TOWARDS MODEL BASED MANAGEMENT OF A NETWORKED UNIVERSITY

Svend Andreas Horgen
Faculty of Information Technology and e-Learning,
Sør-Trøndelag College, Norway
svend.a.horgen@aitel.hist.no

Asbjørn Thomassen
Division of Intelligent Systems,
Norwegian University of Science and Technology,
asbjorn.thomassen@idi.ntnu.no

Abstract

Selecting the most appropriate among the many courses offered in a networked university can be a cumbersome process for both the students and the institutions involved. We have developed techniques to construct a domain model describing the *content* of a course based on existing Web documents of the course. It will be argued that building models of courses in a networked university can improve the e-learning management processes.

Keywords: Quality assurance, knowledge representation, domain models, adaptive hypertext, management, e-learning

1 The need for automated assistance in the management of e-learning

A successful networked university based on e-learning depends on crucial management processes like student curricula planning, organization of exchange programs, quality assurance, and improvement of existing courses. Among important actor groups are the students, the administrators and the individual teachers (course providers). We believe it is possible to automate some of the management processes involved in each group through a *model based approach*, thereby increasing student satisfaction, reducing administrative costs and improving the quality and quantity of the courses offered.

The next section introduces the status in our current project, explaining how content can be modeled from existing static hypertext documents. Thereafter, we propose how such models can actually improve the management of important processes in an e-learning based networked university. We close up the discussion by suggesting some future, promising tracks for work.

2 Building domain models from hypertext

Our work concerns the identification of methods for how the computer can assist in the construction of a domain model for an adaptive hypertext system [Horgen02]. A domain model represents the knowledge in a collection of Web documents (e.g. a university course) in terms of *conceptual descriptions* of *content*. The field of adaptive hypertext concerns how to generate dynamic documents suitable to each user. Schank stresses the need for 1-1 learning situations in education [Schank95], so adaptive hypertext applies to e-learning. By matching a domain model with student models, new Web-pages, whose content are *especially tailored* to the existing knowledge of each individual student, can be generated on-the-fly.

2.1 The overall system

Ashish et.al use formatting information in existing Web pages to "hypothesize the underlying structure" in order to provide integrated access to multiple Web sources in a particular domain. The sources are parsed for sections and subsections relying on Heuristics for font size and indentation spaces [Ashish+97]. We follow a similar approach as the basis for the domain model construction process. Figure 1 visualizes the process towards a domain model. First, the HTML documents at hand are parsed. According to Heuristical rules, various concepts that describe the content of each document are abstracted. Thereafter, the concepts found are related through descriptive relationship types. A more detailed description of how this is done is explained in the following. Finally, note that human evaluation and revision might be necessary.

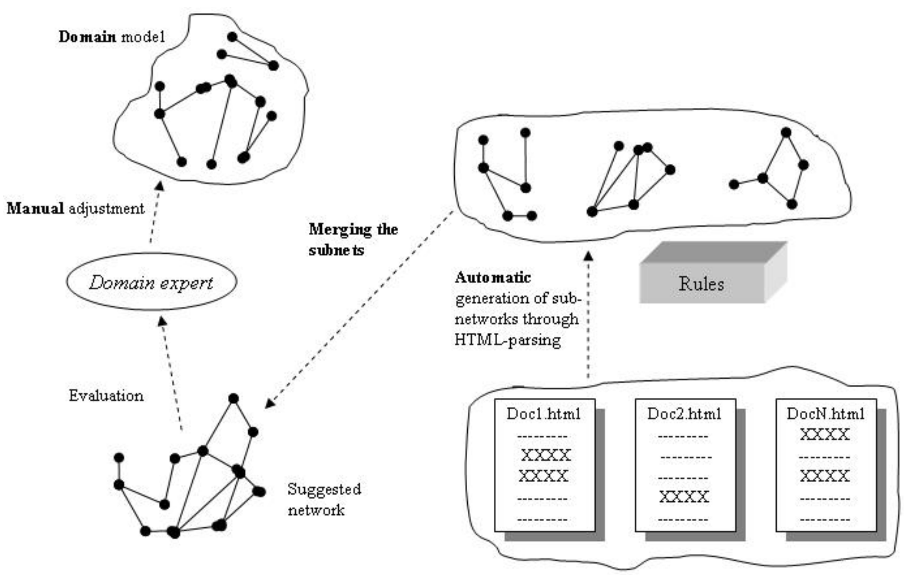

Figure 1: The process of building a domain model representing a collection of HTML documents. After analyzing the documents and merging the results, the domain model can be built. Due to the possible incorrect outcome, manual adjustment might be needed.

2.2 If you're tagged up then prepare for analysis

A domain model can be constructed in many ways. Due to different methods and various approaches, some models might represent the domain knowledge better than others, and a system would perform at best if the domain model is as complete as possible [Davis+93].

We found that HTML tags would be useful in the conceptualization process due to the following facts. First, the nature of hypertext often leads the author to decompose large documents into interlinked smaller ones. Secondly, formatting a document normally involves emphasizing important words, structuring the text through paragraphs, linking related documents both inside and outside the domain, summarizing information in tables and bulleted lists, and using descriptive headings. Beneficially, since the formatting tags are included along with the content in the HTML files, the system can easily search for concepts in different HTML elements like tables, lists, headings, paragraphs and the like.

An Information Retrieval system seems to be useless as an adaptive hypertext system since it provides little or no understanding of document quality [Berners-Lee96]. The basic IR-techniques [Frakes+92], however, may *assist* in the adaptive hypertext system, since what is needed in the conceptualization phase is a proposal of important keywords. The concept candidates are therefore partly a result of a statistical IR-analysis. The selection of final concepts, however, is made by a more intelligent reasoning upon the candidates.

2.3 Heuristics help in the process of finding concepts and relations

Every term in each parsed HTML-element is regarded to be a *candidate* concept. Guided by Heuristics, a conceptualizer module associates *values* with each candidate in an iterative process, finally leaving some with a higher score than others. The most promising candidate is then selected as the concept for the element. The same process applies at both the document and element level. For these processes of "sorting out the wheat from the tares", we found that only using basic IR-techniques like lexical analysis, stopword removal, stemming and term frequency would not yield appropriate results for the task at hand. Tests showed that even though 70% of the terms were actually removed, the remaining set had to be further trimmed in order to be useful. We therefore allowed for the use of a manually constructed domain-specific list (DSL) consisting of keywords regarded as being important for the domain to which the documents belong. Additionally, based on the assumption that the author decides selectively which words to emphasize, the system rewards terms embraced in emphasizer tags (like <B>, <I>, <EM> etc.) Meta, title and header tags are also considered.

In sum, the developed Heuristics build on IR-techniques, HTML tags, the DSL and an analysis of both document and domain structure. The results from the parsing process are passed on to another module for further analysis, which includes a merge of concepts and identification of relationship types. The relationship types identified are either of the `has_deeper_explanation`, `external`, `parent`, `synonymous`, or `prerequisite` type. Examples of some Heuristics used are shown in Table 1:

Meta information and the title element ensure quality to the selection since their content most likely are considered thoroughly from the author's side. Relatively high values can therefore be applied in such cases.	Hyper references between two knowledge sources somewhere within the domain are also relations of type `has_deeper_explanation` between the corresponding concepts.

Table 1: Example of Heuristics for finding concepts (to the left) and relations (to the right).

The domain model consists of the concepts and relations found, and can be visualized as shown in Figure 2. The domain model is actually represented in a file for easy manipulation using the PROLOG computer language.

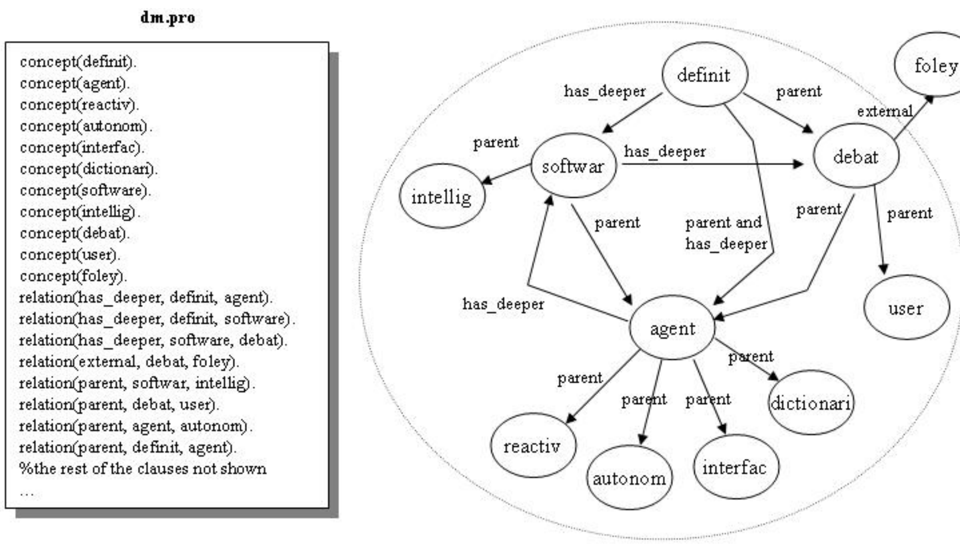

Figure 2: A Prolog representation of the domain model is shown to the left. From the visualization of the concepts (circles) and labeled relations (arrows) found after analyzing a collection of a few documents, we can see that the terms "agent", "software", "definition" and "debate" are key concepts. Indeed, the documents analyzed in this example introduce and discuss the field of Agents.

2.4 Testing and results

A simple prototype of the adaptive hypertext system has been made, and the results seem promising even though a larger empirical test has not yet been performed. In most cases, a domain expert should manually revise the proposal before publishing the final domain model. However, the adjustment task is obviously a lot more pleasant for the author than creating an entire domain model from scratch by hand.

3 Related work

The AHM system uses a model where the documents explain the concepts they are linked to and the links are assigned values that indicate the level of difficulty [daSilva+98].

As mentioned, Ashish et.al use formatting information in existing Web pages to "hypothesize the underlying structure" [Ashish+97] when building the domain model.

The HYPERFLEX system can guide the user to information judged to be relevant by recommending topics based on preferences, goals and needs of different users. Matrices are used to link topics in a document and to link topics with nodes representing particular user goals. The system learns through user feedback, by adjusting the weights of the links [McTear93].

4 Providing new e-learning services

In the previous sections we described how a system can learn the content and the informational properties of a domain and represent this information in a domain model. In this section we introduce how the use of models can provide new services for a virtual university.

4.1 Importance of the models

Models add flexibility to a system. Different models capture the world differently so the choice of representation becomes important [Davis+93]. We believe that domain models describing the knowledge of different courses *at a conceptual level* have advantages for a future networked European university beyond the services offered by existing standards like ECTS. Whereas ECTS yields a standardized way to describe courses [Socrates-Ects], a representation using domain models provides means for matching at a conceptual level, but should be considered as a supplement to ECTS, rather than its competitor.

4.2 Students need guidance when selecting courses

Goals of intelligent tutoring systems (ITS) are curriculum sequencing and interactive problem solving support, which can be achieved using models. In an e-learning setting, an adaptive system should "give hints as to what pages will be most suitable for visiting next" [Weber+97]. Instead of a human advisor reading through many course descriptions in search for an interesting course, a system relying on domain models can offer the student automatic assistance in the difficult process of selecting appropriate courses. Being based on *content*, the models of both domain and the current student knowledge are an excellent foundation for this task. By matching the concepts of a student model with those from various domain models, the task of bridging the gaps[1] in the student model, or advice which courses are most promising to follow in the future, can be achieved. Moreover, the system can also account for student preferences when searching the space of possible curricula. For a concrete picture of a domain model, we refer back to Figure 2.

[1] Gaps in the model reflect gaps in knowledge for the student.

4.3 *Administration*

From the administrative point of view, using models of different courses from various institutions can improve the cooperation in the networked university and ensure quality to the courses offered. The various domain models would reveal which subjects are covered better than others. Merging smaller models into larger ones, would certainly give a good picture of the overall knowledge offered by the entire network.

4.4 *Individual teachers*

Teachers that run the courses might compare a model of their own course with a model from another, similar course, aiming at improvement. Investigation of possible knowledge gaps is another key to this process. E.g. a complex domain model would indicate a difficult course with the need for overview summaries or perhaps a bad structure of the course as is.

The illustration below sums up how models can actually improve the management processes for all three user groups.

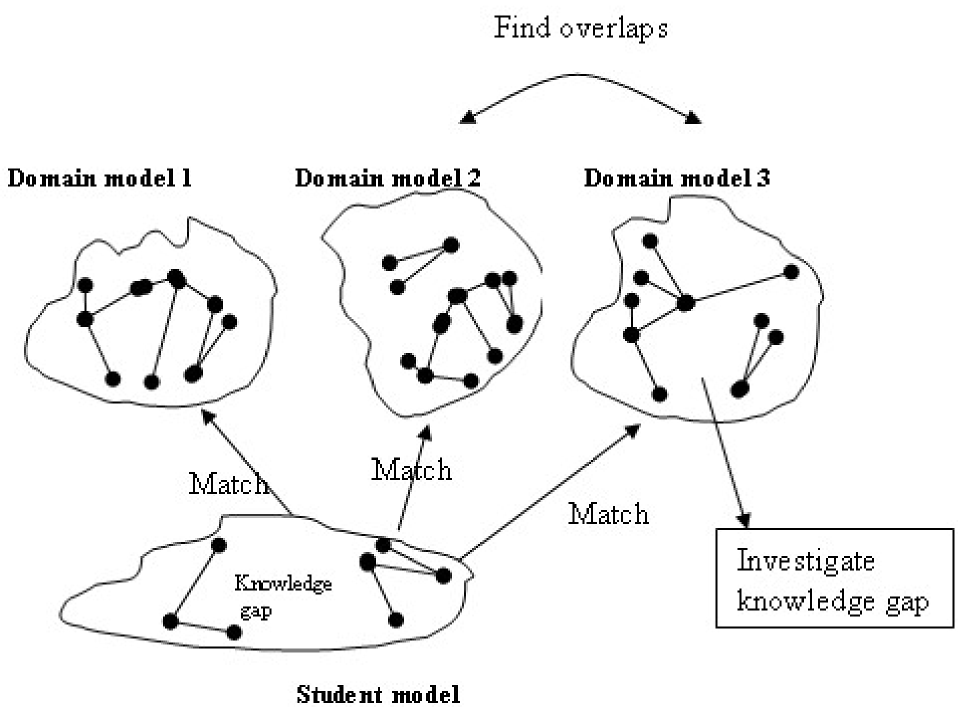

Figure 3: By matching a student model with all domain models, the system can guide the each student with which courses seem most promising. From an administrative point of view, management of courses regarding content can be improved by investigating the corresponding models. Finally, teachers can revise their own courses by peeking at knowledge gaps and compare their model with models from other institutions.

5 Conclusion and future work

In this paper we briefly described our system that generates conceptual domain models based on an analysis of the content in a collection of hypertext documents. Thereafter, we proposed the idea that course modeling can bring forth new services for both the students and the providers in a networked university community. Our work is based on the assumption that hypertext documents already exist for courses in the different institutions. It should be mentioned that making domain models for a course might as well be done manually (e.g. if the course is not published online). We implemented the system in Prolog and C as we wish to integrate the routines into an already existing agent based framework written in C. Future work includes how agents can act as student tutors. Solving queries like "which course to do next" or "compare several models", is a suitable task for an agent.

6 References

Ashish+97 — Ashish, N. - Knoblock, C. Semi-Automatic Wrapper Generation for Internet Information Sources, In Conference on Cooperative Information Systems, 1997, pp. 160-169.

Berners-Lee96 — Berners-Lee, T. Style Guide for Online Hypertext. www.w3.org/provider/style/all.html, unpublished, 1995.

Davis+93 — Davis, R - Shrobe, H - Szolovits, P. What is a Knowledge Representation?, AI Magazine, 14(1), 1993, pp 17-33.

Frakes+92 — Frakes, W. - Baeza-Yates, R. Information Retrieval, Data Structures & Algorithms, Prentice Hall, New Jersey, 1992.

Horgen02 — Horgen, S. A Domain Model For An Adaptive Hypertext System based on HTML, Available on the Web at www.aitel.hist.no/~svendah/ahs.html

McTear93 — McTear, M. User Modelling for Adaptive Computer Systems: A Survey of Recent Developments. Artificial Intelligence Review. (Special issue on User Modelling, edited by M. McTear), Vol 7, 1993, pp. 157-184.

Schank95 — Schank, R. Information is Surprises. Available on the Web at www.edge.org/documents/ThirdCulture/q-Ch.9.html

Socrates-Ects — ECTS – a European Credit Transfer System. Available on the Web at http://europa.eu.int/comm/education/socrates/ects.html

Weber+97 — Weber, B. - Specht, M. User Modeling and Adaptive Navigation Support in www-based Tutoring Systems, Proceedings of User Modeling '97, 1997, pp. 289-300.