Map-Based Horizontal Navigation in Educational Hypertext

Peter Brusilovsky
School of Information Sciences
University of Pittsburgh
Pittsburgh PA 15260
1 (412) 624 9404
peterb@mail.sis.pitt.edu

Riccardo Rizzo
Italian National Research Council
Institute for Educational and Training Technologies
via Ugo La Malfa 153, 90146 Palermo, Italy
39 (091) 6809207
rizzo@itdf.pa.cnr.it

ABSTRACT
This paper discusses the problem of horizontal (non-hierarchical) navigation in modern educational courseware. We will look at why horizontal links disappear, how to support horizontal navigation in modern hyper-courseware, and our earlier attempts to provide horizontal navigation in Web-based electronic textbooks. Here, we present map-based navigation -- a new approach to support horizontal navigation in open corpus educational courseware that we are currently investigating. We will describe the mechanism behind this approach, present a system KnowledgeSea that implements this approach, and provide some results of a classroom study of this system.

Categories and Subject Descriptors
H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems - *Hypertext navigation and maps.*

General Terms
Documentation, Design, Human Factors.

Keywords
Horizontal navigation, electronic textbook, similarity navigation, concept-based navigation, map-based navigation, Self Organizing Map,

1. INTRODUCTION: THE MISSING LINKS
A hierarchically organized hypertext is becoming a dominant model for publishing educational material. The Web has made the idea and the benefits of hypertext clear to almost every developer of educational content. The overwhelming majority of Web-based and CDROM-based educational material is no longer plain linear text, but hierarchical hypertext published either in HTML of PDF formats. Quite often, the hierarchy is rather simple: a table of content with a list links to individual pages with "lectures" or "chapters" that have no further internal structure. However, there are also many well-developed "electronic textbooks" with deep hierarchical structure (chapters, sections, subsections, etc.) and elaborated hierarchical navigation.

This trend was certainly noticed by the producers of tools for developing educational material [7]. For about 4 years already all major producers of so-called courseware authoring tools such as CourseInfo [3], WebCT [31], TopClass [30] are providing support for creating hierarchically structured hypertext courseware. A good number of even more elaborated tools have been developed by various research teams.

Thousands of "courses" and "textbooks" have been created and published in the form of hierarchical hypertext. However, few of them would be considered as real hypertext by traditionalists. The only kind of navigation provided by the absolute majority of existing educational courseware is hierarchical navigation along the table of contents - from parent to child and, from child to parent, sometimes even to siblings of a node and its parent node. Missing are "classic" *associative* hypertext links and classic ways of navigation - from a page to associated pages that are similar, that can enhance the material presented on the page, explain it differently, present and example. In the context of hierarchically organized educational hypertext we call them *horizontal links* and *horizontal navigation* to stress contrast with *vertical* hierarchical navigation.

Traditionally, the horizontal links and the possibility of horizontal navigation were considered as one of the main benefits of hypertext. Education was often a sample domain for the developers of classic hypertext systems to demonstrate these benefits, but today, very little educational courseware packages have horizontal links. What is the reason that this definitely beneficial kind of links is becoming extinct in educational courseware? What can be done to return these missing links back?

This paper discusses the problem of horizontal navigation in modern educational courseware. We start with discussing the reasons causing the disappearance of horizontal links. We then provide a review of known ways to support horizontal navigation in modern courseware. We present our earlier attempts to support horizontal navigation in Web-based electronic textbooks. The main part of the paper presents map-based navigation - a new approach to support horizontal navigation in open corpus educational courseware that we are currently investigating. We describe the mechanism behind this approach, present a system KnowledgeSea that implements this approach, and provide some results of a classroom study of this system.

2. WHY HORIZONTAL LINKS ARE GONE?
There are a number of reasons for the horizontal navigation to become practically extinct. First, creating horizontal links has

always been a large investment of time. It is quite easy and natural for an author to develop educational material as a hierarchy. It is much harder to provide a well-developed set of horizontal links. Each link requires careful consideration of which pages have to be connected, why, and where to attach the link within a page. Moreover, the whole hyperspace should be better planned and "chunked" in advance so that pages can serve as linked resources for each other. The majority of existing educational materials were created by teachers, who are not professionals in hyperspace organization. They have little time to invest in developing quality educational material. Naturally, horizontal links are considered "extras" that are sacrificed first on the way to getting educational material published.

Creating horizontal links is also technically hard. None of the mentioned above commercial tools for developing educational material support authoring horizontal links (a very interesting fact on itself). In some of these tools like CourseInfo, creating non-hierarchical links is simply impossible. Creating links manually requires some reasonable knowledge of HTML that many authors do not possess, and their materials cannot be retrofitted by third parties.

An even more serious problem is the very nature of the modern educational courseware. A carefully planed hyperspace made by a single author and linked once and forever is no longer the model for creating educational hypermedia. Today, teams of experts create education courseware. Given the whole hyperspace, none of them has the clear understanding needed to provide quality links. For example, a developer of educational problems has little knowledge of problem-solving examples developed by another expert and thus has little chances to provide horizontal links from a problem to examples that may be helpful in solving this problem. Second, educational courseware is quite volatile: the rate of changes is quite fast. Courseware is getting updated after every offering of a course. New problems, examples, explanations are added all the time. Clones and versions of an original course are created for special needs. Maintenance of horizontal links in such a volatile material becomes a headache from both aspects - removing links with disappeared target and adding new links.

Finally, the idea of horizontal links came into conflict with the most recent approach in developing educational courseware that is based on educational objects, content re-use and educational metadata [29]. The paradigm here is that the courseware is created from reusable content objects that can be produced by different authors. In this context authoring of "hardwired" links between pages and other educational objects becomes simply impossible (and forbidden by authoring guidelines) since every object could be re-used outside its original context where these links have no original destination.

3. HOW WE CAN FIGHT BACK?

This section reviews the tools and approaches developed by hypertext researchers in order to support horizontal linking and horizontal navigation. These tools can be divided into two groups. The first group makes the job of creating this links easier to overcome the first two problems mentioned in the previous section. The second group solves the problem of horizontal navigation in large and volatile educational courseware that may be created by multiple authors with the application of re-use paradigm

3.1 Helping Authors to Create Horizontal links

The natural way to support horizontal navigation is to help the authors to create horizontal links. This traditional support is also used in modern educational hypermedia. Unlike commercial courseware authoring tools, advanced research-level tools provide good support for authoring horizontal links. Best tools go far beyond simple help in creating a particular link. There are special methodologies for developing rich horizontal links [1] and tools to author them [23]. There are also tools that can help an author to identify pairs of pages that can be linked, for example, tools based on Self Organizing Maps (SOM) [24]. There are even architectures that provide the user with adaptive navigation support to help traversing horizontal links [13]. Some approaches that can support an author in creating horizontal links manually can also be used for creating horizontal links automatically. *Automatic linking* has been always an important research topic in hypertext area. Known approaches that can be used for design time semi-automatic and automatic horizontal linking are based on semantics [11], word-level similarity [25], and ontology [12]. Automatic horizontal links can be also created "postfactum" by processing navigation traces of real users [4; 20]. The manual, semi-automatic, and automatic approaches cited above are aimed to creating "hardwired" links between pages "once and forever". They make authoring easier and can also be of help in the case of team-based development of educational material where none of the authors have a mental picture of the whole hyperspace.

3.2 Dynamic Horizontal Linking

A similar group of solutions exists for providing horizontal navigation in volatile educational courseware. The problem of horizontal linking in the context where both an origin and a destination of a link can change or disappear was on the agenda of hypertext researchers for a long time and produced a stream of research on *dynamic linking*. Dynamic linking is similar in nature to automatic linking, but the focus is different. Dynamic linking was developed for the contexts where pages in the hyperspace are created separately and at different time. While old pages can be removed and new pages added, dynamic linking maintains horizontal connectivity by creating horizontal links "on the fly" in the run-time or in the editing time. The first dynamic linking solution was suggested by StrathTutor system [22]. StrathTutor allowed an author of an educational hypermedia to mark both link origins and hypertext nodes with a set of keywords. So, every link had an origin, but its destination is replaced by a set of keywords. To resolve such a link in the runtime, the system selects a page with a set of keywords most similar to the link (this target page can actually be added to the hyperspace long after the link itself was created). This indexing-based dynamic linking appeared to fit well to the volatile nature of educational hyper-courseware and was used with some variations in several other educational systems. It also fits well the modern courseware re-use approach where all educational objects are created with metadata that include content index component [16]. A pioneer attempt to use dynamic linking with courseware re-use was done in IDEALS-MTS project [15]. The authors expect more research in this direction in the coming years.

Similarity-based navigation, another traditional approach to dynamic linking, is not specific to educational hypermedia alone. The idea of similarity-based navigation is to use some similarity metric (such as typically used in information retrieval) and dynamically link each document to several most similar ones. This approach provides dynamic links in a huge collection of separately authored documents. Since this paper is oriented to a general hypermedia-literate audience, we do not explain this interesting approach in more details. A very good analysis of similarity-based navigation is provided in [27].

4. CONCEPT-BASED NAVIGATION

In our work on dynamic horizontal navigation in educational hypermedia we were concerned with the existing interface side of automatic and dynamic linking. In all the approaches listed in the previous section, the users are faced with linking decisions made completely by the system. Typically, the system simply adds one or more automatic/dynamic links to similar or relevant pages to all pages observed by the user. The problem is that system's decision about similarity may not always be correct from the user prospect. The pages that are similar to the given page in different dimensions are mixed together even though some of these dimensions may be irrelevant to the given user at the given moment. Still, the only freedom left to the user is to blindly choose one of these pages.

The goal of our work was to give a user more freedom in selecting a link in the context of dynamic horizontal navigation. Following our earlier work on adaptive navigation support [5] when an intelligent hypertext and a user work together in selecting the most relevant among existing links, we wanted to create the same "cooperative" approach for dynamic linking and horizontal navigation. Our first solution to this problem was concept-based navigation [9] pioneered in the ISIS-Tutor system [8] and later refined in the InterBook system [6]. Concept-based navigation was created for the same context of navigation between educational pages indexed with concepts. Instead of generating similarity links from one page to another page for "one-step" navigation, we have provided two step navigation via "concept pages" that InterBook calls "glossary pages" to appeal to a familiar metaphor. In InterBook, every concept used for indexing educational pages has a dedicated page in the hyperspace that is called its *glossary page*. On every glossary page the system shows a brief description of this concept provided by an author (as in a real glossary) and generates links to all pages that have this concept in any part of the index. As a result, a glossary page becomes a "jump-station" for one-step navigation to every page related to the concept.

Conversely, the system generates links from every content page to all glossary pages associated with this concept. InterBook can generate two kinds of glossary links. First, InterBook provides a concept bar to the right of every page that shows all concepts related to the page (i.e., concepts from the page index). This mechanism ensures that a student is always able to see the real educational closure of a page. Each concept name on the concept bar is a link to the proper glossary page. Second, additional links from the text of a page to glossary concepts are generated based upon the concept names. For each keyword or key phrase on a page, the corresponding concept becomes a link to the proper glossary page.

With concept-based indexing, InterBook can build a naturally structured and tightly interlinked hyperspace of educational material, which supports advanced navigation. For example, a student can start from a page which describes several concepts, then move to a glossary page which describes one of these concepts. If the student still cannot completely understand the concept, he or she can navigate to one of the pages that provide an example for the concept. Then, the student can select one of the problems related with the concept to test the obtained knowledge. If the problem appears to be hard, the student can analyze the list of the concepts in the problem spectrum and move from a problem to another concept which is not yet clear (and which can be far away in the network from the starting concept).

Thus, in two steps the user can navigate horizontally from one page to a related page using a glossary page. Part of the job in the process is done by the system that generates glossary links on content pages and content links on glossary pages. The system also annotates adaptively all generated links to further help the user in navigation decision. Another part of the job is performed by the user who can choose the most relevant among several possible directions in horizontal navigation by choosing one of the glossary pages to move to from the content page.

The concept-based navigation approach satisfied all the goals we set for it. It worked perfectly with volatile educational material, since all links were generated on the fly. Removed or changed pages do not create any problems with "hanging" links. A newly indexed page added to this dynamically generated hyperspace will immediately connect to all glossary pages of its index concepts. It also lets the system and the user to work together in selecting the relevant direction for horizontal navigation instead of pushing the user to choose blindly from one of "one-step" similarity links as in traditional approaches.

We have been very happy with the concept-based navigation approach when we were working in closed-corpus indexed educational content. However, we have hit a problem when trying to apply it to open corpus navigation. Our goal was to provide horizontal navigation from authored educational pages to similar pages on the Web. The subject we were exploring was C-programming.

We asked the students of our C-programming courses to use not only our own educational material, but also a number of relevant tutorials on the Web. We have found about 10 good C-tutorials on the Web – all of them traditionally organized as hierarchical "hyper-textbooks". Originally, we have provided links to the root nodes of all these tutorials from one of the top pages in our hypertextual learning material. Analysis of user navigation logs showed quite obvious thing – the users were not using these tutorials at all. The information was helpful, but located too far away in the hyperspace. For example, a user who was not completely happy with our learning material on while loops may benefit from a different presentation of this concept in one of the Web tutorials. To get to the right page the user need to navigate up to the page with roots of all tutorials, than descent down the and possibly repeat this process several times trying to find the most helpful page in several tutorials. None of the students in our class was able or willing to do it. We clearly to provide horizontal navigation links from our course material pages right to the relevant pages of on-line tutorials, but it was also clear that we can't use concept-based navigation. Conceptually, concept-based navigation can work perfectly to provide horizontal connections between course pages and open corpus Web pages – it was even demonstrated in [14]. The problem is that concept-based navigation requires manual indexing of every page of every tutorial – hundreds and hundreds of pages. We simply had no resources to do it this way. After several attempts to stay within concept-based approach using automated indexing we have developed and evaluated a different approach to horizontal navigation that can handle the open corpus educational hyperspace. This approach that we have named *map-based navigation* is presented in the following sections.

5. MAP-BASED NAVIGATION: USER SIDE

Map-based navigation is an approach that we have developed to support horizontal navigation for a "mixed corpus" educational hypermedia. Mixed corpus educational hypermedia includes some traditional closed corpus educational hypermedia material that was specially designed for the needs of a particular course and large portions of an open corpus Web material that is relevant for the course, but was not designed for it. The challenge is to provide horizontal navigation links from closed corpus pages to relevant open corpus pages, as well as between different open corpus pages. We have investigated the problem in the context of creating a supportive Web resource for a typical university class on C programming. In this context, the most often used closed corpus Web-based resource is simply a set of lecture slides – one stack for every lecture. The most easily available open corpus resources are hypertextual C tutorials (like http://www.cm.cf.ac.uk/Dave/C/CE.html). As we have mentioned, hierarchical navigation in this context does not work: it was useless just to point the users to the roots of these tutorials. We have developed KnowledgeSea system to help the user to navigate from lectures to relevant tutorial pages and between them.

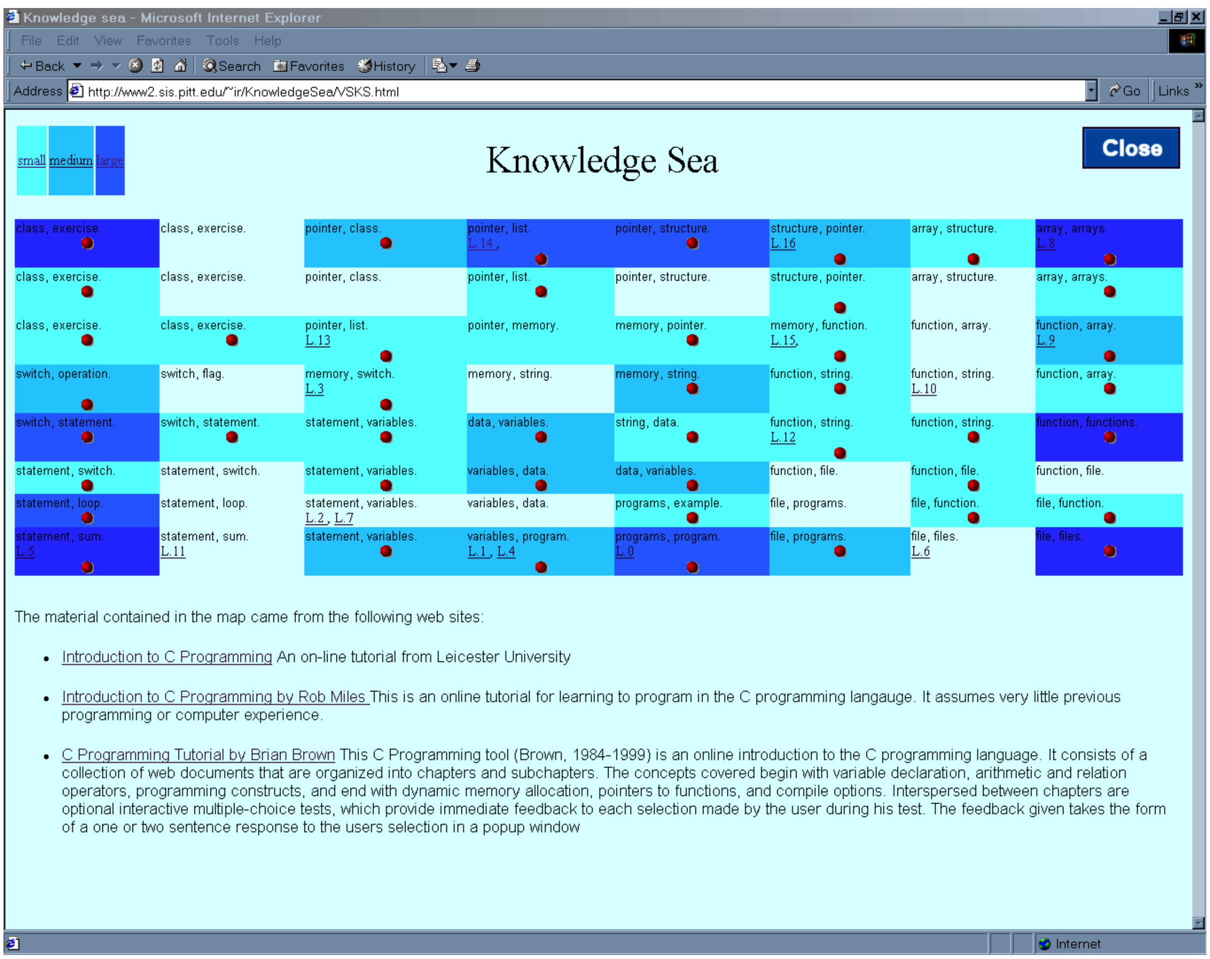

Figure 1: The interface of the KnowledgeSea system. The most compact of the maps is shown.

The core of the KnowledgeSea is a two-dimensional map of educational resources (Figure 1). Each cell of the map is used to groups together a set of educational resources. The map is organized in a way that resources (web pages) that are semantically related are close to each other on the map. Resources located in the same cell are considered very similar. Resources located in directly connected cells are reasonably similar and so on. The map is built using a neural network technology described in the next section. Each cell displays a set of keywords that helps the user to locate the relevant section on the map. A cell also display links to “critical” resources located in this cell. “Critical resources” are those under user consideration, which thereby serve as origin points for horizontal navigation.

For lecture-to-tutorial navigation the critical resources are lectures and lecture slides (see two map cells in the enlarged section on the upper left part of Figure 2). If there is some other educational resources located in the cell a red dot is shown. The cell color indicates the "depth of the information sea" – the number of resource pages lying "under" the cell. Following the metaphor of an information sea, on our map we use several shades of blue in the same way it is used on traditional sea maps to indicate the depths. For example, the light blue color indicates the presence of 1 to 4 related pages, the dark blue indicates more than 10 web pages. The whole set of resources "under" the cell can be observed by "diving". A click on the red dot opens a cell content window (right on Figure 2) that provides a list of links to all tutorial pages relevant to this cell. A click on any of these links will open a resource-browsing window with the selected relevant tutorial page. This page is loaded "as is" from its original URL. It is visualized in another window in order to allow the user to navigate within the tutorial and easily go back to the map. A user can read this page and use it as a starting point to navigate area of interest in the tutorial.

The map serves as a mediator to help the user navigate from critical resources to related resources. The links to critical resources work as landmarks on the map, and, together with the keywords, give an idea of the material organized by the map. If the user is interested to find some additional information on the topic of lecture 14 (devoted to pointers), the first place to look is the cell where the material of this lecture is located (shown as L14 link on the enlarged section of Figure 2). If the user is looking for the material that can enhance the topic of the lecture in some particular direction, the cells that are close to the original cell provide several possible directions to deviate. For example the material related to memory usage in the context of pointers is located underneath of the cell with L14 mark. The links to other critical resources show on the map can help selecting the right direction for deviation. For example, a good place to look for a material that can connect the content of lectures 14 and 15 is a cell between cells where L14 and L15 links are shown. As in the case of concept-based navigation, the map provides a mediated horizontal navigation. Instead of navigating from one page directly to another page, the user moves from a page to a mediator – the information map – that helps the user to select the page related to the original in the “right” sense.

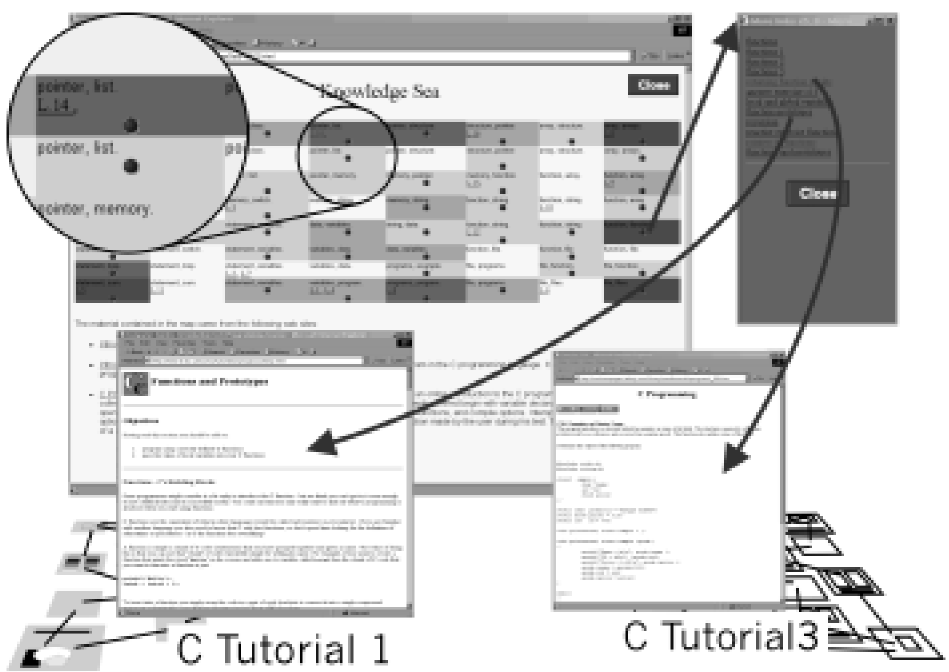

Figure 2: The process of work with KnowledgeSea system. Each cell on the map contains a list of similar pages from three Web tutorials on C language. When the user “dives into” the selected cell, a pop-up window shows the list of links to relevant pages in all available C tutorials. The enlarge cells on the top left shows typical information show on the top of the cell: keywords and links to critical resources (here, slides of lecture 14).

Our knowledge map helps the user to navigate as any regular map, enabling the user to involve spatial orientation and visual memory. The keywords work as legend and the links to critical resources work as landmarks. We have tried to build a stronger connection with real maps by providing several versions of information space maps that differ in the level of details. The version that is shown in Fig 1 is the most concise and the smallest one: only two keywords of each cell is shown and only the lecture number identifies the link to lecture slides. The more detailed map contains six keywords and the number of the lecture, the most detailed shows five keywords and the full title of the lecture. The negative side of having more detailed maps is that they require progressively larger tables and thus are harder to "grasp".

6. THE UNDERLYING TECHNOLOGY

6.1 Information Maps as Spatial Hypertext

The core of the map-based navigation approach is a two-dimensional information map that is built using the ideas of spatial hypertext. Spatial hypertexts allow the user to express the relationships and context of the information in a more flexible way than traditional linking mechanism. In spatial hypertexts the relationship between pieces of information is expressed by using their relative location in a two-dimensional space [21]. A clear advantage of this kind of hypertexts is the possibility to express "constructive ambiguity" [26], that allows the user to create "weak links" between two pieces of information placing them near, but not quite, each other. Two nodes very close are otherwise linked together in a strongest way. Another important advantage of the spatial hypertext is that the user navigation can be supported by visual memory and pattern recognition [21]. Implementation of the spatial hypertexts is almost always a by-hand process, although there are attempts to use automatic techniques [10]. In our case the information map was created automatically using a neural network called Self-Organizing Feature Map [17] as it will be discussed in the following section.

6.2 Self-Organizing Artificial Neural Networks to Develop Information Maps

Artificial neural network (ANN) models have particular properties such as the ability to adapt, to learn or to cluster data. These models are inspired by our present understanding of the biological neural system and are made up of a dense interconnection of simple non-linear computational elements corresponding to the biological neurons. Each connection is characterized by a variable weight that is adjusted, together with other parameters of the net, during the so-called "learning stage". The self organizing networks, and in particular the Self Organizing Feature Map (SOM, sometimes referred as Kohonen map), are ANNs that try to build a representation of some feature of the input vector used as "learning input set" during the learning stage. In this neural network, neurons are organized in a lattice, usually a one or two-dimensional array, that is placed in the input space and is spanned over the input vectors distribution. Using a two-dimensional SOM network it is possible to obtain a map of input space where closeness between units or clusters in the map represents closeness of the input vectors.

Recently this network has been used to classify information and documents in "information maps". These are two-dimensional graphical representations in which all the documents in a document set are depicted. The documents are grouped in clusters that all concern the same topic, and clusters about similar topics are near each other on the map.

The SOM algorithm principle can be explained in an abstract system without reference to any biological structure. The algorithm defines a sort of elastic lattice of simple processing units that are organized in order to fit a set of input points in a high-dimensional input space and to approximate their density function. To each processing unit is associated a vector of weight of the same dimension of the input vectors. Using the weights of each processing unit as a set of coordinates the lattice can be positioned in the input space. During the learning stage the weights of the units change their position and "move" towards the input points. This "movement" becomes slower and at the end of the learning stage the network is "frozen" in the input space.

After the learning stage the inputs can be associated to the nearest network unit. If the surface is visualized the inputs are distributed, as landmark on a map. The main applications of the SOM is the visualization of high-dimensional data in a two dimensional manner, and the creation of abstractions like in many clustering techniques [18].

The effectiveness of the SOM as a tool to cluster information in order to produce links between them and to develop information maps was discussed in many research works. Some studies indicate that the clustering results obtained using the SOM maps can have some meaning for the users. In particular in [19] was validated the proximity hypothesis for which related topics are clustered closely on the map.

In [24] the SOM was trained by using the nodes of a hypertext. The nodes in the same unit or in units connected by the rectangular lattice were considered linked to each other. This organization was compared to the link structure imposed by the author (the number of the link in common was compared to the total number of links). The result can be expressed in terms of "link precision". The 64.5% of the link in the original hypertext was "covered" by the SOM network a result that validate the document organization obtained by using the SOM.

Another important advantage of using the SOM map is that the structure created is a *tessellation* (i.e. a division or splitting) of the of the information space in which documents are represented (the vector space representation techniques will be addressed in the sections below). Each unit of the SOM map identifies an area of the information space (a set of points in the vector space). It will group together all the documents that are represented with a vector that belongs to that area. So that the structure created in the information space after the training space, can be reused to organize other information and documents if they re represented using the same feature of the training set. If a new set of documents is submitted to the trained map it will organize the new information using its "knowledge" on the information domain. This creates a "volatile" link structure that depends on the information space, not on the set of documents and can continue to exist also if all the documents are removed from the map.

This characteristic allows us to develop a scalable system in which new information can be added and old one erased without loosing the spatial organization and the volatile link structure created.

The "volatile" structure created is different from the one addressed by [2]. In that case the structure is always under construction, and documents, and links are continuously deleted or added. In this case links are represented by the neighborhood relationship on the map so it is possible to add or delete document from the map without effect the link relationship: the map will put the document in the right position, creating the relationship to the other documents.

7. IMPLEMENTATION

7.1 Data Representation

In order to organize the documents on a map a mathematical representation of the documents is required. We use the Vector Space Representation (VSR) that is common document encoding based on statistical considerations: each document in a document collection is represented by a vector where each component corresponds to a different word. The component values are calculated by TF*IDF method [28]. In brief, the value depends on the frequency of occurrence of the word in the document (TF component) weighted by the frequency of occurrence in the whole set of documents (IDF component). The calculation of the TF*IDF representation often also includes a normalization factor that is used to obtain a representation vector that is independent from the text length.

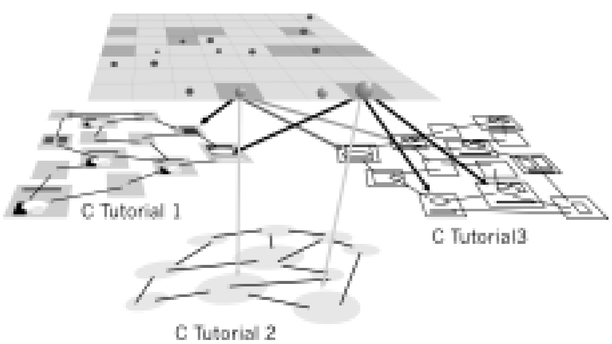

Figure 3: The result of the training phase of the neural network: the pages of the three tutorials on C programming language are grouped under different map cells.

The document set used for the learning phase of the SOM network is constituted by a total number of 210 HTML files from the three tutorials on C programming language. All the pages of the chosen resources were downloaded and processed in order to filter some "noise" (copyright notes, author name and so on). The C code in the page was also removed in order to produce an effective document representation; this should appear strange but it is easy to explain. The TF*IDF representation is a "bag of word" representation i.e. is not able to catch the meaning of the sequence of words (or their context). So it is not possible to catch the meaning of the C code fragment inside a page. Conversely it is possible to catch the meaning of the whole page looking only at the plain text and remove the code that is between the <pre> </pre> tags. Another step in page processing is the extraction of the page title to use as an anchor in the map.

The whole set of pages contains 4249 different words but they were represented using the 500 most common words after the removal of stopwords. All the document representations are collected in a file and submitted to the neural network simulator.

7.2 The system

The user interface part of the KnowledgeSea system is currently implemented as a set of dynamic HTML pages with embedded JavaScript programming. These pages are automatically generated by the following system components:

a set of Perl programs that preprocess the document and create the vector space representation for the learning phase of the SOM map;
the SOM program, the SOM-PAK simulator available over the Internet at http://www.cis.hut.fi/research/som-research/nnrc-programs.shtml
a Perl program that creates the set of HTML pages and JavaScript program that constitutes the interface of the system and memorize the user navigation.
a CGI script that is used to trace and log the user navigation.

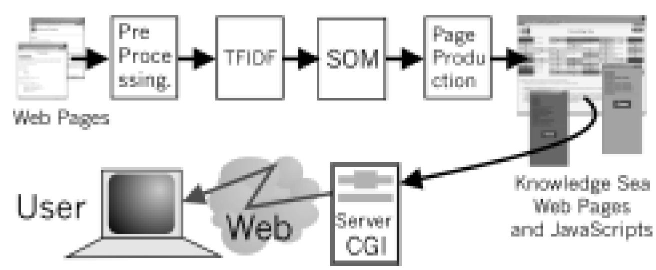

Figure 4: The architecture of the KnowledgeSea system

Figure 4 shows the interaction between the system components. The first step is preprocessing the source files in order to separate content information from the "garbage" that includes formatting and some not relevant information as copyright, author, etc. The second step is the production of a text surrogate by removing stopwords from the text and choosing the set of words that for the vector representation. The third step is the calculation of the document representation. This step was described in more details above.

On the next step, an 8x8 SOM map was used to organize the documents. The size of the map is a compromise between the need for a fine clustering and the need for compactness for visualization purpose. At the end of the learning phase, the map organizes the pages from the various resources. Each cell of the map collects conceptually similar pages from various tutorials. The course lecture slides were processed in the same way as tutorial pages. Each lecture was processed separately with text extraction and C code removal and was finally placed by SOM network into some map cell.

This map was used as input for the Perl script that produce HTML pages with different versions of the SOM map. One of these pages displaying the most compact version of the SOM map is shown on Figures 1 and 2. The JavaScript code embedded into these pages was used to compose and open multiple windows, to collect data about the user navigation, and to send it to the server that was producing the navigation log.

7.3 Map-Based Versus Concept-Based Navigation

It is important to note that the KnowledgeSea system is scalable and can handle large portions of volatile or open corpus learning material. The new learning resources can be added to the map at any time and will be automatically organized into the proper cells of the map on the HTML table. The new documents are processed by about the same way as original set of documents (Figure 4) but faster since is not necessary to train again the SOM map. The link space build by the map is not rigid, it is not a set of defined links between information items as in automatic linking, but an organization of the information space that can be reused many times. In the KnowledgeSea system it is very easy to add a new C tutorial to the already existing information map. This mean that the navigation strategies and the knowledge of the map acquired by the user can also be reused. After several new tutorial are added a user still knows where to go if she wants to look for a material on the particular topic. KnowledgeSea system shows that the map-based navigation approach can support horizontal navigation in large hyperspaces of educational material that includes fragments of open corpus Web resources.

The simplicity of adding the new material to the system distinguishes the Map-based navigation approach from concept-based approach. Adding new resources to be used with concept-based navigation require large investment of time for manual indexing. Adding new resources to the map-based navigation system can be done almost automatically with a very little amount of manual work.

Another difference between the approaches is on the conceptual level: the concept-based navigation used serialistic approach to horizontal navigation since one of many concepts is used a mediator in the process of horizontal navigation. The map-based navigation uses a wholistic approach where the whole map of the information space (with different landmarks) is used as a mediator. Thus, being somewhat similar from the surface point of view (mediator based approaches that enables the user to participate more actively in horizontal navigation), the approaches are really orthogonal by nature and can be successfully used within the same system. We are currently working on a system that supports both concept-based navigation (within an open corpus hypermedia) and map-based navigation.

8. THE STUDY

To evaluate the functionality and the usefulness of the KnowledgeSea system, we have evaluated it in the context of a real C-programming course at the University of Pittsburgh.

The system was available to the students as one of the components of our larger KnowledgeTree system that provides a Web-based access to all learning resources used by the students over the duration of the course. The goal of the KnowledgeSea system was to provide an access to three large hyper-tutorials on C language. As shown on Figures 1 and 2, the information map organized all course lecture slides and all pages from these tutorials. The KnowledgeSea system was available for the students for several weeks during their work with last course lectures and preparation to the final exam. The CGI component mentioned in the section 7 was used to log all user navigation with the system.

During the last week of the course the students were asked to fill-in a short on-line questionnaire about the KnowledgeSea system and their experience with it. The participation was not mandatory, moreover, only those students who spent some considerable time with the system were eligible to fill in the questionnaire. All students who fill-in the questionnaire were rewarded by a few extra credit points. From 39 students in the class 21 choose to participate.

Table 1: The user opinion about the information map organization and the overall performance of the systems. The numbers in the cells show the number of students chosen a particular answer in a multiple-choice questionnaire.

Similarity Questions	Strongly related	Reasonably well related	Poorly related	Not related	Can't judge
The tutorial pages connected to the same cell were	2	18	1	0	0
For a pair of neighboring cells, the overall topics and the connected tutorial pages were	2	17	2	0	0
Performance questions	**Completely**	**Quite well**	**Not quite well, can sometimes be of help**	**It does not help at all**	**No answer**
To what extent the system has achieved the goal of helping the students to access free online tutorials on C language?	1	12	7	1	0

The goal of the questionnaire was twofold. First, we wanted to check how well, from the student's point of view, the map can organize the information. Second, we were interested how useful was the whole system and our particular design decisions. The results that are most relevant to the topic of this paper are presented in Table 1.

As shown by the table, the knowledge organization part of the system performed very well in organizing the open corpus learning material. The students agree that the pages organized under the same and connected cells were quite well related by content. When evaluating the usefulness of the whole system, about 2/3 of the students think that the system has achieved the goal of providing an access to the online C tutorials completely or "quite well" and we are very encouraged with this result. At the same time, several students thought that the systems was helpful only "sometimes" and one student thought that it was not helpful at all. We are currently performing a deeper analysis of the answers and navigation logs to determine why some students has benefited from the system less than others and to find the ways to make the system more useful for everyone.

9. CONCLUSIONS

The system presented is a valuable tool to support the missing horizontal navigation in Web-based education context. The system is based on a self-organizing neural network technology. We choose this technology because it creates the relationship among the resources and, at the same time, produces the information map that is the core of our navigation approach. The approach proposed is less demanding than the concept-based navigation. It allows developing mediated navigation in an non-indexed open corpus document space. The resulting system is scalable and easy to modify because the navigation structure created is not a fixed link structure but a mapping of the information space defined by the document set. New documents can be easily added to the system and placed by the neural network in the proper positions.

The results of our preliminary study shows that map based navigation is a valuable tool for educational hypermedia applications. We also think that map-based navigation is a promising approach for building dynamic hyperspaces in other application areas. We intend to continue exploring map-based navigation in several different contexts. We also interested to compare the concept-based and map-based navigation approaches in the context of single system to determine strong and weak sides of each approach.

10. ACKNOWLEDGMENTS

This work was performed when Riccardo Rizzo was a Visiting Professor at the School of Information Science, University of Pittsburgh. He would like to thank the faculty and the staff of the School for their support during the work on KnowledgeSea.

11. REFERENCES

[1] Bareiss, R. and Osgood, R.: Applying AI models to the design of exploratory hypermedia systems. In: Proc. of Fifth ACM Conference on Hypertext, Seattle, WA, ACM (1993) 94-105.

[2] Bernstein, M., Bolter, J. D., and Mylonas, E.: Architecture for Volatile Hypertext. In: Proc. of 3rd Annual ACM Conference on Hypertext, San Antonio, TX, ACM (1991) 243-260.

[3] Blackboard: CourseInfo, Blackboard Inc. (1999) available online at http://company.blackboard.com/CourseInfo/index.html

[4] Bollen, J. and Heylighen, F.: A system to restructure hypertext networks into valid user models. The New Review of Multimedia and Hypermedia **4** (1998) 189-213.

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

[6] Brusilovsky, P., Eklund, J., and Schwarz, E.**:** Web-based education for all: A tool for developing adaptive courseware. Computer Networks and ISDN Systems. **30**, 1-7 (1998) 291-300

[7] Brusilovsky, P. and Miller, P.: Course Delivery Systems for the Virtual University. In: Tschang, T. and Della Senta, T. (eds.): Access to Knowledge: New Information Technologies and the Emergence of the Virtual University. Elsevier Science, Amsterdam (2001) 167-206.

[8] Brusilovsky, P. and Pesin, L.: An intelligent learning environment for CDS/ISIS users. In: Levonen, J. J. and Tukianinen, M. T. (eds.) Proc. of The interdisciplinary workshop on complex learning in computer environments (CLCE94), Joensuu, Finland, EIC (1994) 29-33, available online at http://cs.joensuu.fi/~mtuki/www_clce.270296/Brusilov.html.

[9] Brusilovsky, P. and Schwarz, E.: Concept-based navigation in educational hypermedia and its implementation on WWW. In: Müldner, T. and Reeves, T. C. (eds.) Proc. of ED-MEDIA/ED-TELECOM'97 - World Conference on Educational Multimedia/Hypermedia and World Conference on Educational Telecommunications, Calgary, Canada, AACE (1997) 112-117.

[10] Chen, C. and Czerwinsky, M.: From Latent Semantic to Spatial Hypertext – An Integrated Approach. In: Grønbæk, K., Mylonas, E. and Shipman III, F. M. (eds.) Proc. of Ninth ACM International Hypertext Conference (Hypertext'98), Pittsburgh, USA, ACM Press (1998) 77-86.

[11] Cleary, C. and Bareiss, R.: Practical methods for automatically generating typed links. In: Proc. of 7th ACM Conference on Hypertext, Washington, DC, ACM (1996) 31-41.

[12] Crampes, M. and Ranwez, S.: Ontology-supported and ontology-driven conceptual navigation on the World Wide Web. In: Proc. of Eleventh ACM Conference on Hypertext and hypermedia (Hypertext 2000), San Antonio, TX, ACM Press (2000) 191 - 199, available online at May 30 - June 3, 2000.

[13] De Bra, P. and Calvi, L.: AHA! An open Adaptive Hypermedia Architecture. The New Review of Hypermedia and Multimedia **4** (1998) 115-139.

[14] Henze, N. and Nejdl, W.: Extendible adaptive hypermedia courseware: Integrating different courses and Web material. In: Brusilovsky, P., Stock, O. and Strapparava, C. (eds.) Adaptive Hypermedia and Adaptive Web-Based Systems. Lecture Notes in Computer Science, Springer-Verlag, Berlin (2000) 109-120.

[15] Hornung, C., Encarnação, L. M., and Barton III, R. J.: PLATINUM: WorldWide distributed courseware production, learning and training using MTS. In: Maurer, H. and Olson, R. G. (eds.) Proc. of WebNet'98, World Conference of the WWW, Internet, and Intranet, Orlando, FL, AACE (1998) 446-451.

[16] IEEE LTCS WG12: LOM: Working Draft Document v6.1, Learning Object Metadata Working Group of the IEEE Learning Technology Standards Committee (2001) available online at http://ltsc.ieee.org/wg12/doc.html

[17] Kohonen, T.: *Self-Organizing Maps*, Springer Verlag, Berlin, 1995.

[18] Kohonen, T., Kaski, S., Lagus, K., and Honkela, T.: Very Large Two-Level SOM for the Browsing of the Newsgroups. In: Proc. of ICANN'96, Berlin, Springer Verlag (1996) 269-274.

[19] Lin, C., Chen, H., and Nunamaker, J. F.: Verifying the Proximity Hypothesis for Self-Organizing Maps. In: Proc. of The 32nd Hawaii International Conference on System Sciences (1999).

[20] Marchand, Y. and Guerin, J.-L.: Nestor: A Trail Blazer for Hypertext. In: Proc. of ICTAI'96, IEEE International Conference on Tools with Artificial Intelligence, Toulouse, France (1996) 420-425.

[21] Marshall, C. C. and Shipman III, F. M.: Spatial hypertext: Designing for change. Communications of the ACM **38**, 8 (1995).

[22] Mayes, J. T., Kibby, M. R., and Watson, H.: StrathTutor: The development and evaluation of a learning-by-browsing on the Macintosh. Computers and Education **12**, 1 (1988) 221-229.

[23] Murray, T.: MetaLinks: Authoring and affordances for conceptual and narrative flow in adaptive hyperbooks. International Journal of Artificial Intelligence in Education **12** (2001) To appear.

[24] Rizzo, R., Allegra, M., and Fulantelli, G.: Hypertext-like structures through a SOM network. In: Proc. of Tenth ACM Conference on Hypertext and hypermedia (Hypertext'99), Darmstadt, Germany, ACM Press (1999).

[25] Rizzo, R., Fulantelli, G., and Allegra, M.: Browsing a Document Collection as an Hypertext. In: Davies, G. and Owen, C. (eds.) Proc. of WebNet'2000, World Conference of the WWW and Internet, San Antonio, TX, AACE (2000) 454-458.

[26] Shipman III, F. M. and Marshall, C. C.: Spatial hypertext: an alternative to navigational and semantic links. ACM Computing Surveys **31**, 4 (1999).

[27] Tudhope, D., Taylor, C., and Benyon-Davies, P.: Navigation via similarity in hypermedia and information retrieval. In: Kuhlen, R. and Ritterberg, M. (eds.) Proc. of HIM'95, Konstanz, Universitätverlag Konstanz (1995) 203-218.

[28] van Rijsbergen, C. J.: *Information Retrieval*, Second ed., Butterworths, 1979.

[29] Verhoeven, B., Cardinaels, K., Van Durm, R., Duval, E., and Olivié, H.: Experiences with the ARIADNE pedagogical document repository. In: Proc. of ED-MEDIA'2001 - World Conference on Educational Multimedia, Hypermedia and Telecommunications, Tampere, Finland, AACE (2001) 1949-1954.

[30] WBT Systems: TopClass, Dublin, Ireland, WBT Systems (1999) available online at http://www.wbtsystems.com/

[31] WebCT: World Wide Web Course Tools, Vancouver, Canada, WebCT Educational Technologies (1999) available online at http://www.webct.com