Accessing Web Educational Resources from Mobile Wireless Devices: The Knowledge Sea Approach

Peter Brusilovsky[1] and Riccardo Rizzo[2]

[1]School of Information Sciences, University of Pittsburgh, Pittsburgh PA 15260, USA
peterb@sis.pitt.edu
[2]Istituto di Calcolo e Reti ad Alte Prestazioni, Italian National Research Council, 90146 Palermo, Italy
ricrizzo@pa.icar.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, a 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 Knowledge Sea system. It also discusses related research efforts and reports the evaluation of our approach in the context of a real classroom.

1 Introduction

The modern Web is the largest treasury of educational resource has ever been available. It's customary nowadays for college professors to recommend a set of useful Web resources for any lecture and to encourage the students to find more relevant resources themselves. 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-enhanced education. While the Web is always "present" the students can't yet access it from 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 wireless handheld computers and providing a wireless connection in some large area is an important step towards the solution, but is not the 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 the 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, a 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, a variety of Web resources is available for any course. The resources often overlap and complement each other, so multiple resources have to be used for studying almost any topic. For example, in our "Programming and Data Structures" course based on C language, we recommend the students to use several free C language tutorials and other on-line resources (such as C language FAQ). Different tutorials cover different topics with different details and also do it using different styles. Altogether, they well complement the course textbook and enable students with different levels of knowledge or different learning styles to get a better comprehension of the subject.

Unfortunately, it is hard to expect a teacher to provide a list of relevant readings for a lecture from more than one source (that is usually a textbook). What a teacher usually can do is provide the links to the home pages of all these tutorials hoping that the students will be able to locate tutorial fragments that are relevant for each lecture. Unfortunately, as we have found in the course of our research, the students almost never do it. Even on a desktop computer finding relevant reading fragments buried deeply under the tutorial home pages and distributed over several tutorials is a challenging activity that requires good navigation skills, a large screen, and a fast Internet connection (Figure 1). Mobile computers with small screens and slower connection need another interface to accomplish the same task.

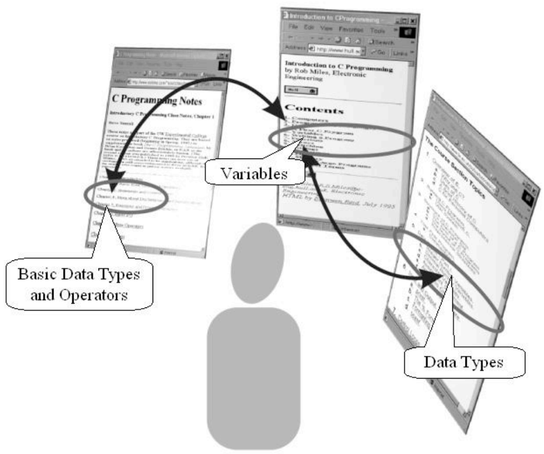

Fig. 1. Studying from multiple on-line resources

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. Hyperspace maps are generally regarded as one of the most important tools in hypertext navigation. A map can provide concise navigation and orientation support for a relatively large hyperspace. Traditionally hypertext maps are designed manually by hypertext authors. This manual approach is totally inappropriate for a heterogeneous distributed Web hyperspace 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 builds a two dimensional representation of the inputs. SOM is a very attractive technology for developing compact maps for a large hyperspace 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.

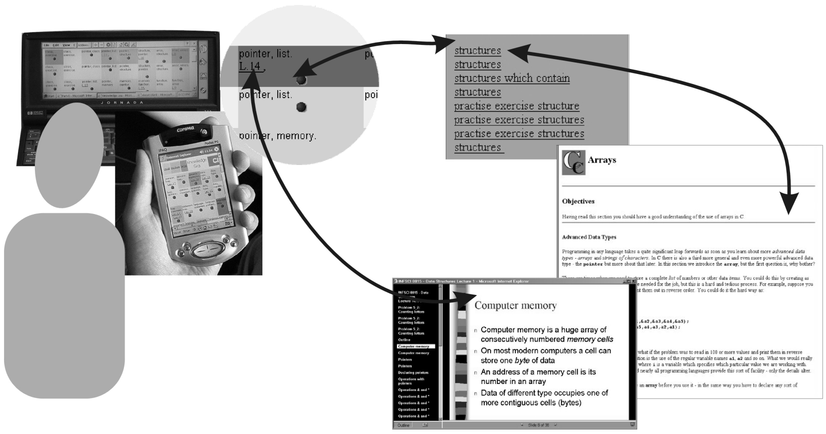

Fig. 2. A session of work with the Knowledge Sea system.

A two-dimensional map of educational resources developed with SOM technology is the core of our Knowledge Sea system for map-based access to multiple educational resources (Figure 2). Knowledge Sea was designed to support a typical university class on C programming. In this context, the goal of the students is to find the 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. In this context, the goal of the Knowledge Sea system is to help the user navigate from lectures to relevant tutorial pages and between them.

The users see the Knowledge Sea map as an 8-by-8 table (Figure 2). Each cell of the map is used to group together a set of educational resources. The map is organized in such 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 locate the relevant section on the map. It also displays links to "critical" resources located in the cell. By critical resources we mean resources that are known to the user and that can serve as origin points for map-based navigation. For example, for lecture-to-tutorial navigation the critical resources are lectures and lecture slides known to the users (see two map cells in the enlarged section on the upper left part of Figure 2). The cell color indicates the "depth of the information sea" – the number of resource pages lying "under the surface" of the cell. Following the "information sea" metaphor we use several shades of blue in the same way they are used on geographic maps to indicate depth. For example, light blue indicates "shallow" cells with just a few resources underneath while deep blue indicates "deep cells" that have the largest number of resources. The resources "under" the cell can be observed by "diving". A click on the red dot opens the cell content window (right on Figure 2) that provides a list of links to all tutorial pages assembled in the cell. A click on any of these links will open a resource-browsing window with the selected relevant page from one of the tutorials. 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 an 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 in finding 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 shown 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 Knowledge Sea map is automatically built by an artificial neural network. Artificial neural networks are formed by a set of interconnected simple processing units that 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, sometimes referred as Kohonen map) [4]. 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 (Figure 3).

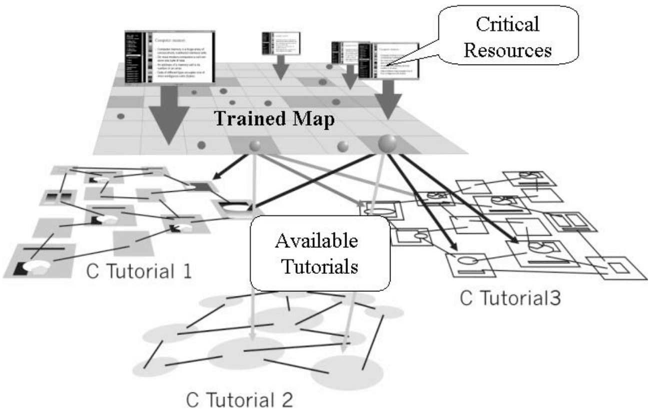

Fig. 3. The organzation of different input and the structure of the map.

3.1 SOM Algorithm

The SOM algorithm is explained below referring to a N_1 x N_2 rectangular grid (the extension to a hexagonal grid that does not favor horizontal and vertical directions is straightforward).

Each unit $i = \{1,2,..., N_1 \times N_2\}$ has a weight vector:

$$w_i(t) \in \Re^n \qquad \textbf{(1)}$$

where *i* defines 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 $\mathbf{X} = \{\mathbf{x}_k, k=1,2,... m\}$, with $\mathbf{x}_n \in \Re^n$. During the learning stage these vectors are presented to the network. After a sufficient number of 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 $\mathbf{x} \in X$;
3. Select the best matching unit *c* (b.m.u.) as:

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

4. Update the network weights

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

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 *b.m.u* and 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 (Figure 4).

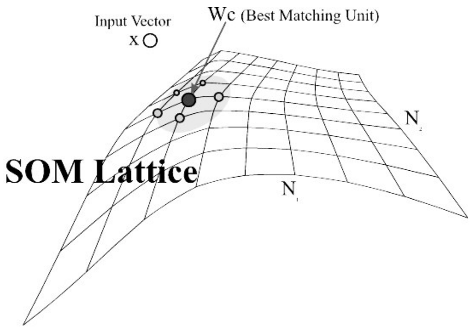

Fig. 4. A representation of the SOM learning algorithm. The gray area is the neighborhood of the best matching unit

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:

$$\lim_{t \to \infty} h(c,i,t) = 0 \tag{2}$$

The *h(c,i,t)* takes the max value on the *b.m.u* and decays on the units that 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 5. If their indexes set is denoted $N_c(t)$ then the function is defined as:

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

Where:

- $N_c(t)$ is a function of time and is shrinking during the time
- $\alpha(t)$ is defined as learning rate and is monotonically decreasing during the time.

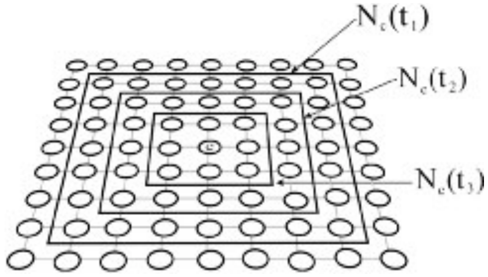

Fig. 5. *Nc(t)* gives the set of nodes that are considered the neighborhood of the node *c*. $t_1<t_2<t_3$

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: it 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 8 or 9 times longer than the ordering phase and during this phase there are no large variations of the unit weights. The parameter α_0 should be small (0.2 or less) and constant or slightly decreasing. The smoothing kernel initial value should be narrow enough to change just a few units or only the *b.m.u*.

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 is just one part of the developed system. In order to prepare the learning set of the SOM map the HTML documents were preprocessed in order to remove "noise" (copyright notes, author name, HTML tags, C code, and so on) and encoded using TF*IDF approach. With TF*IDF, each document is represented by a vector where each component corresponds to a different word. The value of the component is proportional to the occurrence of the word in the document and inversely proportional to its occurrence in the whole set of documents [8]. The calculation of the TF*IDF often includes a normalization factor to obtain a representation vector that is independent from the text length.

The document set used for the learning phase of the SOM network included 210 HTML files from three Web-based tutorials on C programming language. The whole set of pages contained 4249 different words. They were represented by the 500 most common words after the removal of stopwords. All document representations were collected in a file and submitted to the neural network simulator. At the end of the learning phase each cell of the map collected conceptually similar pages from various tutorials.

The output of the neural network simulator was used to build a set of HTML pages that the user accesses interacting with the system. All pages were designed to fit the screen of a handheld PC such 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 unit of the map and is labeled by representative keywords.

The system is also scalable: it is possible to add new resources to the system simply by building the TF*IDF representation and submitting the vectors to the Self-Organizing Map. The neural network will classify the new vectors into the right cells.

5 A Challenge of a Narrow Screen

In order to choose which map geometry will fit small computer devices, several different maps were trained using different approaches. Since our first mobile platform was the HP Jornada with a relatively wide screen, we have started with a popular 8x8 SOM map. This geometry and this size provided enough space to organize all documents. The learning stage in this case was not complicated and the standard value of parameters 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_o	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 (Handspring) platform, we have started to experiment with Palm-based devices. The standard Palm screen is relatively narrow (160 pixels). With our current Web interface it can fit only 3-4 map cells in a row. To adapt the map approach to Palm-size screen, we have explored 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, we had to use the hexagonal geometry for the map to have the cells more tighten. Second, it was necessary to split the learning phase in three sessions and to use non-standard values of the parameters. The parameter values are provided in the Table 1. A representation of the geometry of the two maps is shown in Figure 6.

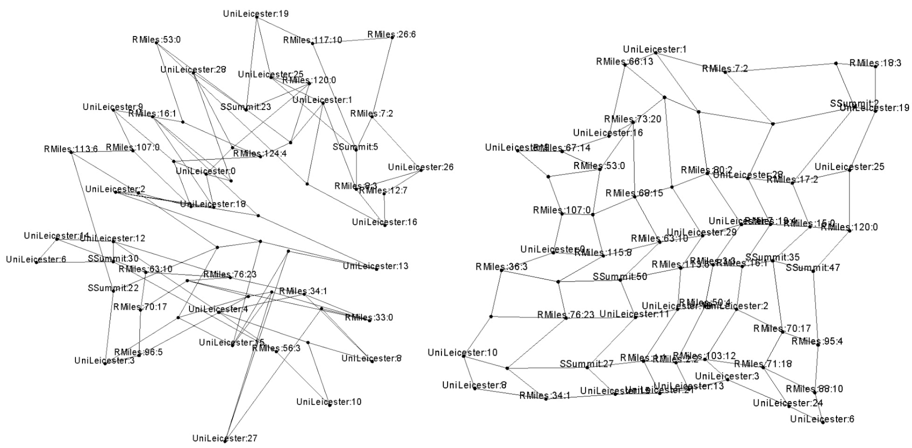

Fig. 6. The geometry of the 4x15 map (left) and the 8x8 map (right) after the learning phase.

Despite the efforts we have put into developing 4x15 maps, we were not satisfied with the results. The resulting map did not look very natural (its geometry on Fig, 6 shows it clearly) and contained too many cells with no information. We concluded that this map could be more confusing than helpful for the students and ceased our work with narrow screens. Fortunately, the introduction of newest wireless Palm devices with 320x320 screens allows us to continue our work with Palm-based devices.

6 Similar Work

There are a number of known attempts to use SOM for developing various "information maps" - two-dimensional graphical representations in which all the documents in a document set are depicted. The documents on a SOM are grouped in clusters. Clusters that group documents on similar topics are near each other on the map. The effectiveness of the SOM as a tool to cluster information and to develop information maps was discussed in many research works. Some studies indicate that the clustering results obtained with SOM maps have meaning for the users. In particular, the proximity hypothesis (related topics are clustered closely on the map) was validated in [6].

In the WEBSOM system a SOM document map was used as a Web interface to classify Usenet newsgroup articles. The paper [3] reports the application of SOM network to organize 4600 documents. The documents were messages from the "comp.ai.neural-nets" newsgroup. In [5], a document map capable of organizing 131500 newsgroup messages was built using a parallel SIMD computer.

The computational complexity of a SOM neural network is particularly emphasized using TF*IDF representation because of the high dimensionality of the resulting vector space. The paper [7] argued that it is difficult to generate a map for large document collections (i.e. Gigabytes of data). This paper proposed a method for improving the speed of learning by exploiting the fact that the representing vectors are sparse vectors with many zeros.

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 navigate from a set of critical items (for example, lectures) to similar items. The use of a map distinguishes our approach from traditional "intelligent" hypertext that explores automatic and dynamic linking. Traditional automatic and dynamic linking ignores the user's intelligence in finding relevant hyperspace paths substituting it by "machine intelligence" that can offer ready to be used one-click links to relevant items. Our map-based approach relies on both "machine intelligence" in organizing a hyperspace map and the 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 two programming-related courses at the University of Pittsburgh. Unfortunately, due to the insufficient number of Jornada organizers we were not able to run a large-scale evaluation of our approach on mobile devices. Instead, we have performed a formative questionnaire-based evaluation of 8x8 Knowledge Sea map used on a desktop computer. We have made the system available to the students of our courses, logged the student interaction with the system, and administered a non-mandatory questionnaire at the end of each course. The analysis of the student answers to some of the questions was partially reported in [2]. It has demonstrated that students regarded Knowledge Sea as a powerful tool for accessing external educational resources. Most impressed the students were with the system ability to place similar resource pages close to each other.

Only one question in the evaluation questionnaire was directly related to the issues of mobile access. The students were asked in which context they would expect to use the Knowledge Sea system from a Jornada-like device if it could be accessible from anywhere. The format of the question was "multiple selection"; the students were able to check any subset of the four offered options that ranged from "in the classroom" to "anywhere". Figure 7 summarizes the answers of 72 students who used the system in the context of an introductory programming during one of the three consecutive semesters (Spring 2002 to Spring 2003). It was a surprise for us to see that the locations selected most often (by about 60% of students) were home and library. Less than 40% of the respondents considered using the system in class and less than 35% "from anywhere". It shows that students are not quite ready for "anytime, anywhere" access. They consider a mobile device more as a different kind of computer and tend to use it in the context where they traditionally use computers (home, lab, and library).

Fortunately, the student attitude to the use of mobile technology in education is changing as rapidly as the mobile devices are becoming common in everyday life. Figure 8 that splits the data presented on Figure 7 into three consecutive semesters shows that the percentage of students who are ready to access our system "from anywhere" has grown steadily over the 1.5 years of our study. At the same time, the percentage of student considering the use of mobile devices in a context where regular computers were more appropriate has declined.

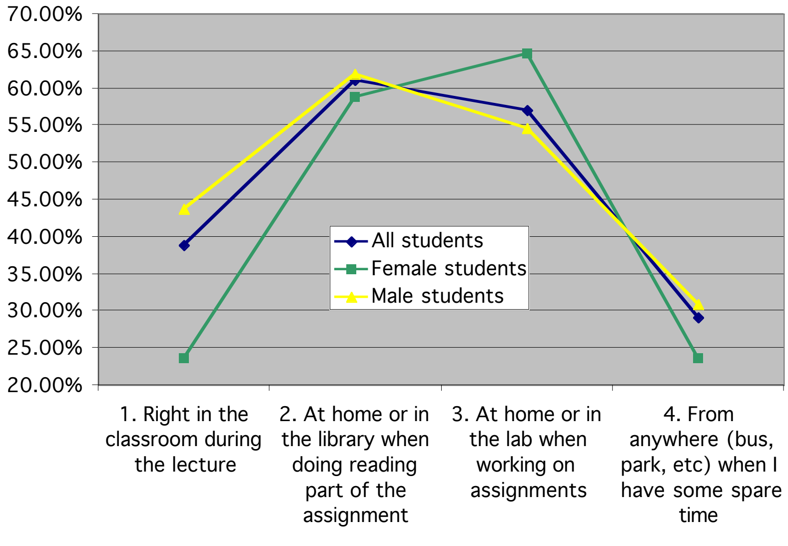

Fig. 7. Percentage of students considering the use of Knowledge Sea in different contexts

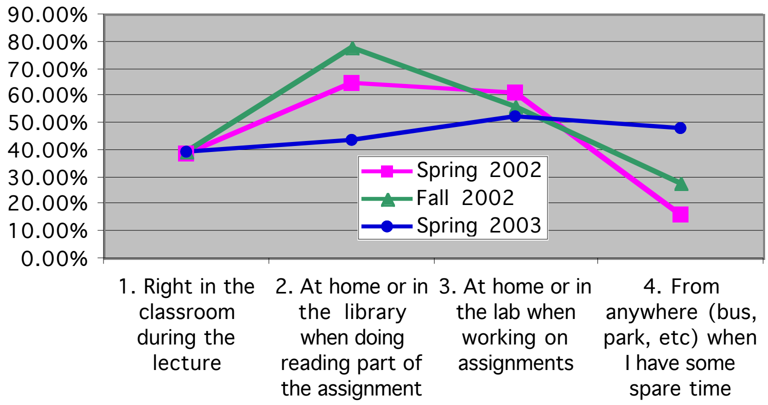

Fig. 8. The change of the percentage of students considering the use of Knowledge Sea in different contexts over three consecutive semesters.

Another observation brought by our study is the difference between the attitude of male and female students to mobile technology. As a cohort, female students who have filled the questionnaire (17 out of 72) were slightly behind their male classmates in being ready to use the Knowledge Sea system outside of traditional context. As shown by Figure 7, female students are more eager to use the system in the currently most traditional "desktop" context - at home when working on an assignment. They are less eager to use the technology in non-traditional places - like a lecture theatre or a bus. Another evidence is that females have checked generally fewer options among the offered four than male students. None of the female students selected all four options (checking all would mean that they are ready to access our system really from any context) while 10% of male students did so. Also, more than 47% of female students checked just one of the four contexts while only about 40% of male students did so.

Summarizing the results we can conclude that many students are not "mentally ready" to use mobile devices for educational needs "anytime, anywhere" as the proponents of the technology hope. Moreover, female students are slightly behind their male classmates in embracing the technology. At the same time, the prospects of educational use of mobile devices look quite bright since the students' attitude to this technology changes rapidly in the desired direction.

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 are now investigating the same map and the same interface in the context of a larger hyperspace of educational material (6 and more external tutorials instead of 3). We are also developing an improved interface for the system and working on integrating the map-based information access approach with our earlier work on adaptive hypermedia [1] and adaptive Web-based systems to develop an adaptive version of Knowledge Sea.

References

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

[2] Brusilovsky, P. and Rizzo, R.: Using maps and landmarks for navigation between closed and open corpus hyperspace in Web-based education. The New Review of Hypermedia and Multimedia **9** (2002) 59-82.

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

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

[5] Lagus, K., Honkela, T., Kaski, S., and Kohonen, T.: Self-organizing maps of document collections: A new approach to interactive exploration. In: Sinoudis, E., Han, J. and Fayad, U. (eds.) Proc. of Second International Conference on Knowledge Discovery and Data Mining, Menlo Park, CA, AAAI Press (1996) 238-243.

[6] 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).

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

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