Automatic Generation of a Navigation Structure for Adaptive Web-Based Instruction

Judith Masthoff[1]

University of Brighton, UK
Judith.Masthoff@brighton.ac.uk

Abstract. A new approach to adaptive web-based instruction is outlined and advocated, which is based on models and principles that allow the course material to organize itself. This is particularly useful when course material originating from multiple authors is to be combined into a coherent whole.

1 Background

In previous research, an "artificial teacher" has been developed, which adapts instruction to the learner [1]. This artificial teacher needs an abstract description of the course material (often called metadata). Additionally, there needs to be sufficient variation in the course material to be able to reach a high level of adaptation. We are developing a system ('Authoring Coach') that coaches authors to provide both the metadata and the variation needed for the adaptation to be effective. It shall

- Provide an easy user-interface for entering course material and metadata.
- Stimulate authors to enter metadata by clarifying its purpose and consequences.
- Stimulate authors to provide variation by indicating the amount of adaptation possible with the current material and how this can be increased.
- Enable multiple authors to contribute without need for coordination. An essential aspect of the World Wide Web has been that is has organically grown: authors from around the world (without coordination) have contributed material. We use the same principle for the authoring of courseware.
- Generate personalized web-based lesson books from the material provided by the authors, which are easy to use, have a good narrative flow, and allow the artificial teacher to monitor and optimally support student learning.

This paper focuses on the last two points, and expands [8] by describing the generation of a navigation structure in more detail.

[1] The author is supported by Nuffield grant NAL/00258/G.

2 Existing Systems

To model the teaching domain, both InterBook [2] and NetCoach [3] use concepts, which are organized in a network, with links reflecting different types of relationships between them. Concepts in InterBook are "elementary pieces of knowledge for the given domain". All examples mentioned are noun phrases, like "production rule". In NetCoach concepts are "internal representations of pages", like "Chapter-2-1-2". In InterBook, the author provides an electronic textbook that is hierarchically organized (chapters, sections, etc). Each page has a set of *outcome concepts* and a set of *prerequisite concepts* associated with it (analogous to pre- and post-conditions in programming). These are used to support adaptive navigation and hyperlink annotation. NetCoach also uses prerequisites, but as relations between concepts. In MetaLinks [4], authors provide the hierarchical relationship between pages.
In authoring tools like InterBook, MetaLinks, and NetCoach, authors explicitly provide the textbook hierarchy (like page1 has subsection page1-1). In contrast, Authoring Coach will *generate* a hierarchy on the basis of the concept network and the prerequisites and outcomes associated with pages (and some additional information, see below). In a sense, it is related to the Microcosm system [10], which automatically inserts links into pages, but it concentrates more on producing the hierarchy of an educational textbook.

3 Generation of a navigation hierarchy

3.1 How to get a chapter, section, subsection like structure

We would like to generate a chapter, section, subsection like structure. The reason for adopting this kind of structure is that it is easy to learn, and has successfully been used in paper-based books. The system could in principle use five sources of information to base the structure on:

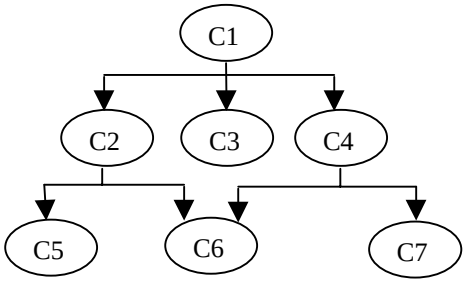
1. The *concept network* can be regarded a domain ontology. It provides information about which concepts are 'children' of which other concepts.
2. *Outcomes* describe what a student will be able to do after successfully studying a page (note: page can be explanation plus exercises). The use of concepts to express outcomes, as in InterBook, is not enough to make the outcomes sufficiently precise and unambiguous. It can result in multiple pages in an electronic textbook that cover apparently the same outcome. This reduces the opportunities for adaptive navigation support and makes it hard to automatically generate navigation structures. As advocated in [8], we express each outcome as a combination of an action verb with a concept. For instance, an outcome of a page will not be "search methods", but something like "explain search methods" or "implement search methods". This will allow authors to specify the outcomes of a page more accurately. As argued in [8], a page title will be generated from the concepts and verbs that describe the page's outcomes.

3. *Prerequisites* describe what a student should be able to do, before starting to study a page. Like outcomes, they are expressed as a combination of an action verb with a concept.
4. Other descriptive information about the pages, like author etc. (See Section 3.2.2).
5. The content of the pages, i.e., the text, images, video, etc they contain.

To generate an initial version of the structure, only the concept network and outcome information is used. The other information will be used to refine the structure, as described in sections below.

Each concept is treated as a section in the electronic book. If a concept is a child of another concept, then it will be included as a subsection. Note that this assumes no cycles in the concept network. Each section contains the pages that use the section's concept in their outcomes. In the navigation structure, the link to each page is its title, which (as in [8]) is generated on the basis of the verbs and concepts used in its outcomes.

For example, suppose a concept network, and pages with outcome and prerequisite relations as given in Figure 1. Figure 2 shows the navigation structure resulting from including the concepts (left), and their associated pages (right).



Page		Verb	Concept
P1	outcome	V1	C1
P2	outcome	V3	C1
P3	outcome	V1	C1
P4	outcome	V2	C6
	prerequisite	V3	C7
P5	outcome	V3	C7
P6	outcome	V1	C6
	prerequisite	V2	C6
P7	outcome	V4	C5
	prerequisite	V1	C1

Fig. 1. Example of a concept network and pages associated with concepts via outcomes and prerequisites. Cs denote concepts, Ps pages, Vs verbs

In the example of Figure 1, all pages have exactly one outcome, and at most one prerequisite. This does not necessarily have to be the case. A page can have multiple outcomes, and multiple prerequisites. Suppose a page has two outcomes: V8 C3 and V9 C5. In that case, the page will appear twice in the hierarchy, once below C3 and once below C5. The page title in that case would be generated from both V8 C3 and V9 C5. In theory, this approach may lead to many occurrences of the same page in the hierarchy, and very long page titles. However, our Authoring Coach will encourage authors to keep pages short (preferably, no scrolling should be necessary), and focussed (preferably, only covering one outcome). Therefore, we expect pages with multiple outcomes to be exceptions, and if a page has multiple outcomes that number to be low.

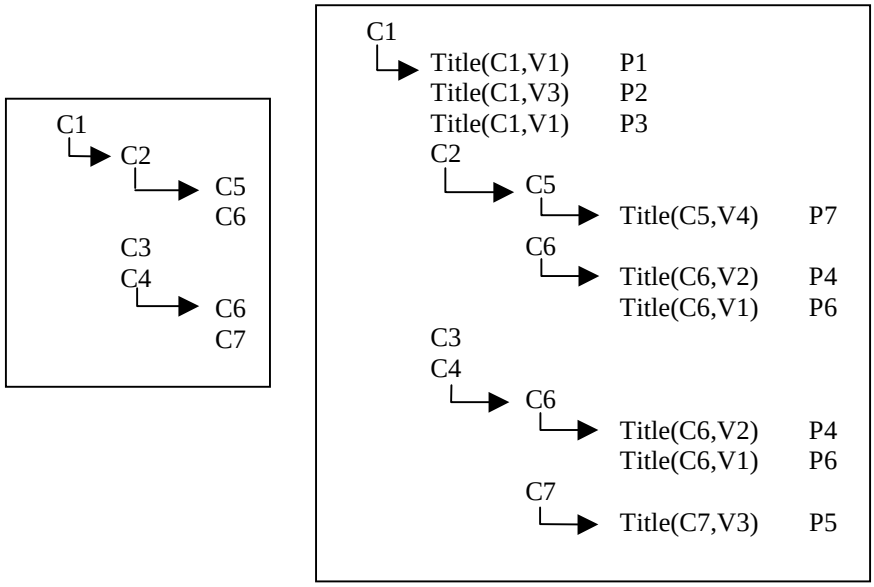


Fig. 2. Navigation structure after including concepts (left), and after including outcomes and associated pages (right). Title(C,V) denotes the title generated for a page with outcome V C.

In this paper, we do not distinguish between different kind of pages. It is, however, envisaged, that there may be a need for at least two kinds: explanation pages that teach the students the outcomes, and exercise pages that assess the student's mastery of the outcomes (a distinction with example and test pages could also be made).

3.2 What to remove?

3.2.1 Removing unnecessary sections

The navigation structure should make it easy and efficient for students to navigate and study the course material. Efficiency can be improved by removing sections that have no content; that do not contain any pages or subsections. For instance, in the example above, clicking on C3 would lead to nothing. So, C3 can be removed. It is also inefficient to have a section with only one page (or one subsection) in it. After all, after clicking on the section, the user would only have a choice of one item. For instance, C5 has only one page associated with it and no subsections, so clicking on C5 would lead to seeing only one link. The user would have to click that link, making it rather inefficient. So, C5 could be removed, and its page P7 could be included as a page of C5's parent C2.

To summarize: A concept will be included in the navigation structure if it has

1. at least two children that are included in the structure, i.e., two subsections, or
2. two pages that use it in their outcomes, or
3. one page that uses it in its outcomes and one child that is included in the structure.

Note that more sophisticated algorithms are possible, which also remove sections if they have more than one child, but their parents do not have many children. The reason for this would be to balance the navigation tree in order to improve efficiency.

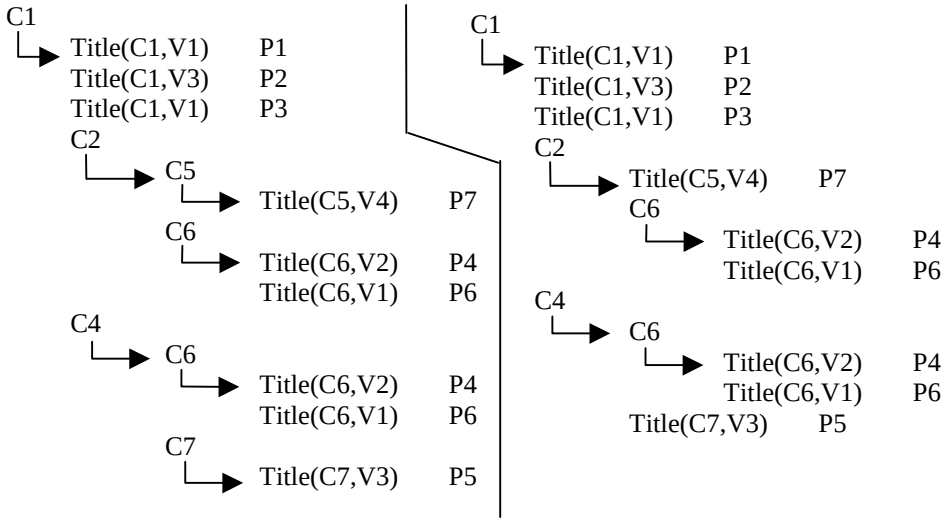


Fig. 3. Navigation structure after removing C3 (left), and after also removing C5 (right).

3.2.2 Dealing with multiple occurrences

The concept network does not have to be a tree: a concept can be a child of multiple other concepts. Additionally, a page can use multiple concepts in its outcomes. In those cases, the generated hierarchy may contain multiple instances of a concept (as section title) or page. Paper-based textbooks include all material only once. So, the question arises whether we should remove multiple occurrences, and, if so, which occurrence to leave. For instance, in the example above, concept C6 is a child of both C2 and C4. As a result, our current navigation structure contains C6 twice, including all its material (P4 and P6). In this example, there are three possibilities:

1. Remove the occurrence of C6 as subsection from C4. This results in Figure 4.
2. Remove the occurrence of C6 as subsection from C2.
3. Leave C6 as subsection of both.

Removing multiple occurrences can substantially simplify the navigation structure. For instance, in the example, removing C6 from C4, not only removes C4 and all its material (Figure 4, left), but also allows the additional simplification of removing C4 (using the rule described in the previous section) (Figure 4, right). However, it poses the problem of deciding which occurrence the leave. This could be the one resulting in the least simplification, or the first one in the hierarchy after ordering (see Section 3.3). An alternative worth considering would be to leave the one the student sees first. For instance, if the student clicks on C2 first, they will see C6. Clicking on C4 afterwards will not show C6. On the other hand, if the student clicks on C4 first, they would see C6, but not see it when clicking on C2 afterwards. Note that this may exclude the possibility of further simplifications.

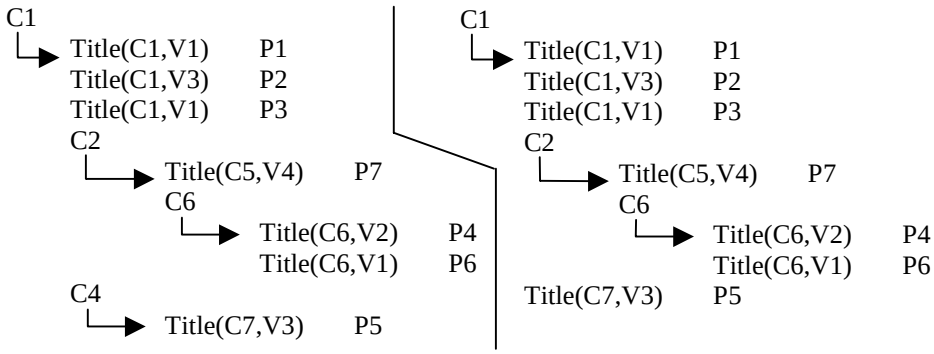


Fig. 4. Navigation structure after removing an occurrence of C6 and its children (left), and after also removing C4 when simplifying according to the rules in Section 3.2.1 (right).

We have, however, chosen the third option: include all occurrences. Removing occurrences would require the student to remember where they had seen certain material, if they wanted to consult it again. Maintaining multiple occurrences will allow the student to find material more easily, and to follow different paths through the course more easily. Whether this is the right decision needs to be investigated in an experiment.

3.2.2 Dealing with alternatives

Different authors may have different views on how a specific outcome should be taught. So, different pages can exist with the same outcomes. We call these pages *alternatives*. As alternatives have the same outcomes, they will share the same generated titles (see [8]). Pages P1 and P3 are alternatives in the example above. As alternatives share the same title, that title needs to be included in the navigation sequence only once.

The decision on which page out of a set of alternatives to include in the textbook can be postponed till the moment that the student (or artificial teacher) clicks on the title for the first time. Several strategies can be used. A page can be chosen on the basis of:

- Author: written by the same author as the previous page in the section (if exists), or by the same author as the last visited page. This may support narrative flow. Students (or teachers) may also provide an ordered list of their favorite authors.
- Its learning style. This would require the author to annotate learning style related characteristics of a page. Alternatively, deductions could be made based upon the page content, like use of images, formulas, words like "example".
- A quality measure. Experts could review (alternative) pages and indicate a quality rating. Students could also rate pages seen. Alternatively, deductions could be made based upon the time spend on the page (not very reliable), links clicks, and the students performance on exercises after having visited the page.
- Its prerequisites being most closely met. Different authors may have different views on the relative order in which outcomes need to be addressed. This can lead to pages sharing the same outcomes, but having a different set of

prerequisites. When links are annotated, like in [7], the annotation should be the most favorable one, i.e., if there is a page with the outcome whose prerequisites have been met, then the "ready to be learned" indication ought to be given.

The student can select an alternative, via links available from the page.

3.2.3 Making a custom book

We want authors to contribute to an organically growing amount of material that may cover every possible topic (as the web does). This makes it highly unlikely that a student would need or want to learn everything. Note that even when restricting the authors to write about a restricted domain, like "Java programming", there still would be large differences between students with respect to their foreknowledge and the outcomes to be reached. That is why courses on different levels tend to use different textbooks.

Different electronic textbooks can be generated to suit individual students or groups of students. Authors, teachers and students can choose the outcomes to be covered by a book. The system will only include pages that are related to achieving those outcomes, either directly or indirectly (needed for prerequisite). The foreknowledge of the student(s) could also be specified, so that pages with outcomes that have already been mastered are not included.

3.3 Ordering siblings

Pages can have outcomes that use different verbs on the same concept. For instance, a page with outcome "implement loops", and another page with outcome "explain loops". We call such pages siblings. In the example above, pages P1 and P2 are siblings, as are P4 and P6. When generating the hierarchy, siblings are ordered in such a way that a page with outcome X will precede all siblings with prerequisite X. For instance, page P4 needs to precede P6, as P4 has outcome V2 C6, which is a prerequisite of P6 (see Figure 1).

Concepts can be children of the same parent concept. We also call such concepts siblings. For example, concepts C2 and C4 are siblings. When generating the hierarchy, as far as possible, siblings are ordered in such a way that a concept with outcome X will precede all siblings with prerequisite X. The outcomes of a concept are the outcomes that contain it and the outcomes of its children. The prerequisites of a concept are the prerequisites that contain it and the prerequisites of its children minus its outcomes. For instance, concept C2 has outcomes V4 C5, V2 C6 and V1 C6, and prerequisites V1C1 and V3 C7. Concept C4 has outcomes V1 C6, V2 C6 and V3 C7, and no prerequisites. As V3 C7 is an outcome of C4 and a prerequisite of C2, C4 should precede its sibling C2.

A page can have an outcome that contains a concept that has a child concept. We call such a page and child concept siblings. For instance, page P1 is a sibling of concept C2. The same kind of rule can be used for ordering them. For instance, as V1 C1 is an outcome of P1 and a prerequisite of C2, page P1 should precede concept C2.

Within these constraints, different strategies can be used to support narrative flow.

- A page or concept with outcome X is followed immediately (or as soon as possible) by a page or concept with prerequisite X.
- A page is followed by another page by the same author.

The relative level of verbs in Bloom's taxonomy [5] can also aid the ordering process. For instance, "explain" precedes "design", which in its turn precedes "evaluate".

Note that the ordering of siblings needs to precede the simplification process of removing not-needed sections, but needs to follow the process of generating a custom book.

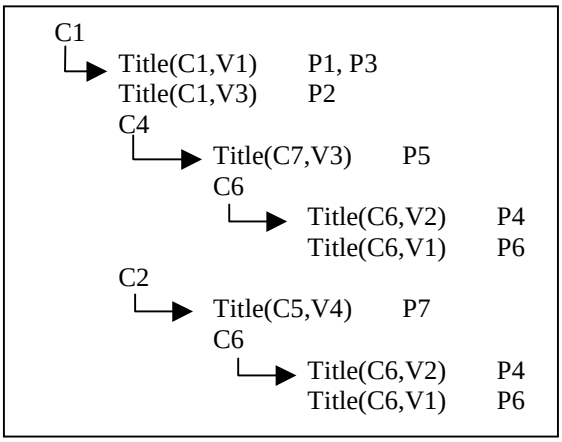


Fig. 5. Navigation structure after ordering siblings.

4 Evaluation

4.1 Criteria for a navigation structure

Before evaluating the automatic generation of a navigation structure, we need to decide what criteria such a navigation structure should meet. The literature on usability, such as [9], mentions a number of guidelines that are particularly relevant for navigation structures:

1. Consistency. The same words should be used to indicate the same objects. This means that pages about the same topics should have similar titles. Our use of generated titles based on outcomes seems to fulfill this criterion. Consistency also means that the navigation structure should not *perceivably* change over time (this does not exclude the use of link annotation). So, if the user has clicked on a link in the past, then clicking on it again should have the same result: show the same submenu, or the same page. User would not expect the sections of a chapter in their textbook to change overnight. We have taken this into account in the way we deal with alternatives (see Sections 3.2.3). If the course content could change

because of authors adding material, consistency could be served by adding a "New section added by author" annotation.

2. Visibility of system status. The user should know at any time where they are, how they got there, and where they can go from there. This advocates the use of a hierarchical, tree-like structure, where the user can immediately see where they came from (parent in the tree) and where they can go next (children in the tree, and perhaps other parts of the tree like parent and siblings). This is why we have chosen to generate that kind of structure.
3. User control and freedom. We believe that the possibility to generate custom books supports user control. However, the automatic generation of the navigation structure might make authors feel less in control. We have to investigate whether it does diminish the authors' feeling in control, and what control over the generation process authors would like to have.
4. Efficiency. Experienced users should be able to find their way around quickly. This means that once a user has seen certain information, they should be able to find it again without any problems. We believe that incorporating all occurrences (see Section 3.2.2) will increase efficiency, but we have to investigate whether that is true.
5. Error prevention. Prevent users from making mistakes rather than just producing good error messages. This means that error messages like "no pages available about this topic" should not occur, by preventing users from clicking on those topics. Our removal of superfluous sections (see Section 3.2.1) is a way of error prevention. Also, our incorporation of multiple occurrences (see Section 3.2.2) can prevent errors (users searching for information in the wrong place).

In addition to these usability related criteria, the main criteria seem to be narrative flow and educational impact (an order which is good for a student learning about the topics).

4.2 Outline of experiments

To evaluate the approach outlined above, we want to conduct a number of experiments. One such experiment could take place at the Adaptive Hypermedia conference. Various experimental designs can be used:

1. A group of subjects, named "Authors" are asked to independently write pages about a certain topic of which they are knowledgeable. At the conference, this could be Adaptive Hypermedia. They are required to provide outcomes and prerequisites for each page, in terms of concepts and verbs provided.
2. Alternatively, an initial network of concepts and an initial list of verbs could be provided, but authors could be allowed to add new concepts and verbs.
3. Alternatively, no concept and verbs would be provided, and authors would have to build up the concept network and verb list themselves.

It can be investigated whether it makes any difference whether

1. Authors can see descriptions of pages added by others, but not the content
2. Authors can see both descriptions and content of pages added by others

3. Neither descriptions nor content of pages added by others. The reason for doing this would be because it might make it more likely that alternatives will occur.

After this has taken place, a second group of subjects can be asked to (independently or as a group) look at the page descriptions (prerequisites, outcomes, author) and concept network and decide on a suitable navigation structure. This can be compared to the navigation structure generated by the Authoring Coach. It can also be investigated how looking at the page content would influence the decision of navigation structure.

6 Conclusions

This paper describes in more detail the automatic generation of navigation structures for adaptive web-based instruction systems that was first introduced in [8]. Though this work is only in its starting phases, it is clear that the choice to generate hierarchies automatically poses interesting problems and opportunities. If we want web-based instruction systems to grow organically, while maintaining narrative flow and user guidance then this seems a way forward.

Clearly, future work should address the evaluation of this approach, and we have sketched how this could be done. It will be interesting to investigate its applicability to different domains. We are also investigating how authors could be supported in producing the concept network (similar problems could occur as those faced in the semantic web). Finally, we are extending our work on the Authoring Coach in the area of giving advice to authors about the completeness, and possibilities for adaptivity of the material produced. Some initial ideas on this were discussed in [8].

References

1. Masthoff, J. An agent-based interactive instruction system. Ph.D. thesis, Eindhoven University of technology. (1997)
2. Brusilovsky, P., Eklund, J., and Schwarz, E. Web-based education for all: A tool for developing adaptive courseware. Computer Networks and ISDN Systems, Proceedings of Seventh International World Wide Web Conference (1998) 291-300
3. Weber, G., Kuhl, H-C., Weibelzahl, S. Developing adaptive internet based courses with the authoring system NetCoach. Proceedings of the third workshop on adaptive hypertext and hypermedia, UM2001, TU/e Computing Science Report 01/11. (2001).
4. Murray, T. MetaLinks: Authoring and affordances for conceptual and narrative flow in adaptive hyperbooks. Submission to the International Journal of Artificial Intelligence in Education. As accessed on http://helios.hampshire.edu/~tjmCCS/cv.html December 2001.
5. Bloom, B.S. (Ed.). Taxonomy of educational objectives: Book 1 Cognitive domain. Longman, London (1979)
6. Reiter, E. and Dale, R. Building natural language generation systems. Cambridge university press (2000)
7. Brusilovsky, P., Schwarz, E., and Weber, G. ELM-ART: An intelligent tutoring system on World Wide Web. In C. Frasson, G. Gauthier, & A. Lesgold (Eds.): Third International

Conference on Intelligent Tutoring Systems, ITS-96. Lecture Notes in Computer Science, Vol. 1086. Springer Verlag, Berlin (1996) 261-269.
8. Masthoff, J. Towards an Authoring Coach for Adaptive Web-Based Instruction. Proceedings of the Adaptive Hypermedia Conference, Malaga, Spain. (2002)
9. Nielsen, J. Heuristic evaluation. In Nielsen, J., and Mack, R.L. (Eds.), Usability Inspection Methods, John Wiley & Sons, New York, NY. (1994)
10. Hall, W., Davis, H., Hutchings, G. Rethinking hypermedia: The microcosm approach (Electronic Publishing Series, No 4). Kluwer Academic Publishers, 1996.