Accessing Web Educational Resources from Mobile Wireless Devices

P. Brusilovsky[1] and R.Rizzo[2]

[1]School of Information Sciences, University of Pittsburgh, Pittsburgh PA 15260, USA
peterb@sis.pitt.edu
[2]Institute for Educational and Training Technologies, Italian National Research Council, 90146 Palermo, Italy
rizzo@itdf.pa.cnr.it

Abstract. This paper addresses the issue of finding and accessing online educational resources from mobile wireless devices. Accomplishing this task with a regular Web search-and-browse interface demands good interface skills, large screen, and fast Internet connection. Searching for the proper interface to access multiple resources from a mobile computer we have selected an approach based on self-organized hypertext maps. This paper presents our approach and its implementation in the KnowledgeSea system. It also discusses related research efforts and reports the evaluation of our approach in the context of a real classroom.

1 Introduction

Large volumes of relevant educational resources are available currently on the Web for the students of almost any college-level course. In particular, the students of programming-oriented courses could use multiple Web-based courses, tutorials, language and tool manuals, educational simulations, quizzes, and other useful resources available on multiple Web sites. It is currently anticipated that the students access these resources from computers at home or at the university labs. This model contradicts with the popular "anytime, anywhere" slogan of Web-based education. While Web is always "present" the students are can't yet access it anywhere. It is certainly a restriction to an educational flexibility - like a requirement to read a textbook always at home or in class, but not outside, in a café, or while riding a bus. The use of mobile wireless handheld devices potentially allows the students to access educational resources really "anywhere", however, a number of steps have to be preformed to make it really happen. The problem here is not simply technical. Supplying all students with a wireless handheld computer and providing a wireless connection in some large area is an important step towards the solution, but is not a solution on itself. The problem is that almost all expository and objective Web-based educational resources have been designed for relatively large screens and relatively high bandwidth. Special research efforts have to be invested to develop educational resources that are suitable for use with handheld devices or to adapt existing resources for the new platform.

The goal of our group at the Department of Information Science and Telecommunication at University of Pittsburgh is to explore different ways in which mobile wireless devices can be used for college education. Having both information science and telecommunication faculty under the same roof, school-wide wireless network, and dozens of wireless handheld devices, we have very nice settings for developing new systems and exploring them in the classroom. The focus of one of our research project is the access to multiple educational Web resources from mobile devices. As we have mentioned above, there are a variety of Web resources available for any course. None of these resources are usually designed for a specific course. As a result, the resources are often overlap and complement each other at the same time, so multiple resources have to be used for studying almost each topic. For example, in our "Programming and Data Structures" course based on C language, we recommend the students to use several free C language and Data Structures tutorials and other on-line resources (like C language FAQ). Different tutorials cover different topics at different details and also do it using different styles. Altogether, they well complement the course textbook and enable the students with different level of knowledge or different learning styles to get a better comprehension of the subject.

one source (that is usually a textbook). What teacher is usually can do is to provide the links to the home pages of all these tutorials. It is expected that the students will be able to find fragments of these and similar tutorials that are relevant for each lecture. On a desktop computer finding relevant reading fragments in several tutorials is a challenging activity that requires some navigation skills, a large screen and fast Internet connection. Mobile computers with small screens and slower connection simply need another interface to accomplish the same task.

Searching for the proper interface to access multiple resources on a mobile computer we have considered several options and finally selected an approach based on self-organized hypertext maps. This paper presents our approach and its implementation, discusses related works, and reports the results of using our approach in the context of a real classroom.

2. Navigating multiple educational resources with a self organized map

The core of our approach to navigating educational resources is a self-organized hyperspace map. Hypertext maps are generally regarded as one of the most important tools in hypertext navigation. A map can provide a concise navigation and orientation support for relatively large hyperspaces. Traditionally hypertext maps are designed manually by hypertext authors. This manual approach is totally inappropriate for a heterogeneous distributed Web hypespace that has no single author. However, there are a number of known approaches to automated or automatic building of hypertext maps. The approach that we have chosen is based on the Self-Organizing Map (SOM), an artificial neural network that build a two dimensional representation of the inputs. SOM is a very attractive technology for developing compact maps of large hyperspaces since it builds a map representing only the neighborhood relationship between the objects. In these maps only the relative distance between objects is reported and any other information is lost. This is a common practice some kind of maps, for example in subway map in which the representation is distorted in order to be more clear to the user, but the sequence of the station and the intersections are correctly reported (Durant & Kahn, 1998). The map will not respect the distance scale but allows the user to move in the subway system.

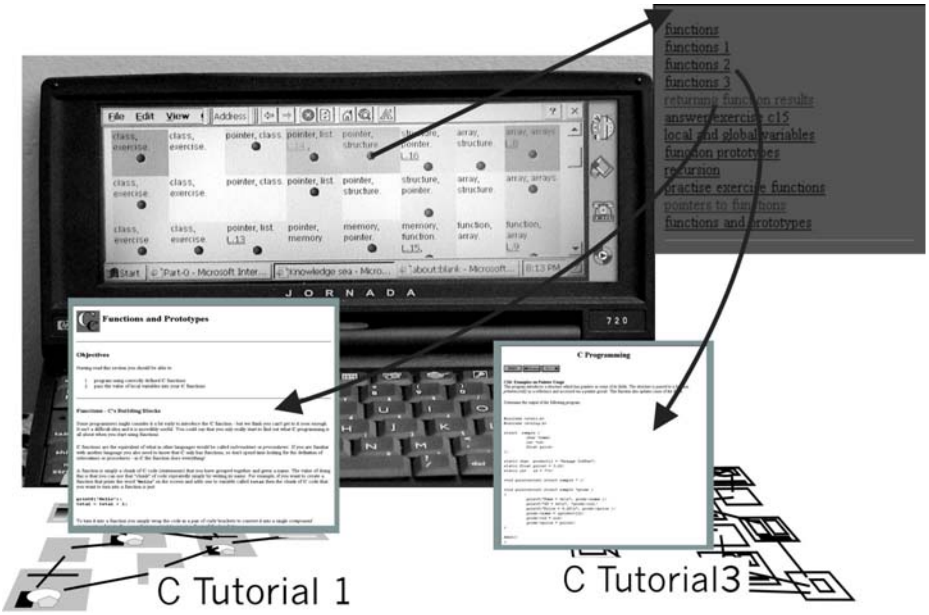

Fig. 1. A session of work with the Knowledge Sea system.

A two-dimensional map of educational resources developed with SOM technology is a core of our KnowledgeSea system for map-based access to multiple educational resources (Figure 1). KnowledgeSea was designed to support a typical university class on C programming. In this context, the goal of the students is to find most helpful material as a part of readings assigned for every lecture in the course. The most easily available Web educational resources are multiple hypertextual C tutorials[1]. In this context, the goal of KnowledgeSea system to help the user to navigate from lectures to relevant tutorial pages and between them.

The users see the KnowledgeSea map as an 8-by-8 table (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.

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. By critical resources we mean resources that are under user consideration thus 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 1). 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 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 that (right on Figure 1) 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 page from one of the tutorial. This page is loaded "as is" from its original URL. 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. These 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 1). 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. The map helps the user to select the page related to the original in the “right” sense.

3 The Mechanism of the Self-Organizing Map

The KnowledgeSea map is automatically built by using an artificial neural network. Artificial neural networks are constitute by a set of interconnected simple processing units tat can “learn” to process the input data by using a supervised learning algorithm or using self-organization. The neural network used to build the document map is the Self-Organizing Map (SOM, sometime referred as Kohonen map) (Kohonen, 1995). In this neural network the units are organized in a sort of elastic lattice, usually two-dimensional, placed in the input space (in our case the hyperspace spanned by the set of documents). During the learning phase this lattice “moves” 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 units of the map can be labeled using the input vectors and the map can be visualized as a two-dimensional surface with the inputs vectors distributed on it. Input vectors that are near each other in the input space are near each other on the map.

[1] See, for example:
http://www.hull.ac.uk/Hull/CC\ Web/docs/cnotes/contents.html, http://www.le.ac.uk/cc/iss/tutorials/cprog/cccc.html,

3.1 SOM Algorithm

The SOM algorithm will be explained referring to a N_1 x N_2 rectangular grid, that is depicted in Figure 2, the extension to the hexagonal grid, that is to be preferred to not favor horizontal and vertical directions, is straightforward.

Each unit $i=\{1,2, \ldots , N_1 \text{ x } N_2 \}$has a weight vector:

$$w_i(t) \lfloor \; \Sigma^n \qquad \textbf{(1)}$$

where *i* define the position of the unit inside the array. The SOM model also contains the *h(c,i,t)* function that defines the "stiffness" of the elastic surface to be fitted to the data points. This function depends on the relative position of the two units *c* and *i* on the network grid and contains some parameters that are updated during the learning stage.

Suppose we have a set of *m* training vectors $\mathbf{X}=\{\mathbf{x}_k,\ k=1,2,\ldots\ m\}$, with $\mathbf{x}_n \lfloor \; \Sigma^n$, during the learning stage these vectors are presented to the network. After enough learning steps the weight of each neural unit will specify a codebook vector for the input distribution, these codebook vectors will sample the input space.

The unit weights (codebook vectors) will be organized such that topologically close units of the grid are sensitive to inputs that are similar. The learning algorithm is below:

1. Initialize the unit weights w_i, the discrete time $t=0$, and the parameters of the function *h(c,i,t)*;
2. Present the input vector $\mathbf{x} \lfloor \; X$;
3. Select the best matching unit *c* (b.m.u.)} as:

$$\left\| \mathbf{x} - \mathbf{w}_c \right\| = \min_{i=1,2,..N_1 x N_2} \left\{ \left\| \mathbf{x} - \mathbf{w}_i \right\| \right\}$$

4. Update the network weights

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + h(c,i,t)[\mathbf{x} - \mathbf{w}_i(t)]$$
$$i = 1,2,\ldots, N_1 \leftrightarrow N_2$$

5. Update the parameters of the function *h(c,i,t)*
6. Increment the discrete time *t*
7. If t=<t_{max} then go to step 2.

The learning function is indicated in step 4. In this step the weight vector of the *b.m.u* and the weight of the nodes that are close to the *b.m.u* in the array will activate and update their weight vectors, moving towards the input vector.

The amount of movement is modulated by the *h(c,i,t)*, the so-called neighborhoods function, a smoothing kernel defined over the lattice points. For the convergence of the algorithm it is necessary that:

$$\lim_{t \blacklozenge \times} h(c,i,t) = 0 \qquad \textbf{(2)}$$

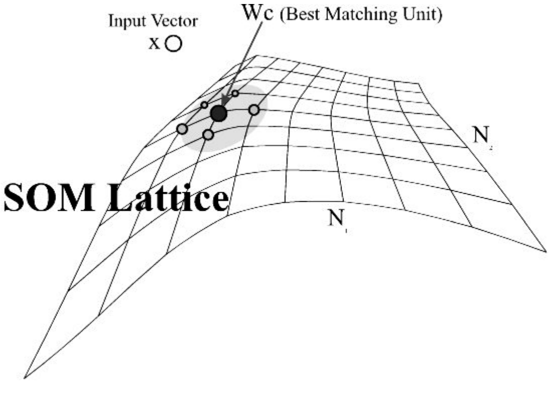

Fig. 2. A representation of the SOM learning algorithm. The gray area is the neighboring of the best matching unit

The *h(c,i,t)* takes the max value on the *b.m.u* and decay if the units are distant from it. In the literature two functions are often used for the *h(c,i,t)*: the simpler one refers to a square neighborhood set of array point around the *b.m.u.* as shown on Figure 3. If their indexes set are denoted $N_c(t)$ the function is defined as:

$$h(c,i,t) = \begin{cases} \alpha(t) & if \quad i \in N_c \\ 0 & otherwise \end{cases} \tag{3}$$

Where:
$N_c(t)$ is a function of time and is shrinking during the time as depicted in Figure 3
$\alpha(t)$ is defined as learning rate and is monotonically decreasing during the time.

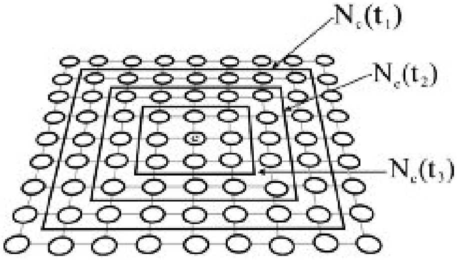

Fig. 3. *Nc(t)* gives the set of nodes that are considered the neighborhood of the node *c*. $t_1<t_2<t_3$

The other widely applied smoothing neighborhood kernel is written in terms of the Gaussian function.

3.2 Parameter values

If the SOM network is not very large (a few hundred nodes at most) the selection of parameter values is not very crucial. As a "rule of thumb", it is possible to start with a fairly wide $N_c(0)$, even more than half the diameter of the network, and letting it shrink with time. An accurate function of time is not very important for the learning rate $\alpha(t)$ it can be linear, exponential or inversely proportional to *t*. The accuracy of the learning depends on the number of steps in the learning phase: they should be at least 500 times the number of the network units.

There is no theoretical way to determine the amplitude of the parameters that have been chose by tentative. By empirical observation the learning stage is divided into two phases of very different length:
ordering phase: in this phase the network organizes the weights of the units in order to roughly approximate the input distribution. The parameters should have the following initial values: α_0 near to the unit (e.g. 0.8) and the smoothing kernel should be large enough to take almost the whole network when the weights are changed.
convergence phase: the convergence phase is the refining phase in which the vectors reach their final positions. It is long 8 or 9 times the ordering phase and during this phase there are not big variations of the unit weights. The parameter α_0 should have a little value, say 0.2 or less, and is constant or slightly decreasing; the smoothing kernel initial value should be enough narrow to change few units or only the *b.m.u.* unit.

A rough way to evaluate the quality of the result obtained after the learning stage is to calculate for each input vector $\mathbf{x}_k \in X$ the *b.m.u. c* and to evaluate the quantity *A* defined as:

$$A = \frac{1}{m} \sum_{k=1}^{m} \left\| \mathbf{x}_k - \mathbf{w}_c \right\| \tag{4}$$

It is convenient to calculate several maps with different initial values and to choose the best result.

4 The Implementation of the System

The neural network constitutes only one of the parts of the developed system. In order to prepare the learning set of the SOM map the HTML document were preprocessed in order to filter some "noise" (copyright notes, author name, HTML tags and so on). After that the TF*IDF document encoding was used. In this statistical encoding each document is represented by a vector were each component corresponds to a different word and the value depends of the occurrence of the word weighted by the frequency of occurrence in the whole set of documents (Salton, Allan & Buckel, 1994). 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. The C code in the pages was also removed in order to produce an effective document representation. This is necessary because the TF*IDF representation doesn't take into account the order of the words that is the key of the meaning f the C code fragments.

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. 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. The training of the map is the subject of one of the next subparagraphs. 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 output of the neural network simulator was used to build the set of HTML pages that constitute the interface of the system and collect the references to the tutorial pages. This set of pages is the one that the user accesses when she is connected to the system. All the pages of the system were designed to be easily visualized on the screen of a handheld PC as the HP Jornada. The home page of the system contains only the map visualized as an HTML table. Each cell of the table corresponds to a neural units of the map and is labeled by some keywords chosen depending on it position in the space of the document.

The system is also scalable: it is possible to add new resource to the system simply building the TF*IDF representation submitting the vector to the Self-Organizing Map. The neural network will classify the new vector in the right cell it will only be necessary to rebuild HTML file of the system interface.

5 A Challenge of a Narrow Screen

In order to choose which map geometry will fit in the small computer device several different maps where trained, using different approaches. Since our first mobile platform was HP Jornada with a relatively wide screen, we have started with a popular 8x8 SOM map. This geometry and this dimension allow having enough space to organize all the documents and are easily to visualize on a desktop computer. The learning stage in this case was not complicated and the standard value of parameter sufficient. The obtained 8x8 map was successfully used by our students for several month and it is this map that was used in a study presented below.

Table 1. Parameters value for the Self-Organizing Map Training

	Ordering phase	Convergence phase 1	Convergence phase 2
t_{max}	10000	30000	50000
α_0	0.2-0.1	0.05-0.02	0.01-0.005
$N_c(0)$	3	2	1

Later, when a wireless card become available for the Palm platform, we have started to experiment with Palm devices. The Palm screen is relatively narrow and with our current Web interface can fit only 3-4 map cells in a row. To adapt the map approach to Palm-size screen, we have decided to explore a non-traditional 4x15 geometry. The goal was to obtain visualization scrollable only in vertical dimension in order make it easy to navigate the map. For this geometry the learning stage was more complicated. First of all the geometry of the map was chosen hexagonal in order to have the cells more tighten, second it was necessary to split the learning phase in three sessions and to use values of the parameter different from the standard ones. The parameter values are in the table 1. A representation of the geometry of the two maps is in Fig. 4.

This is necessary because the map now has the shape of a ribbon, and this “ribbon” needs to touch all the places in the input space populated by documents. To do that it is necessary that the “movement” in the input space is slow.

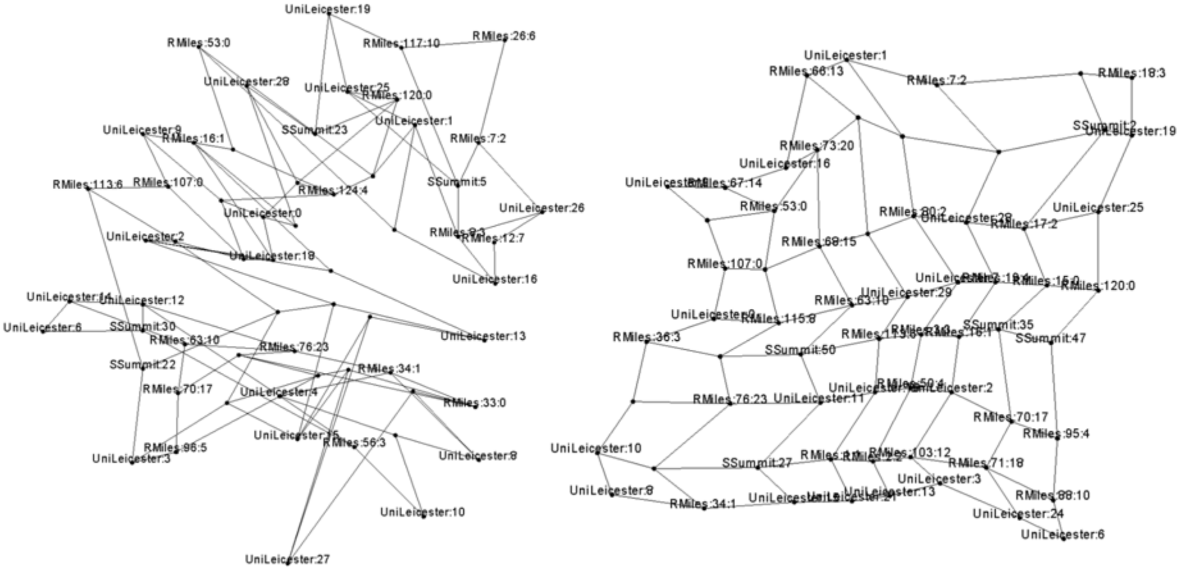

Fig. 4. A representation of the two different map geometries after the learning phase. On the left the 4x15 map and on the right the 8x8 map. The cells are labeled using a reference to the Web pages of the tutorial.

Despite the efforts we have put into developing 4x15 maps, we were not satisfied with the results. The resulting map does not look very natural and contained too many cells with no information attached. We concluded that this map could be more confusing than helpful for the students. Instead of continuing our experiments with narrow maps we are now attempting to develop a better interface for using reliable 8x8 maps on Palm platform.

6 Similar Works

There are a number of known attempts to use SOM for developing various "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, which all concern the same topic, and clusters about similar topics are near each other on the map. 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 (Lin, Chen & Nunamaker, 1999) was validated the proximity hypothesis for which related topics are clustered closely on the map.

In the WEBSOM system a SOM document map was used as a Web interface to classify Usenet newsgroup articles. The paper (Kaski et al., 1998) report the application of SOM network to order 4600 documents based on full text contents. The documents are messages from the "comp.ai.neural-nets" newsgroup. In (Lagus et al., 1996) the document map obtained is capable of organizing 131500 newsgroup messages and was built using a parallel SIMD computer. The Web interface of the *WEBSOM* system is based on an image map where a user can enlarge the region of interest and enter a visualization hierarchy to find the articles needed.

The computational complexity of a SOM neural network is particularly emphasized using TFIDF representation vectors because of the high dimensionality of the vector space obtained. This consideration is taken into account in the paper (Roussinov & Chen, 1998) where it is said that it is difficult to generate a map for large document collections (i.e. Gigabytes of data). The paper proposes a method of improving the speed of the learning stage by exploiting the fact that the representing vectors are sparse vectors, with many zeros, and a new calculation algorithm is proposed.

Our approach combines the ideas of "information mapping" using SOM with the ideas of dynamic navigation in an open corpus hyperspace. Our goal is not simply to "map" the information, but to help the user to navigate from a set of critical items (for example, lectures) to similar items. The use of a map distinguishes our approach from traditional hypertext automatic and dynamic linking. Traditional automatic and dynamic linking ignores user's intelligence in finding relevant hyperspace paths by a "machine intelligence" that can offer a ready to be used one-click links to relevant items. In some sense, it serves as a guide to a blind person in a new city. Our map-based approach relies on both "machine intelligence" in organizing a hyperspace map and user's own intelligence in selecting a proper link on the map. It is similar to providing a city visitor with a map developed by an intelligent professional guide.

7 The Evaluation

The functionality and the usefulness of our map-based information access approach was evaluated it in the context of a real Programming and Data Structures course at the University of Pittsburgh. The KnowledgeSea system was available to the students as one of the component of our KnowledgeTree portal that provides a Web-based access to all learning resources that the students use over the duration of the course. The goal of the KnowledgeSea navigation component was to provide an access to three large hyper-tutorials on C language. As shown on Figure 1, the information map organizes all course lecture slides and all pages from these tutorials. The KnowledgeSea was available for the students for several weeks at the end of the course during their work with last course lectures and preparation to the final exam. User navigation with the system was logged for the further study.

During the last week of the course the students were asked to answer a 13 question 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. In total, 21 students out of 40 choose to participate.

The goal of the questionnaire was twofold. First, we wanted to check how well, from the student’s point of view, the map organizes the information. Second, we were interested how useful was the whole system and our particular design decisions. Nine of the questions asked the students to rate different aspects of the system using a 4-point Likert scale. A sample of results is presented in Table 2.

Table 2: 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 quiestionnaire

Questions	1	2	3	4	Average
The tutorial pages connected to the same cell were (1 - Strongly related to 4 - Not related)	2	18	1	0	1.95
For a pair of neighboring cells, the overall topics and the connected tutorial pages were (1 - Strongly related to 4 - Not related)	2	17	2	0	2.0
To what extent the system has achieved the goal of helping the students to access free online tutorials on C language? (1 - Completely to 4 - It does not help at all)	1	12	7	1	2.38
The overall interface of the system was (1 - Very good to 4 - Poor)	2	11	5	3	2.43

As shown by the table, the knowledge organization part of the system performed very well in organizing the 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 we are very encouraged with this result. The overall idea to attach the resources to cells on a map and show where the lecture belongs was judge “very easy” by the 19% of the students and “quite easy” by the 42.8%. 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.

Trying to find out the features of the system that need an improvement, we have observed that the overall interface was judged “very good” by the 9.5 % of the students and “good” by the 52.3%. This is generally a positive feedback, however, the average score for this question is 2.43 that is the worst among the 9 Likert-scale questions.

The remaining 4 questions were fill-in or multiple answer. One of the questions is particularly interesting for the context of this paper. The students were asked in which context they would expect to use the KnowledgeSea system from a Jornada-like device if it could be accessible from anywhere. We have been surprised that the majority of the students consider to use the system at home or in the library. Only few of them have indicated an interest to use the system in a bus, park or from “anywhere to examine thee material when I have some spare time”, as was asked.

So the system is successful in the integration of different resource and good as a tool for rapid navigation between different resources, however the students prefer to use it in a classical study environment. It is not clear yet whether this answer was caused by the nature of information access task, the nature of the system or simply by the students' prejudice to using computers in a familiar context.

8 Lessons Learned and Future Works

Overall, we can conclude that SOM-based access to multiple information resources is a very useful technology. The 8x8 map that we have explored has worked well for the students. This map is large enough to provide a reasonable split of diverse content, yet is small enough to fit a Jornada-like handheld. We plan to continue investigating the same map and the same interface for the context of a larger hyperspace of educational material (6 or more external tutorials instead of 3).

At the same time, we plan to work on the interface of the system that seems to be its "weakest" feature. The present interface is based on text, keywords and landmarks that are easily visualized on this kind of devices. The vertical scrolling that is necessary to scan the complete map do not effect the navigation probably because there is enough horizontal space to display all the objects. Switching to the smaller PDA thing is getting worse. Although the “ribbon” map was studied to fit the PDA devices it has to be said that it was not a success. Table visualization still needed the horizontal scrolling so that the meaning of the work was lost.

The problems came from the text: the screen is too small to visualize a text table: just two keywords visualized on the screen are enough to completely fill the map cells and the overall view that the goal of the work was not reached. So it will be necessary to change the visualization metaphor. The map is still a good way of visualizing a complex structure, but it will be necessary to better exploit the screen resources of the smaller devices. Our intention is to change the visualization metaphor to make the system suitable also for the smaller wireless devices, mainly Palm platform. At the moment we are considering two options: to switch from text-based visualization to graphical metaphor to fit the PDA screen or to use hierarchical maps. Although hierarchies can be difficult to manage in this application they can be a feasible way to guide the user in navigation. We also working on integrating the map-based information access approach with our earlier work on adaptive hypermedia (Brusilovsky, 2001) and adaptive Web-based systems to develop an adaptive version of KnowledgeSea.

Acknowledgments

Riccardo Rizzo is currently 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 reported in this paper.

References

Brusilovsky, P. (2001) Adaptive hypermedia. *User Modeling and User Adapted Interaction* **11** (1/2), 87-110, http://www.wkap.nl/oasis.htm/270983.

Durant, D. and Kahn, P. (1998) MAPA™: s system for inducing and visualizing hierarchies in websites. In: K. Grønbæk, E. Mylonas and F. M. Shipman III (eds.) Proceedings of Ninth ACM International Hypertext Conference (Hypertext'98), Pittsburgh, USA, June 20-24, 1998, ACM Press, pp. 66-76.

Kaski, S., Lagus, K., Honkela, T., and Kohonen, T. (1998) Statistical Aspect of the WEBSOM System in Organizing Document Collections. In: D. W. Scott (ed.) *Computing Science and Statistics 29*. Fairfax Station, VA: Interface Foundation of NorthAmerica Inc., pp. 281-290.

Kohonen, T. (1995) Self-Organizing Maps. Berlin: Springer Verlag.

Lagus, K., Honkela, T., Kaski, S., and Kohonen, T. (1996) Self-Organizing Maps of Document Collections: a New Approach to Interactive Exploration. In: E. Sinoudis, J. Han and U. Fayad (eds.) (Proceedings of Second International Conference on Knowledge Discovery and Data Mining)AAAI Press, pp.

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

Roussinov, D. and Chen, H. (1998) A Scalable Self-organizing Map Algorithm for Textual Classification: A Neural Network Approach to Thesaurus Generation, Communication and Cognition. *Artificial Intelligence* **15** (1-2), 81-112.

Salton, G., Allan, J., and Buckel, C. (1994) Automatic Structuring and Retrieval of Large Text Files. *Communications of the ACM* **37** (2), 97-108.