Education in WWW

Julita Vassileva

In this paper we shall describe briefly the history and tendencies in Computer-Based Learning and our approach to developing computer-based learning on the WWW, the DCG on WWW.

1. State of the Art in Computer-Based Learning

The field of Computer Based Learning (CBL) has been developing since the beginning of the "Computer Age", i.e. from the 60ies. The development of instructional paradigms has been strongly influenced by the technological development. Seven main approaches to computer based learning can be outlined (see Figure 1).

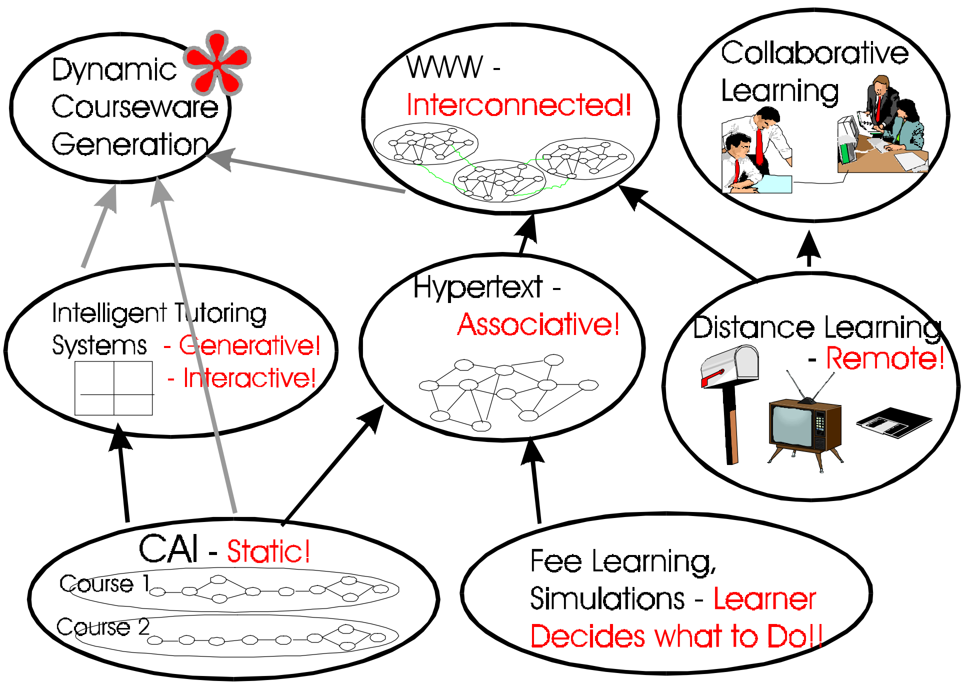

Figure 1: History and Main Approaches for CBL

1.1 Computer Assisted Instruction (CAI)

Computer Assisted Instruction (CAI) is the historically first approach which appeared in the 60's and follows the metaphor of an electronic book (Nievergelt, 1975). The majority of CAI programs present teaching materials as a linear sequence of screens consisting of text and graphics, allowing interaction with the student by means of asking pre-defined questions (drill and practice) and checking the correctness of the answer by comparing with a pre-defined sample by the author of the course. When the answer is not correct, a small deviation of the line of the course follows, where the correct answer is shown. With the emergence of personal computers, CAI programs have become enormously popular - such programs can be nowadays bought in every bookstore. They accompany complex software products to provide an introduction to their features, instead of a "Getting started" manual. The modern CAI involves multimedia presentations, which makes them extremely valuable as supplementary materials for school instruction. The disadvantage of CAI is that though they are intended as tools for personalized instruction, they provide a very limited individualization. The learner has very limited control and freedom. Despite the illusion of interaction, CAI has a pre-defined contents like a textbook, written by the author with a given teaching goal and strategy and they can't be influenced by the learner. CAI cannot allow a student to behave in an unpredictable way. To do so would require a CAI system to be able to understand the student, the subject being taught, and various teaching strategies, among other things (McCalla, 1987). The issue of amount and quality of student control in CAI has been extensively discussed in literature (Steinberg, 1989). There is a strong evidence that different individuals prefer different degrees of control on their learning. In general, there is a tendency to reduce control and to support individualized retrieval of information.

To industrialize the creation of CAI courseware, powerful authoring tools have been created offering impressive graphical and animation capabilities, including video and multimedia. Unfortunately the cost of authoring courseware is still significant: A 1994 study of computer-based training (Reinhardt, 1995) found that the mean number of hours required to create a single hour of courseware was 228. At a conservative rate of $ 100 per hour, that works out to more than $ 20 000 for one hour of teaching. Finally, it pays back, especially for corporate training: as Steve Linsk, former product marketing manager for multimedia tools at Asymetrix (referred by Reinhardt, 1995) said, "it is still significantly less than transporting employees to a central location, putting them up in hotels, and then forfeiting their lost productivity - all to stuff their heads with information they will largely forget." For schools, however, it still doesn't pay back since they are not enough saturated with computers in order to ensure that many students will make use of the courseware.

Authoring tools are becoming more powerful and can be used by people without the need to have any programming knowledge, i.e. they can be widely used by teachers to prepare their own courses. Of course, it is not clear whether they will really make use of these tools. How many of the teachers make use of their writing abilities to write their own textbooks? It is more realistic to see teachers willing to adapt existing CAI-courses to their own preferences, adding new materials to already developed courses. If we want to see teachers taking a more active role, we have to provide them with help in the design of the course, to ensure that the course really covers all important concepts in the domain and achieves some goals. However, there is a significant shortage of tools for curricula development.

1.2 Free Learning Environments and Simulations

Microworlds are computer-based environments which offer a set of commands (tools) for creating and operating with objects and for investigating the properties of the created "world" consisting of these objects. Such environments are also called „open-ended“ learning environemnts to stress the fact that they provide no guidance or teaching, but just a set of tools and it is upto the student to decide what s/he will do with them, (just like creating models of houses or cars with LEGO-stones). Microworlds have been developed and successfully used by children in schools in the 70's and 80's, due to the increasing popularity of LOGO (Papert, 1980) which features "turtle" graphics. A LOGO-based environment for geometry (Sendov & Dicheva, 1988), (Sendov et al., 1989), for example, can help the student define formally figures and to see them being visualized on the screen. The student can investigate the relationships between the objects, or discover regularities (proportions) and formulate theorems from observing what happens with the figures when one varies different parameters. The results of many studies have shown that microworlds increase the student's motivation in acquiring the material. Students get braver and take initiative to investigate, to discover and formulate the problems, and to search for solution by experimenting. In this sense, knowledge is expected to emerge on a meta-cognitive level, though not explicitly taught or supported by the system.

Device and system simulations put students in control of the experiments they perform on the simulation model. This creates a feeling of realistic environment and enables "learning by doing". There are domains where the real object of teaching (phenomenon / equipment / experiment) is dangerous, expensive, or impossible to observe, for example, training surgeons for complex operations, training nuclear power - plant operators, air-traffic control and pilots for airplanes Teaching in these doamins may benefit greatly from employing simulations (Draman, 1992). Computer-based simulations are used successfully in many other fields where it is needed to train skills of diagnosing, analyzing, performing actions on technical devices. Observing the behavior of complex systems (e.g. economical) while varying one or more parameters and developing hypotheses about the dependencies among the components of these systems is again a teaching method which targets developing meta-cognitive skills, like the ability to form and refute hypotheses. In simulation-based microworlds as well as in open-ended learning environments in general, the system is passive: it provides no support or advice. These systems provide a set of tools or manipulation possibilities to the student, but they contain no educational knowledge in the form of scenario, curriculum, plan, or instructional strategy. The reason is that the goal of such systems is to promote the students to discover things and work independently.

1.3 Hypermedia

Hypertext and hypermedia present an alternative approach to the individualization of instruction. Knowledge is represented there as a set of independent documents which can be traversed by following associative references (for an excellent review, see Kuhlen, 1991 or Begoray, 1990). This technique can be compared to browsing through an encyclopaedia without really searching a specific item. Words that are explained by independent articles are highlighted in some way which serves as an explicit reference. Students are free to decide how to move from one to another document. In contrast with the linear control of the learning process performed by CAI, hypermedia systems offer non-linear spaces in large (possibly world-wide) network environments which support free browsing, communication and cooperation between individuals. Although a few pioneers had thought out the basic concepts much earlier, the idea of hypertext was not popular before the late 1980's (Conklin, 1987). Nowadays all big encyclopaedias are already available as hypermedia documents on CD-ROM. However, if the network of documents is too large (e.g. the WWW), the student can get "lost in the hyperspace". That is why navigational aids like global and local maps (Yankelovich et. al, 1988), guided tours (Stubenrauch, 1989), anchors (Palaniappan et. al, 1990), fish-eye views (Furnas, 1986) or more "intelligent" approaches like student-model based suggesting of items to be "visited" and disabling some of the links (Brusilovsky et. al, 1993), i.e. integrating the possibility for free browsing with a directive way of teaching.

1.4 WWW

WWW made it possible to create a globally distributed hypermedia system. Enormous amount of materials exist on WWW and a large part of them are appropriate as learning materials. The big potential of WWW for education is in the wide access to such materials, independent of geographical location and hardware platform. Of course, many problems arise concerning the different languages of the materials, different cultures and simply because of the huge amount of such materials. Also interactivity is not inherent for hypermedia; if interactive teaching materials for the WWW have to be developed, the authors have to design explicitly this interactivity, for example, by means of JAVA programs. However, new authoring tools for the WWW are appearing all the time and the development in this field is so rapid that we can expect that the problem with interactivity and authoring will not be so acute in the near future.

1.5 Intelligent Teaching Systems

An attempt to apply techniques from the field of Artificial Intelligence in order to obtain a qualitatively different degree of individualization led to establishing the field of "AI and Education" (Clancey & Soloway, 1990), (Kaplan & Rock, 1995). An ideal Intelligent Teaching System (ITS) — we use this term rather than Intelligent Learning Environments, Intelligent Tutoring Systems, Intelligent CAI etc. since we believe it is general enough to cover all these — "knows" the subject which it is teaching. That means that it possesses a model of the domain knowledge: either an encapsulated "black box" model like in simulations, or explicit expert knowledge represented in some form (e.g. rules in Anderson et. al, 1990). By use of this knowledge, the system is able to reason about the domain, to solve problems, to generate plans for action and is able to answer the student's questions and to explain what and why it is doing. It also possesses information about the student, his preferences, psychological features, level of knowledge or even an explicit model of his knowledge and misconceptions. Some advanced student modeling techniques have been developed to diagnose and represent the student's metacognitive abilities. An ITS has also explicitly represented pedagogical knowledge (teaching strategies) which helps the system decide how to optimally present the teaching material, how to sequence the interactions, how to react and manage the initiative in the dialogue. The goal of ITS is teaching not mere facts, but understanding; not ready solutions, but methods for problem solving. Typically ITSs perform three styles of teaching:

- *tutoring*: this style is more directive, and presentation oriented, like CAI. However, the selection and sequencing of the teaching materials is not prescribed, but done by explicitly represented generic teaching strategies taking into account the individual student model. In this way the system generates an individualised path through the learning material for every student.
- *Socratic dialogue:* the system involves the student in a dialogue, and causes the student to discover his misconceptions by asking questions which lead to controversies with his existing beliefs.
- *coaching:* this style is characterised with the passive role of the system, which stays in the background, observes and evaluates the student's performance and is always available to provide help, to criticise the solution and to interfere when necessary.

A synthesis of Tutoring, Coaching and Socratic dialogue can implement a return to a very old educational model: apprenticeship. The apprenticeship model provides encouragement and instruction when needed and lets the "master" (teacher) fade back when the student has reached an acceptable level of performance. It is extremely difficult to mix interactively these three styles of learning and to give the student the freedom to explore the material and try on his own how to solve the problems. We have developed a framework for reactive instructional planning (Vassileva, 1992, 1995) which allows such an integration and we are currently investigating the didactic issues involved in taking pedagogical decisions to support such a mixed initiative environment.

An ITS which is equipped with a simulation of the domain and performs a coaching style of teaching is an extremely powerful learning tool, especially in fields where it is hard or impossible to do apprenticeship in real life, like surgery or learning to fly an airplane. The majority of existing ITSs are coaches based on expert systems, microworlds or computer simulators.

The marriage of multimedia to ITS offers the potential to create intelligent systems that can instruct and demonstrate, using sound, animation and video. It is not needed to encode the media into the tutoring sequence, an explicit knowledge representation allows reasoning about the domain and dynamic access to the appropriate media at the moment it is needed.

1.6 Distance Learning

Distance learning has developed quite early and independently of CAL. In many countries Distance-, Open- and TV-universities have been developed in the 70-ies, which distributed learning materials by correspondence or broadcasting. CAL materials were often used, since unlike printed materials and broadcasts, they provided the possibility for interaction.

1.7 Collaboration Environments

Collaboration environments became possible with the development of networking and its application to distance learning. It became possible to form “virtual classrooms” providing the learners with the opportunity not only to see a life session of a lecture, but to ask the tutor questions which can be answered in real time. Soon the bigger potential of such virtual classrooms was discovered. The fact that individual cognitive development is influenced by social interactions has been studied by a number of researchers in different areas. In his work "Mind in Society" (1978), which became the basic reference in constructivist literature, Vygotsky hypothesizes that social interactions play a fundamental role in shaping internal cognitive structures. Experiments have also shown that two children working together can successfully perform a task which cannot be performed by children of the same age working alone. In general, a peer of the same level of a student serves two possible roles for the student in learning (Chan et al., 1992): a collaborator and a competitor. As a collaborator a peer provides cognitive conflict and scaffolding. The concept of scaffolding (Bruner, 1975) captures the contribution a child (or an adult) can make to another child's learning by observing his behavior, providing hints, guidance or advice, as well as feedback, correction or evaluation. Scaffolding involves a kind of cooperative problem solving effort by both learners. As a competitor, a peer is a source of motivation. It is certainly desirable for a learning environment to be able to enhance a student's achievement-striving behavior. However, competition can have a negative effect on some learners who are inhibited rather than motivated in a competitive situation (Collins et. al, 1989).

Many collaborative environments have been developed since the beginning of the 90's, starting with Chan & Baskin's (1990) "Integration kid", a system that allows two children to work collaboratively, compare solutions, discover mistakes and self-correct the mistakes in their own solutions. If there remain mistakes not discovered, the teacher points them out. Later collaborative environments take advantage of networking and allow collaboration among a whole class of students (Hoppe, 1995) where the system suggests a peer for the student according to his level of progress (student model) for solving the current problem. Impressive multimedia systems integrating cognitive apprenticeship and a simulation based collaborative environment have been developed (Katz & Lesgold, 1994), (Duncan et al., 1995). These systems make use of the booming research and industrial applications in Computer Supported Collaborative Work (CSCW) systems which is a field of high commercial interest, esp. for companies.

1.8 Towards Integration

A tendency of integrating all above mentioned paradigms can be seen in the last two years. One direction is "intellectualizing" of "unintelligent" paradigms, like CAI, hypermedia (Brusilovsky et al., 1993), simulation-based environments and microworlds, collaboration environments (Katz, 1995). Another clear direction is scaling up all applications over local and large area networks, including Internet (Andrews et al, 1995; Bouras et al., 1995; Chee, 1995). A rapidly increasing amount of experience with virtual reality systems comes to join the discussed technologies in their efforts for making students have realistic experiences (Carpenter-Smith et al., 1995).

Our work on dynamic courseware generation on the WWW is a step towards the integration of the state-of-the-art paradigms in CBL. It will be described in the next sections.

2. DCG on the WWW: Integration of CAL, ITS, Distance and Collaborative Learning Systems

With the emergence of the WWW it became possible to provide learners with unlimited access to teaching materials. Authoring became easier since to produce such materials (HTML-files) one can already use common text-editors. However, learners in the WWW face one significant problem: they have to cope with a huge amount of materials and to navigate through links which sometimes are not relevant to the learner's goal. Often they get lost in the hyperspace, forget where they started and loose a focused perspective over the field.

CAI provides one simple solution to the problem by limiting the browsing activity of the learner to a pre-defined sequence of teaching materials (course). This solution is, however, quite inflexible to accommodate various learner's goals and success in acquiring the material. This problem can be solved by providing learners (clients) with access to a tool for automatic generation of adaptive courses on a server.

We have developed a tool, called "Dynamic Courseware Generator" (DCG), for automatic generation of individualized courses according to the learner's goal and previous knowledge which can dynamically adapt the course according to the learner's success in acquiring knowledge (Diessel, Lehmann, Vassileva, 1994), (Vassileva, 1992, 1995). It has been tested in several domains: the structure and functioning of a toasters and transistors, medical diagnosis and jazz. The DCG allows:

- automatically assembling CAI courses with different goals from existing Teaching Materials (TMs);
- creating different courses for students with different knowledge and personal characteristics;
- changing the course dynamically according to the progress of the student.

We have recently implemented the DCG on a WWW-server and now it can be used for domain-authoring and automatic generation of adaptive courses on the WWW.

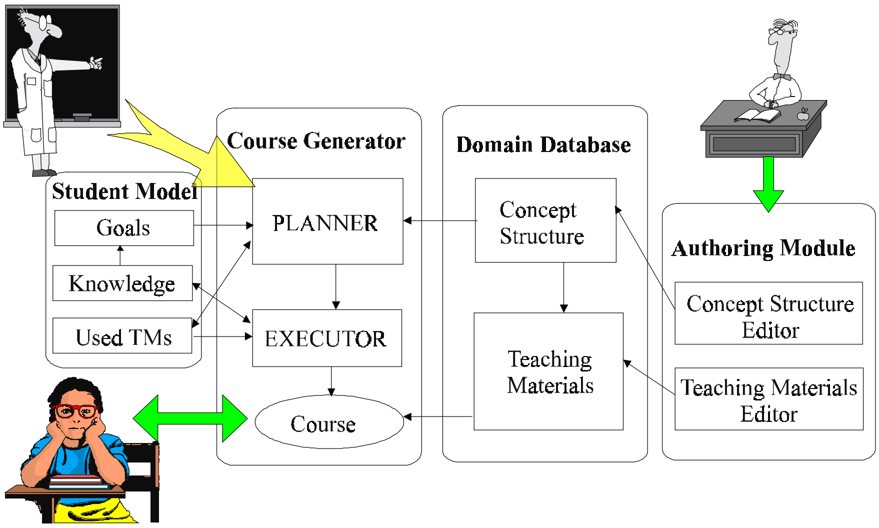

Figure 2: Architecture of the DCG

3. The DCG on the WWW

The main idea of the DCG architecture is the explicit representation of the concept structure of the domain, separated from the teaching materials which the student sees on the screen (see Figure 2). The concept structure is used by the system as a road-map to generate a plan of the course. Given a certain goal-concept that the learner wants to acquire and a student model containing the concepts already known by the student (initialized with a pre-test), the planner searches for sub-graphs that connect the concepts known by the learner with the goal-concept. One of these plans is selected and offered to the learner to follow. The learner sees a sequence of teaching materials (TMs) related to each concept from the plan. At every point the learner can wish to be tested on his/her knowledge of the current concept and s/he will be given a set of test-items. A student model is created for every learner which is an overlay over the concept structure and uses a simple numeric heuristic to calculate the score of the learner's knowledge of every tested concept. If the learner is not able to achieve the needed score for a given concept, needed to proceed further with the plan, a new planning takes place.

3.1 Overall architecture

We adopted a client-server architecture for the system on the WWW (see Figure 3). The DCG is placed on a WWW-server; it offers teaching and authoring services. A client can be every WWW-browser. Learners and authors /teachers can be clients and use the teaching and authoring services offered by the DCG-server. The teaching materials used in the courses generated by the system are also distributed on various sites in the WWW.

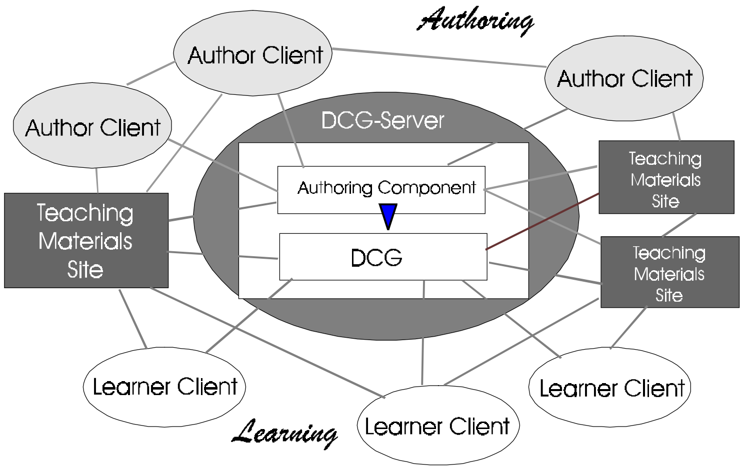

Figure 3: Client-Server Architecture of the DCG on the WWW.

3.2 Authoring with the DCG on the WWW

The authors can also be distributed over the net on their sites. Authoring comprises three stages (see Figure 4):

1. creating/modifying a domain structure for a given domain,
2. providing links from each concept / relation to appropriate html-files on the WWW that can be used as TMs for this concept,
3. loading the domain on the DCG server.

A special graphical editor for concept structures allows the creation of concepts, inter-relating them with various types of semantic relations (e.g. abstraction, aggregation, analogy, temporal, causal, etc.). One-directional 1:n - and n:1 - relations are used to represent hierarchies with respect to abstraction, aggregation and causal relations, and bi-directional 1:1 - relations to represent analogical and temporal relations.

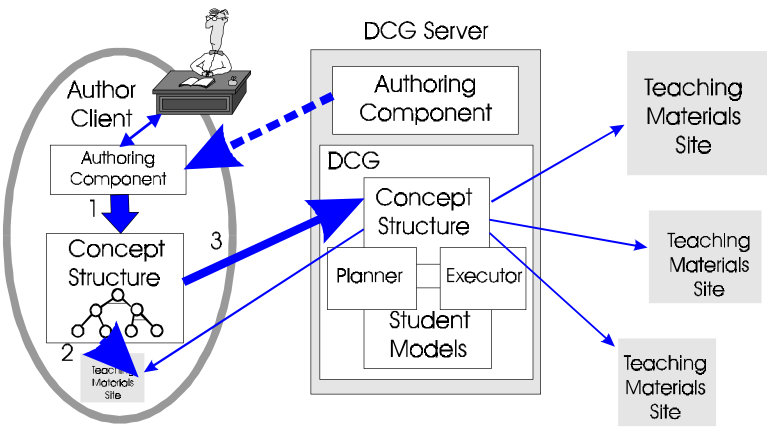

Figure 4: Authoring.

Every concept and relation can be linked to the http-addresses of one or more teaching materials (represented as html-files) and testing materials (html-files with an attached JAVA procedure which carries out the interaction and answer-evaluation). One teaching material can address one or more concepts/relations and one concept/relation can be presented with one or more teaching materials. For the creation of the teaching and testing materials the author can use every html editor or reuse existing html-sources. The creation of testing materials requires only a link to the JAVA procedure and assigning the right answer to the question.

3.3 Learning with the DCG on the WWW

The learner sends a request for a course to the DCG-site stating the desired domain and learning goal. The DCG sends them a preliminary test to initialize a student model for each learner and enters him/her as a user. Then the learner receives an individual course-plan and a copy of a JAVA program, called "Executor" which runs on the learner's site. The Executor tests the student's knowledge on the concepts and updates the student model locally at the learner's site. In case of student's inability to perform successfully on the test related to a certain concept that is needed for continuing with the plan, the Executor contacts the DCG site; it sends a copy of the Student model and requests from the Planner a new course-plan, appropriate for the new state of knowledge of the student. The learning process is presented in Figure 5.

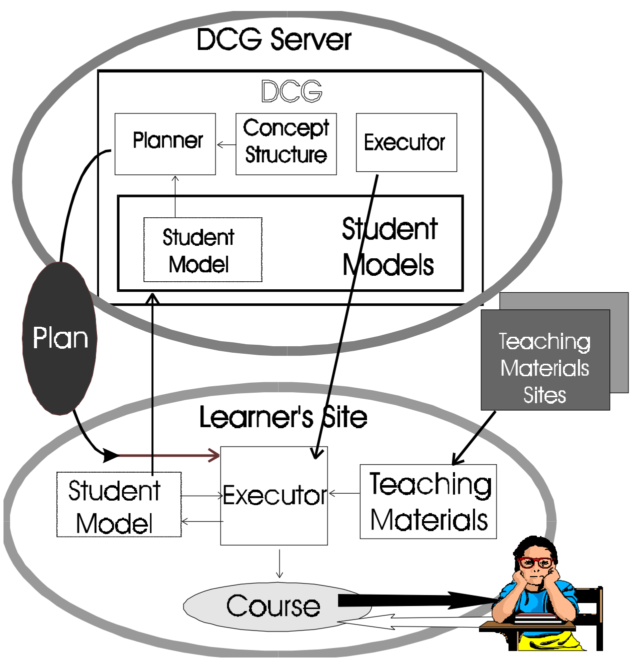

Figure 5: Learning with the DCG.

3.4 Components of the DCG on the WWW

As mentioned before, because of the client-server architecture adopted, some of the components of the DCG are kept only on the DCG server (the Domain Concept Structure and the Planner), some exist both on the DCG server and the client (the Student Model and the Executor) and some are spread throughout the various WWW sites (the Teaching and Testing Materials).

<u>The Domain Concept Structure</u>

The structure of the domain concepts is kept on the DCG server. It is represented as an AND-OR graph consisting of the domain concepts connected with relations with different semantics (aggregation, abstraction, causal, analogical, simple prerequisite etc.). For different domains different semantic relations are important, for example, the most often type of relation in technical domains is aggregation. However, in procedural domains causal and prerequisite relations are more important. Planning can take place with respect to one or several types of relations, depending on the learning goal. A learning goal is defined as a goal-concept and types of relations with respect to which the plan should be made. The system offers for every domain a set of possible meaningful goals in a menu from which the learner can select.

<u>The Teaching and Testing Materials</u>

The teaching materials are html-files which can be distributed on different sites in the WWW. At authoring stage http-links from the concepts to the desired html-files are provided. Test-items should also be defined for every concept with an indication of their difficulty and a coefficient, showing how much a correct and how much a wrong answer to the test contributes to the overall score of the related concept/link in the student model.

<u>Student Model</u>

There are two instances of the Student Model. One is very dynamic and exists on the client (learner's) site. A copy of this instance is stored on the DCG-server every time the student ends a session or re-planning is required. It is used for taking down statistics of students success with different concepts which is useful for improving the teaching materials and domain concept structure (e.g. decomposing complex concepts). The student model can be stored anonymously on the DCG site, if the student wishes so. However, in this case the information from previous sessions about the student can not be used and every time the learner wants to learn for a new goal, a new pre-test has to be done. The current state of student knowledge is represented as an overlay with the concept structure in the data base, containing the system's estimations that the learner knows a certain concept. These estimation is built using a simple formula from the number and difficulty of successfully solved tests-items related to this concept.

Planner

On basis of the learner's goal and the model of the student's current knowledge, the Planner generates a plan of the course. The plan is an AND-OR graph, i.e. a sub-set of the concept structure consisting of concepts and relations that have to be taught during the course.

Executor

The Plan Executor is downloaded at the learner's site in the beginning of the session. It takes the plan as input and generates a course by searching in the WWW for the appropriate teaching materials (html-files linked to the concept), loading them at the learner's site and presenting them to the student. When the learner wishes to be tested on his/her knowledge of the concept, the Executor selects a test-item, presents it to the student, accepts and evaluates his/her answer and updates correspondingly the Student model. It can re-invoke the Planner to create a new plan, if the level of knowledge of a concept, which is crucial for the success of the plan, is not satisfactory. In this case, it sends the contents of the Student model to the DCG site.

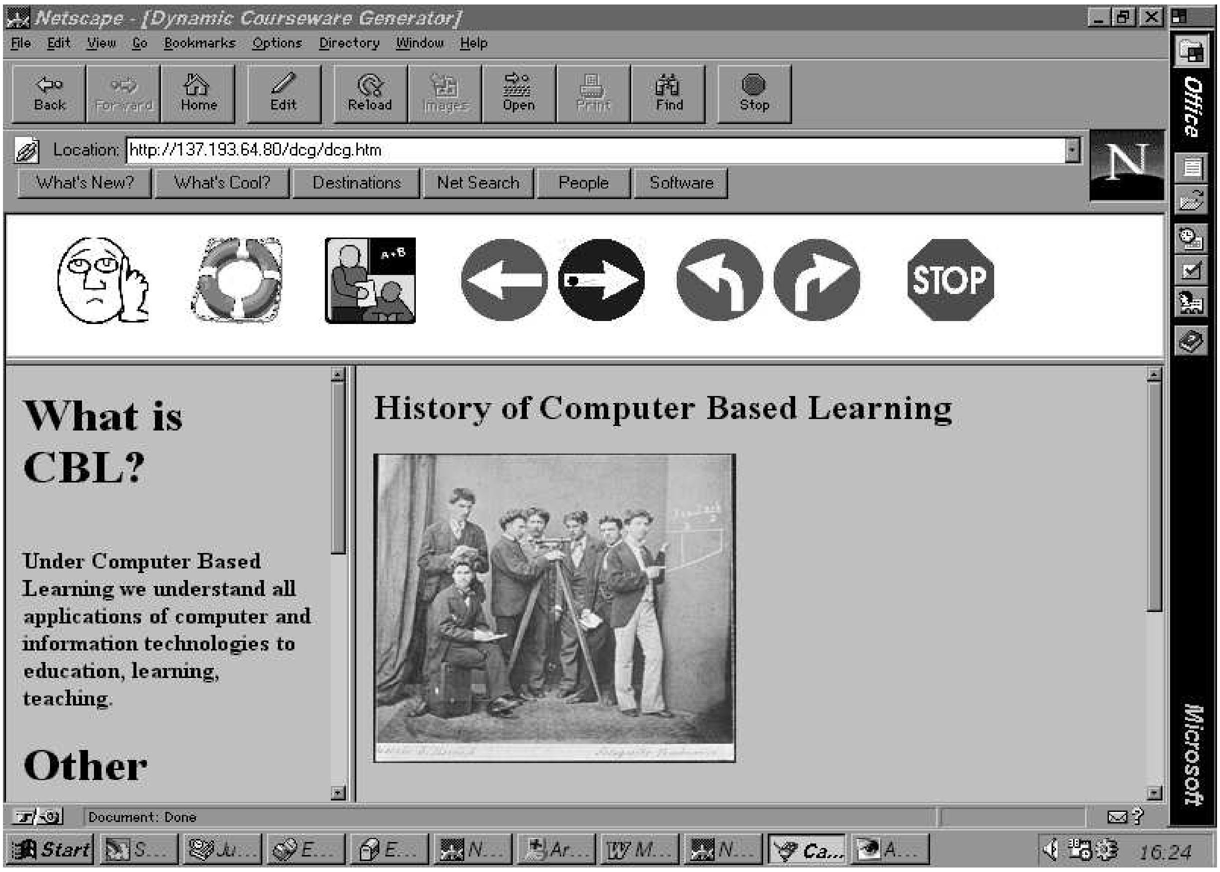

Figure 6: The DCG-Interface for the Learner

3.5 Evaluating and Using the DCG on the WWW

The DCG has been already evaluated as a stand-alone authoring tool in several domains and its advantages have been shown (Vassileva, 1994, 1995) quite clearly (18 hours of authoring for 1 hour of instruction). The DCG on the WWW has been experimentally implemented in the domain of Computer Based Learning Systems (see Figure 6). The goal of the implementation was to provide an example and to catalyze ideas about how such a system could be integrated in the teaching process the University. We do not expect that the DCG on WWW will make a „breakthrough“ in the paradigm of university teaching, but it can be easily and usefully integrated in the existing organization. The following main applications were outlined:

- *Lecture Support, Distance and Continuous Education*. The courses generated with the DCG can be used as additional learning materials (as an interactive script) supporting lectures given regularly or occasionally at the University. The specifics of our university is that the students are officers who are obliged to serve in the Army five years after graduating. Interactive courses on the WWW accompanying the basic and specialized lectures offered at the university would provide a „umbilical cord“ between our students and their Alma Mater. They will have the possibility to deepen and actualize their knowledge permanently.

- *Re-use and Sharing of Domains*. The distributed architecture of the DCG allows for authors to collaborate and cooperate in editing domain structures and relating TMs to the concepts. It also allows a reuse of TMs and domain structures. Libraries of often used "basic" concepts can be developed. An electronic domain represented in the DCG, linked to actual documents on the WWW, such as "hot" scientific papers, or just textbook explanations can be shared by lecturers teaching the same subject at different universities, where everyone can make extensions and modifications according to his/her personal view.

- *Learners as Authors*. Modern learning theories point out the positive effects of the learner feeling "ownership" of the problem, and to the effects of explanation and self-explanation over the development of reflective skills and own (not necessarily correct) knowledge structures. For this reason often students are left to plan and design a lesson themselves (e.g., by organizing the lectures as seminars presented by students). This enables them to create an own, independent view of the domain, ability to search for new information, to develop knowledge structures and to integrate them within already existing ones. The DCG as an authoring tool can be used for carrying out this type of projects. For example, a student or a team can be assigned the task of authoring a certain theme (subdomain). The student/ team has to review the literature, to discover important concepts and relationships and create a domain structure, to create or find related materials on the WWW and to link them to the concepts in the domain structure. The structure and materials will be discussed and criticized by the lecturer and the class, and the domain produced in this way can be used later by the DCG for automatic generation of instructional courses on this theme.

We believe that ITS-authoring tools with distributed architectures like the DCG can provide a good start towards WWW-based adaptive educational systems.

REFERENCES

Anderson, J., Boyle, C., Corbett, A. and Lewis, M. (1990) Cognitive modelling and intelligent tutoring, *Artificial Intelligence*, 42, 7-49.

Andrews K., Nedoumov A., Scherbakov N. (1995) Embedding Courseware into the Internet: Problems and Solutions, in Proceedings ED-MEDIA'95, Graz, 69-74.

Begoray, J. (1990) An introduction to hypermedia issues, systems and application areas, International Journal of Man-Machine Studies, vol. 33, 121-147.

Bouras C., Fotakis, D., Kapoulas, V, Lampas P, Papoutsopoulos G. (1995) HIPPOCRATES: A multimedia tool for distance education, in Proceedings ED-MEDIA'95, Graz, 103-108.

Bruner, J. (1975) The Ontogenesis of Speech Acts, Journal of Child Language, vol.2, 1-19.

Brusilovsky P., Pesin L., Zyryanov M. (1993) Towards an adaptive hypermedia component for an intelligent learning environment. In Bass L.J., Gornostaev J. and Unger C. (eds.) Human-Computer Interaction. Lecture Notes in Computer Science 753, Springer-Verlag, Berlin, 1993, 348-358.

Carpenter-Smith T., Holyoak D., Olson, K. (1995) Integrated Virtual Reality System for Learning Object Oriented Technology, in Proceedings ED-MEDIA'95, Graz, p. 745.

Chan T.W., Baskin, (1990) Studying with the Prince, Proceedings ITS-88, Montreal, 194-199.

Chan, T.W., I.L. Chung, R.G. Ho, W.J. How, (1992) Distributed Learning Companion System: WEST Revisited, in Proceedings ITS-92, Montreal, Lecture Notes In Computer Science No 608. Springer Verlag.

Chee, Y. (1995) MINDBRIDGES: A distributed multimedia learning environment to support collaborative knowledge construction, in Proceedings ED-MEDIA'95, Graz, 151-156.

Clancey, W. & Soloway, E. (1990) AI and Learning Environments: Preface to the Special Issue of J. of Artificial Intelligence on AI and Education, vol. 42, no 1, 1-6.

Collins A., Brown, J.S: & Newman, S. (1989) Cognitive apprenticeship: teaching the craft of reading, writing and mathematics, in Resnick, L. (ed.) Knowing, learning and instruction: Essays in honour of Robert Glaser, Lawrence Erlbaum Assoc.: Hillsdale, NJ.

Conklin, J. (1987) Hypertext: An Introduction and Survey, IEEE Computer, vol. 20, no. 9., 17-41.

Diessel Th., Lehmann A., Vassileva J. (1994) Individualized Course Generation: A Marriage Between CAL and ICAL. Computers & Education, Vol. 22, No.1/2, pp.57-64.

Draman, M. (1992) Simulation and Training in Education: Evolution towards Generic Intelligent Systems. Kaylan, R. Ören, T. (eds.) Proceedings of the Advances in Simulation ' 92 Symposium.

Duncan, P., Cannon-Bowers, J., Johnston, J., Salas, E. (1995) Using a Simulated Team to Model Teamwork Skills: The Team Model Trainer, in Proceedings ED-MEDIA'95, Graz, 187-192.

Furnas, G. (1986) Generalized Fisheye Views, in Proceedings of CHI'86, New York: ACM, 16-23.

Hoppe, H.-U. (1995) The Role of Multiple Student Modeling to Parametrize Group Learning, in Proceedings of the 7. World Conference on AI and Education, Washington, AACE.

Kaplan, R. & Rock, D. (1995) New Directions for Intelligent Tutoring, AI Expert, February 1995, 31- 40.

Katz, S. (1995) Towards a research methodology for developing truly supportive computer supported collaborative learning environments, in Proceedings ED-MEDIA'95, Graz.

Katz, S. and A. Lesgold (1994) Implementing Post-Problem Reflection within Coached Practice Environments, in Brussilovsky, P. (ed.) Proceedings of the East-West International Conference on Computer Technologies in Education. Crimea, Ukraine, September 1994.

Kuhlen, R. (1991) Hypertext - ein nicht lineares Medium zwischen Buch und Wissensbank, Springer Verlag.

McCalla, G. (1987) The Promise of AI for Computer Assisted Learning, in Proceedings ICCAL in Post-Secondary Education "Learning in Future Education".

Nievergelt, J. (1975) Interactive Systems for Education - the New Look of CAI, in Proceedings of the 1975 IFIP Conference on Computers in Education, Lecarme O. and Lewis, R. (eds.), North-Holland: Amsterdam, 465-471.

Palaniappan, M., Yankelovich, N., Sawtelle, M. (1990) Linking Active Anchors: A Stage in the Evolution of Hypermedia, Hypermedia, vol.2, no.1, 47-66.

Papert, S. (1980) Mindstorms: Children, Computers and Powerful Ideas, Harvester Press, Brighton.

Reinhardt A. (1995) New Ways to Learn, Cover Story, Byte, March 1995, 50-71.

Sendov Boj., Dicheva D. (1988) A Mathematical Laboratory in Logo Style, in: Fr. Lovis and E.D. Tagg (Eds.), Computers in Education, IFIP, ECCE'88, North-Holland, Amsterdam, 213-217.

Sendov Boj., Filimonov R., Dicheva D. (1987) A System for Teaching Plane Geometry. Proceedings of the Second International Conference "Children in the Information Age", Sofia, 215-226.

Steinberg, E. (1989) Cognition and Learner Control: A Literature Review 1977-1988, J. CBI, Vol. 16, no. 4, 117-121.

Stubenrauch, R. (1989) Touring a Hyper-CAI System, in Computer Assisted Learning, Proceedings of the 2nd International Conference ICCAL'89, Dallas, May 1989, Lecture Notes In Computer Science 360, Springer, 541-551.

Stubenrauch, R. (1992) Aspects of CAI and Hypermedia, Ph.D. Thesis, Graz University of Technology, Graz.

Vassileva J. (1992) Dynamic Courseware Generation within an ITS-shell Architecture. Proceedings ICCAL'92, Lecture Notes in Computer Science No 602, Springer: Berlin-Heidelberg, 581-591.

Vassileva J. (1995) Dynamic Courseware Generation: at the Cross Point of CAL, ITS and Authoring. in Proceedings of ICCE'95 - International Conference on Computers in Education, Singapore, 5-8 December 1995, 290-297.

Vassileva J. (1995) Reactive Instructional Planning to Support Interacting Teaching Strategies, Proceedings of the 7-th World Conference on AI and Education, Washington, August 16-19, AACE.

Vassileva, J. (1994) A New Approach to Authoring of Adaptive Courseware for Engineering Domains, in Proceedings CALISCE'94, Paris, 241-248.

Vygotsky, L. (1978) Mind in Society: The Development of Higher Psychological Processes, Harvard University Press, Cambridge, MA.

Yankelovich, N, Haan, B., Meyerowitz, N. Drucker, S. (1988) Intermedia: The Concept and Construction of a Seamless Information Environment, IEEE Computer, Vol. 21, no.1, 81-86.