Hypertext, *Commun. ACM Vol. 33*, No. 3 (Mar. 1990)

[Rumbaugh et al., 1991] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy and W. Lorensen, Object Oriented Modelling and Design, *Prentice Hall*, Englewood Cliffs (1991)

[Schwabe et al., 1996] D. Schwabe, G. Rossi and S. Barbosa, Systematic Hypermedia Application Design with OOHDM, *Proceedings of the seventh ACM conference on hypertext (Hypertext '96)*, Washington DC (Mar. 1996)

[Schwabe & Rossi, 1998a] D. Schwabe and G. Rossi, Developing Hypermedia Applications using OOHDM, *Proceedings of the ninth ACM Conference on Hypertext (Hypertext '98)*, Pittsburgh (Jun. 1998)

[Schwabe & Rossi, 1998b] D. Schwabe and G. Rossi, An O.O. approach to web-based application design, *Draft* (1998)

[Snoeck et al., 1999] M. Snoeck, G. Dedene, M. Verhelst and A. Depuydt, Object-Oriented Enterprise modeling with MERODE, *Universitaire Pers Leuven*, Leuven (1999)

[Thüring et al., 1991] M. Thüring, J. Haake, and J. Hannemann: What's ELIZA doing in the Chinese Room - Incoherent Hyperdocuments and how to Avoid them, *Proceedings of the third ACM Conference on Hypertext (Hypertext '91)*, San Antonio (Nov. 1991)

[Thüring et al., 1995] M. Thüring, J. Hannemann and J. Haake: Hypermedia and Cognition: Designing for comprehension, *Commun. ACM Vol. 38*, No. 8 (Aug. 1995)

[Trigg, 1988] R. Trigg, Guided Tours and Tabletops: Tools for Communicating in a Hypertext Environment, *ACM Trans. Office Inf. Syst. Vol. 6*, No. 4 (Oct. 1988)

Agent-oriented Architecture for Task-based Information Search System

Lora Aroyo[1] and Paul de Bra[2]
[1]Faculty of Educational Science and Technology, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands
aroyo@edte.utwente.nl

[2]Computing Science Department, Eindhoven University of Technology, P.O.Box 513, 5600 MB Eindhoven, The Netherlands
debra@win.tue.nl

Abstract

The topic of the reported research discusses an agent-oriented architecture of an educational information search system AIMS - a task-based learner support system. It is implemented within the context of 'Courseware Engineering' on-line course at the Faculty of Educational Science and Technology, University of Twente, The Netherlands. AIMS is endowed with an explicit conceptual mapping of the domain knowledge structures and the user model in order to provide user effective educational information support. It is focused on task-based search, supported by conceptual information structuring and visualisation and user modelling. The overall information retrieval process is split into several sub-processes that are distributed among team of information agents. The agents' behaviour is modelled according to the main activities involved in the process of task-based information retrieval. The system we describe is a knowledge-based course information support system whose main goal is to provide students and teachers with an intelligent help in respect to their preliminary defined curriculum tasks and goals. This way they are facilitated to handle efficiently the information within a specific subject domain. The students can use this system to perform and refine search queries for documents and to explore and navigate within the course domain structure. The teacher is facilitated with authoring environment for editing and construction of domain knowledge, course structure and documents database. We will present the general system architecture of AIMS and will give a detailed description of all system modules.

Pilot experiments are being performed within the 'Graphical User Interface Development' course in Eindhoven Technical University, to evaluate and validate the theoretical assumptions behind the AIMS visualisation and related functionality.

1. Introduction

Although the process of searching and finding relevant information is quite an old one, nowadays it becomes very prominent within the context of the on-line and web-based education. Information retrieval, search results and information presentation are the main variables in this process. Their effectiveness is evaluated in respect to search hit rate, document access, search query formulation/reformulation, relevance determination, human-computer interaction and visualisation.

In this paper we discuss an agent-based information system that aims at providing combined adaptive information support for students and instructors within the context of on-line course environments. The main goal is to improve the usability and maintenance of information in such environments. AIMS supports the user in accessing, selecting and understanding the requested information. The idea is to integrate several technologies that supplement to each other in respect to more effective information retrieval within the framework of educational environments. The technologies we combine are intelligent agents, task-based search, concept map for information structuring and information visualisation, and user modelling as an overlay of the domain model (Aroyo, Dicheva, 99). Task-based search approach is considered to achieve better results in respect to the relevancy of the search results. Agent-oriented architecture provides quick and simple internal system communication, as well as flexible option for updating the system with new modules. Concept mapping technique allows for more flexible and dynamic knowledge representation, when supporting domain ontology. Concept map is an attractive information visualisation scheme in a human-like memory representation model (Collins, Quillian 70).

We discuss a prototype of an Agent-based Information Management System (AIMS) that implements the proposed approach. The reported work is performed within the framework of a Ph.D. research project related to the use of Agents technologies in Virtual Study Environments. AIMS was created to support the students from

the Faculty of Educational Science and Technology participating in the course of 'Courseware engineering'. The intention was to integrate AIMS in the web-based course environment already created for this course. We believe that the combination of agent technologies as system architecture together with information visualisation techniques in the context of concept mapping, task-based search algorithm and user modelling has a positive impact on the effectiveness of information search and better system adaptiveness in this process.

The paper starts with an introduction to the problems of educational information search and their solution proposal within AIMS prototype implementation. In the following section we will focus on description of the AIMS general architecture. The last part presents some conclusions and future perspectives.

2. AIMS General Architecture

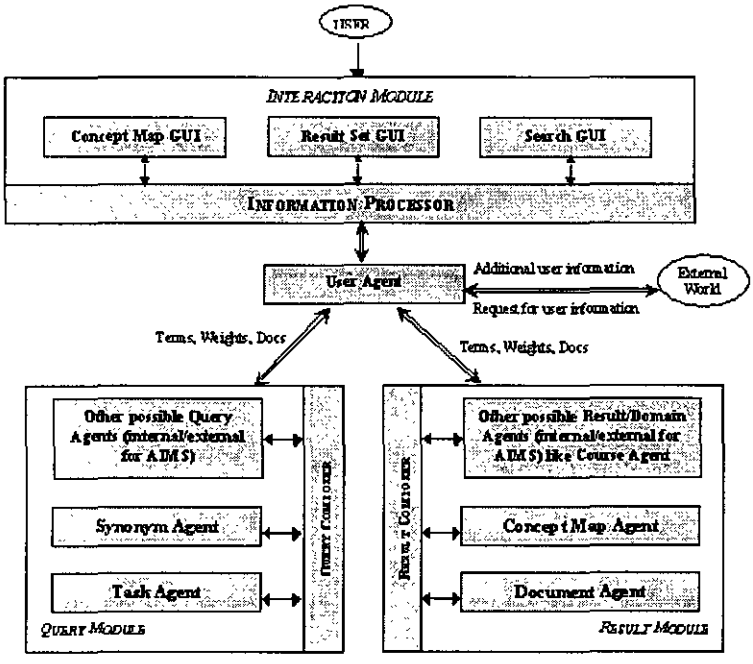


Fig. 1 General System Architecture of AIMS

AIMS introduces the notion of agents as basic system entities, where agents are defined as self- contained problem solving software entities (Wooldridge and Jennings, 1995), which are autonomous, goal-driven and environment sensitive objects, situated within the environment and being able to sense and react to it, over time, in pursuit of its own agenda (Franklin, Graesser, 1997). Thus, an agent-based architectural framework is applied employing multi-agent team for task-based learner support. It involves also attractive visual representation of the information domain, and a conceptual organisation of information resources involving semantic mapping techniques (Aroyo, De Diana, Dicheva 98).

AIMS is built as an agent-oriented architecture. It includes the following main modules: Interaction Module, Query Module, Result Module and User Agent. As the system architecture is agent-oriented, the main activities within AIMS are carried out by agents, who are generally responsible for information retrieval and information presentation activities, domain knowledge representation and building, co-ordination activities among system agents, user profiling and system adaptation. As mediators between the users and the information content, agents are contributing to the general system adaptiveness. As a result of their collaborative intelligent behaviour, the system provides an intelligent user-oriented information support for learners and instructors.

The AIMS team of agents includes the following agent types:

- User Agent - responsible for the entire user information within AIMS. It co-ordinates the data between the main system modules by providing them with specific user preferences;
- Synonym Agent - acts within the query module and performs procedures related to assigning synonyms to keywords;
- Task Agent - refines the presented user query with the terms within the context of the current course task;
- Concept Map Agent - handles the subject domain concept map of terms and links;
- Document Agent - responsible for the update and management of documents within the current subject domain;
- External Agents - different types of domain or query related agents, who can contribute to the search refine and result set construction.

2.1 Interaction Module

The Interaction Module is responsible for handling user input information from the two interaction modes:

- *Browsing and searching mode.* In this mode the user is able to browse within the terms and links of the current subject domain. He or she can also view different metadata for domain terms or search results documents. The users can scan the results set of documents and browse through their content;
- *Refine mode.* In this mode the user is able to reformulate his or her search query. There are two ways of performing this refine activity - direct text refine or visual refine. In the first case users can include or excluding keywords. In the second case, users can visually with the mouse select terms from the subject domain concept map to refine their search. They can also visually appoint documents from the result set as the most relevant ones and to perform search query based on their keyword combination;

The interaction module transforms the user input from those two modes in the form of terms, weights and documents and sends them to the other system modules via the user agent. User agent has strategies implemented to recognise the information coming from the two different modes and to interpret it for the other modules.

The components in the Interaction module are communicating with other system modules and entities via the Information Processor. It takes care for the integration of the results from Concept Map, Document Set and Search GUIs in a form readable for the other system entities. Concept Map GUI presents a view of the current request for terms and links in a form of a concept map. Document Set GUI presents the current result set of documents together with their descriptions and details. Search GUI presents the last request for user search query.

2.2 Query Module

The Query Module is responsible for processing and automatic reformulation of user's search query. Its components can be internal or external entities for the system. New components could be integrated in case of need for additional functionality. For the current AIMS purposes we have Synonym and Task agent, which are responsible of reshaping of user search query and adapting it to user preferences and tasks performed. Query Composer is responsible for the generation of a common query as an input for the Search Engine. It communicates with the search engine via the User Agent, by performing several iteration for query refine and final formulation. The User Agent is responsible for supplying the query to the search engine.

2.3 Result Module

The Result Module is responsible for retrieving and integrating the search results from the user's request. It performs searches within the domain knowledge base (documents, terms and links repository). It also takes care of results rating and refining with the user preferences implemented within the User Agent. Result module updates the status of the course assignments and topics progress. It works with Concept Map and Document Agents, which are generating set of result terms and documents for the users query. This module can work also with other external and internal agents, whose tasks are domain knowledge related. Course structure is also maintained within this module. The result module communicates with other system entities via the User Agent. It receives its input from the Result Composer, who is responsible for the integration of the result set from all the agents in this module

2.4 User Agent

User Agent is the central communication entity in AIMS system. User Agent contains strategies for User Model maintenance, information collection, and processing of user information from the GUIs to the agent pools. It receives the integrated user input from the Information Processor and distributes it respectively to Query and Result Modules. It also realises the communication of the system with the External World.

3. Conclusions and future perspectives

The topic of the reported research is the study of agent-based information management in dynamic and complex information environments, where the focus is on effective search in and use of complex information domains.

This paper presents an approach to effective and efficient information learner support by connecting together in one system intelligent agents, information visualisation and information classification techniques. We discussed an agent-based information system AIMS, where the core is a knowledge base containing the domain model, the user model and a set of rules for concept analysis and decision making. Concept mapping technique is used as a main knowledge structuring approach. This model allows organising the different topics around the key concepts (terms) of the domain. The Concept Map is also applied for the information visualisation and user presentation. The system capitalises on the advantages of visualisation approaches, agent technologies, information concept mapping and user-oriented task-based search approaches, while combining their strong points in instructional design context. It keeps the balance between visual and textual information presentation, detailed and general information views, task-based and traditional search approach, user-centred and fully controlled educational environments.

A pilot experiment is being performed to evaluate and validate the theoretical assumptions behind the AIMS visualisation. Sixty students from the 'Graphical User Interface Development' course in Eindhoven Technical University, the Netherlands, are taking part in the experiment. All students were split randomly into two groups - control and experimental. The students in both groups were introduced to the AIMS user interface. The system, which the students in the control group are using is with disabled 'task-based search' option. The experimental group is allowed and encouraged to use the 'task-based search' option. Within these major groups the students were also divided into smaller groups of three in order to complete a course assignment. The students were given the http link to AIMS, an experimental assignment, a questionnaire and a time checking table. The time checking table was handed to them in order to take measures of time in previously assigned breaking points. The course assignment includes course tasks related to user interface evaluation methodologies and the experimental assignment conducts the steps and order of the experiment. The experimental study is conducted in the period of 15 days and the size of the groups remains constant through the duration of the study. As a measurement instrument for this experiment a questionnaire is used constructed especially for this study. The time checking table and the student assignments are also planned as measurement instruments.

The results of the current experiment will be used as input for a second pilot experiment, which will take place in November 1999 with students from the Antwerpen University. Some of the suggestions and recommendations of the students involved in the first experiment will be used for improving the AIMS GUI before the second experiment. The results of both experiments will be summarised and will serve as a basis for developing a new improved version of the system.

References

Aroyo, L., De Diana, I., Dicheva, D. (1998). Agents to Make Your Information Meaningful and Visible: An Agent-Based Visual Information Management System, *Proc. of WebNet'98, Conference*, AACE, Orlando, USA.

Aroyo, L. & Dicheva, D. (1999). Information Retrieval and Visualisation within the Context of an Agent-based Information Management System.Agents that Make Your Information Meaningful and Visible: An Agent-Based Visual Information Management System, *Proc. of the EdMedia'99 Conference*, AACE, Seattle, USA.

Collins, Quillian, . (1970) Semantic Memory. *Readings in Cognitive Science*. California: Morgan Kaufman Publishers.

Franklin, S. & Graesser, A. (1997). Is it an Agent, or Just a Program? *ECAI Workshop, 1996*, Hungary.

Travers, M. (1996). Programming with Agents: New Metaphors for Thinking About Computation, Massachusetts Institute of Technology. Available through WWW: http://mt.www.media.mit.edu/people/mt/thesis/mt-thesis.html.

Wooldridge, M. & Jennings, N. (1995). Intelligent Agents: Theory and Practice, *Knowledge Engineering Preview, 10(2)*.

Computing Science Reports

Department of Mathematics and Computing Science
Eindhoven University of Technology

In this series appeared:

96/01	M. Voorhoeve and T. Basten	Process Algebra with Autonomous Actions, p. 12.
96/02	P. de Bra and A. Aerts	Multi-User Publishing in the Web: DreSS, A Document Repository Service Station, p. 12
96/03	W.M.P. van der Aalst	Parallel Computation of Reachable Dead States in a Free-choice Petri Net, p. 26.
96/05	T. Basten and W.M.P. v.d. Aalst	A Process-Algebraic Approach to Life-Cycle Inheritance Inheritance = Encapsulation + Abstraction, p. 15.
96/06	W.M.P. van der Aalst and T. Basten	Life-Cycle Inheritance A Petri-Net-Based Approach, p. 18.
96/07	M. Voorhoeve	Structural Petri Net Equivalence, p. 16.
96/08	A.T.M. Aerts, P.M.E. De Bra, J.T. de Munk	OODB Support for WWW Applications: Disclosing the internal structure of Hyperdocuments, p. 14.
96/09	F. Dignum, H. Weigand, E. Verharen	A Formal Specification of Deadlines using Dynamic Deontic Logic, p. 18.
96/10	R. Bloo, H. Geuvers	Explicit Substitution: on the Edge of Strong Normalisation, p. 13.
96/11	T. Laan	AUTOMATH and Pure Type Systems, p. 30.
96/12	F. Kamareddine and T. Laan	A Correspondence between Nuprl and the Ramified Theory of Types, p. 12.
96/13	T. Borghuis	Priorean Tense Logics in Modal Pure Type Systems, p. 61
96/14	S.H.J. Bos and M.A. Reniers	The I^2C-bus in Discrete-Time Process Algebra, p. 25.
96/15	M.A. Reniers and J.J. Vereijken	Completeness in Discrete-Time Process Algebra, p. 139.
96/17	E. Boiten and P. Hoogendijk	Nested collections and polytypism, p. 11.
96/18	P.D.V. van der Stok	Real-Time Distributed Concurrency Control Algorithms with mixed time constraints, p. 71.
96/19	M.A. Reniers	Static Semantics of Message Sequence Charts, p. 71
96/20	L. Feijs	Algebraic Specification and Simulation of Lazy Functional Programs in a concurrent Environment, p. 27.
96/21	L. Bijlsma and R. Nederpelt	Predicate calculus: concepts and misconceptions, p. 26.
96/22	M.C.A. van de Graaf and G.J. Houben	Designing Effective Workflow Management Processes, p. 22.
96/23	W.M.P. van der Aalst	Structural Characterizations of sound workflow nets, p. 22.
96/24	M. Voorhoeve and W. van der Aalst	Conservative Adaption of Workflow, p.22
96/25	M. Vaccari and R.C. Backhouse	Deriving a systolic regular language recognizer, p. 28
97/02	J. Hooman and O. v. Roosmalen	A Programming-Language Extension for Distributed Real-Time Systems, p. 50.
97/03	J. Blanco and A. v. Deursen	Basic Conditional Process Algebra, p. 20.
97/04	J.C.M. Baeten and J.A. Bergstra	Discrete Time Process Algebra: Absolute Time, Relative Time and Parametric Time, p. 26.
97/05	J.C.M. Baeten and J.J. Vereijken	Discrete-Time Process Algebra with Empty Process, p. 51.
97/06	M. Franssen	Tools for the Construction of Correct Programs: an Overview, p. 33.
97/07	J.C.M. Baeten and J.A. Bergstra	Bounded Stacks, Bags and Queues, p. 15.
97/08	P. Hoogendijk and R.C. Backhouse	When do datatypes commute? p. 35.

97/09	Proceedings of the Second International Workshop on Communication Modeling, Veldhoven, The Netherlands, 9-10 June, 1997.	Communication Modeling- The Language/Action Perspective, p. 147.
97/10	P.C.N. v. Gorp, E.J. Luit, D.K. Hammer E.H.L. Aarts	Distributed real-time systems: a survey of applications and a general design model, p. 31.
97/11	A. Engels, S. Mauw and M.A. Reniers	A Hierarchy of Communication Models for Message Sequence Charts, p. 30.
97/12	D. Hauschildt, E. Verbeek and W. van der Aalst	WOFLAN: A Petri-net-based Workflow Analyzer, p. 30.
97/13	W.M.P. van der Aalst	Exploring the Process Dimension of Workflow Management, p. 56.
97/14	J.F. Groote, F. Monin and J. Springintveld	A computer checked algebraic verification of a distributed summation algorithm, p. 28
97/15	M. Franssen	λP-: A Pure Type System for First Order Loginc with Automated Theorem Proving, p.35.
97/16	W.M.P. van der Aalst	On the verification of Inter-organizational workflows, p. 23
97/17	M. Vaccari and R.C. Backhouse	Calculating a Round-Robin Scheduler, p. 23.
97/18	Werkgemeenschap Informatiewetenschap redactie: P.M.E. De Bra	Informatiewetenschap 1997 Wetenschappelijke bijdragen aan de Vijfde Interdisciplinaire Conferentie Informatiewetenschap, p. 60.
98/01	W. Van der Aalst	Formalization and Verification of Event-driven Process Chains, p. 26.
98/02	M. Voorhoeve	State / Event Net Equivalence, p. 25
98/03	J.C.M. Baeten and J.A. Bergstra	Deadlock Behaviour in Split and ST Bisimulation Semantics, p. 15.
98/04	R.C. Backhouse	Pair Algebras and Galois Connections, p. 14
98/05	D. Dams	Flat Fragments of CTL and CTL*: Separating the Expressive and Distinguishing Powers. P. 22.
98/06	G. v.d. Bergen, A. Kaldewaij V.J. Dielissen	Maintenance of the Union of Intervals on a Line Revisited, p. 10.
98/07	Proceedings of the workhop on Workflow Management: Net-based Concepts, Models, Techniques and Tools (WFM'98) June 22, 1998 Lisbon, Portugal	edited by W. v.d. Aalst, p. 209
98/08	Informal proceedings of the Workshop on User Interfaces for Theorem Provers. Eindhoven University of Technology ,13-15 July 1998	edited by R.C. Backhouse, p. 180
98/09	K.M. van Hee and H.A. Reijers	An analytical method for assessing business processes, p. 29.
98/10	T. Basten and J. Hooman	Process Algebra in PVS
98/11	J. Zwanenburg	The Proof-assistemt Yarrow, p. 15
98/12	Ninth ACM Conference on Hypertext and Hypermedia Hypertext '98 Pittsburgh, USA, June 20-24, 1998 Proceedings of the second workshop on Adaptive Hypertext and Hypermedia.	Edited by P. Brusilovsky and P. De Bra, p. 95.
98/13	J.F. Groote, F. Monin and J. v.d. Pol	Checking verifications of protocols and distributed systems by computer. Extended version of a tutorial at CONCUR'98, p. 27.
98/14	T. Verhoeff (artikel volgt)	
99/01	V. Bos and J.J.T. Kleijn	Structured Operational Semantics of χ , p. 27
99/02	H.M.W. Verbeek, T. Basten and W.M.P. van der Aalst	Diagnosing Workflow Processes using Woflan, p. 44
99/03	R.C. Backhouse and P. Hoogendijk	Final Dialgebras: From Categories to Allegories, p. 26
99/04	S. Andova	Process Algebra with Interleaving Probabilistic Parallel Composition, p. 81

99/05	M. Franssen, R.C. Veltkamp and W. Wesselink	Efficient Evaluation of Triangular B-splines, p. 13
99/06	T. Basten and W. v.d. Aalst	Inheritance of Workflows: An Approach to tackling problems related to change, p. 66
99/07	P. Brusilovsky and P. De Bra	Second Workshop on Adaptive Systems and User Modeling on the World Wide Web, p. 119.
99/08	D. Bosnacki, S. Mauw, and T. Willemse	Proceedings of the first international syposium on Visual Formal Methods - VFM'99
99/09	J. v.d. Pol, J. Hooman and E. de Jong	Requirements Specification and Analysis of Command and Control Systems
99/10	T.A.C. Willemse	The Analysis of a Conveyor Belt System, a case study in Hybrid Systems and timed μ CRL, p. 44.
99/11	J.C.M. Baeten and C.A. Middelburg	Process Algebra with Timing: Real Time and Discrete Time, p. 50.
99/12	S. Andova	Process Algebra with Probabilistic Choice, p. 38.
99/13	K.M. van Hee, R.A. van der Toorn, J. van der Woude and P.A.C. Verkoulen	A Framework for Component Based Software Architectures, p. 19
99/14	A. Engels and S. Mauw	Why men (and octopuses) cannot juggle a four ball cascade, p. 10
99/15	J.F. Groote, W.H. Hesselink, S. Mauw, R. Vermeulen	An algorithm for the asynchronous *Write-All* problem based on process collision*, p. 11.
99/16	G.J. Houben, P. Lemmens	A Software Architecture for Generating Hypermedia Applications for Ad-Hoc Database Output, p. 13.
99/17	T. Basten, W.M.P. v.d. Aalst	Inheritance of Behavior, p.83
99/18	J.C.M. Baeten and T. Basten	Partial-Order Process Algebra (and its Relation to Petri Nets), p. 79
99/19	J.C.M. Baeten and C.A. Middelburg	Real Time Process Algebra with Time-dependent Conditions, p.33.