

Dynamic Courseware Generation on the WWW*

Julita Vassileva and Ralph Deters
Federal Armed Forces University Munich
Institute for Technical Computer Science
85577 Neubiberg, Germany
[jiv, deters]@informatik.unibw-muenchen.de

Abstract

We have developed a tool for the authoring of adaptive CAL courses, called "Dynamic Courseware Generator" (DCG). It generates an individual course according to the learner's goals and previous knowledge and dynamically adapts the course according to the learner's success in acquiring knowledge. The DCG runs on a WWW server. The learner receives from this server an individualized course targeted to a specified goal. Afterwards, s/he is adaptively guided by the course through a space of teaching materials on the WWW. Unlike other CAL courses on the WWW, a course produced by the DCG is interactive, it tests the learner's knowledge and dynamically adapts to the student's progress. The authoring tool can be used also for collaborative authoring and learning.

1 Introduction

With the emergence of the WWW it became possible to provide learners with unlimited access to teaching materials. Authoring became easy 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 their learning goals. Often they get lost in hyperspace, forget where they started and lose a focused perspective on the field. Various approaches exist for supporting the user navigation when learning from the WWW. Most of them use a secondary pedagogical structure of concepts connected with prerequisite relationships which overlays the hyper-link structure of the documents (Nykänen, 1997). For example, AST (Specht et al, 1997) uses prerequisite structure of concepts to suggest which concepts should be learned first in order to understand the teaching goal-concept. Similarly, ELM-ART (Brusilovsky et al, 1996) annotates the links from the current concept which are "ready to be learned" or "not ready to be learned" yet. These approaches provide a type of navigation support called *local orientation support* (Brusilovsky, 1996).

* Published in *British Journal of Educational Technologies* (1998), **29** (1), 5-14.

We believe that often *adaptive guidance* through the material is better suited to the needs of learners than local orientation support, since it achieves a goal-based, focussed and coherent way of teaching which is suitable when time is limited and learners are motivated. CAI provides this type of teaching -- a traditional CAI course is a pre-defined sequence of teaching materials (some of which can be interactive) aimed at achieving some teaching goal. A CAI course is designed always having in mind a certain learner population, which is assumed to possess already a certain background knowledge. This solution is, however, quite inflexible to accommodate the wide variety of learners on the WWW, with their possible learning goals, background knowledge and differences in the way of acquiring the material. This problem can be solved by providing learners on the WWW with access to a tool for automatic generation of individualized adaptive courses.

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 the concepts from the plan (Diessel et al, 1994), (Vassileva, 1992, 1995). It has been tested in several domains: the structure and functioning of electrical devices, medical diagnosis, training motoric skills (typewriting) and arts (jazz). The DCG allows:

- automatically assembling CAL courses with different goals from a pool of teaching materials;
- creating different courses for learners with different knowledge;
- changing the course dynamically according to the progress of the learner.

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.

The paper is organized as follows. Section 2 presents in main lines the idea and functioning of the DCG on WWW. Section 3 describes the architecture of the system in more detail. Section 4 presents briefly an experimental implementation and concludes with outlining the possibilities for integrating the DCG on WWW in university teaching.

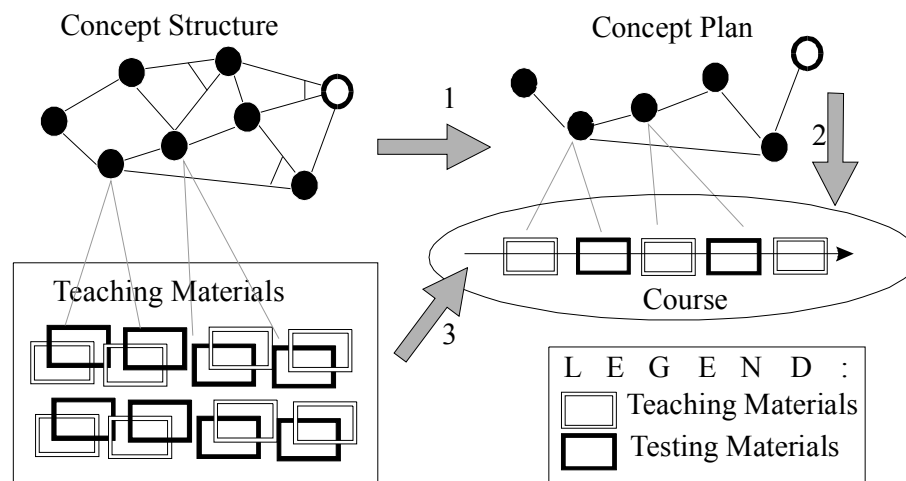


Figure 1. Main Idea of Dynamic Courseware Generation.

2 DCG on the WWW

2.1 Main Idea and Architecture of DCG

The main idea of the DCG is applying AI planning techniques to create a concept (content) plan of a course which achieves a certain goal concept starting from some set of initial concepts. For this purpose the DCG uses an explicit representation of the concept structure of the domain, which is kept separately from the teaching materials. The concept structure is represented as an AND/OR graph where nodes represent concepts and arcs correspond to relationships between the concepts. Given a certain goal-concept that the learner wants to acquire and a student model containing the concepts already known by the learner (initialized with a pre-test), an AI-planner program searches for plans (paths in the graph) that connect the concepts known by the learner with the goal-concept (see Figure 1). One of these plans is selected and is used as a skeleton of the course. For each concept, teaching materials are presented from the database of teaching materials and as a result, the learner sees a sequence of teaching and testing materials, which looks for him / her exactly like a CAI-course. The learner is tested at every concept of the plan and his/her student model is updated. If the learner is not able to achieve the needed score for a concept that is needed to proceed further with the plan, a new planning takes place to find an alternative plan for achieving the learning goal. The architecture of the DCG is presented in Figure 2.

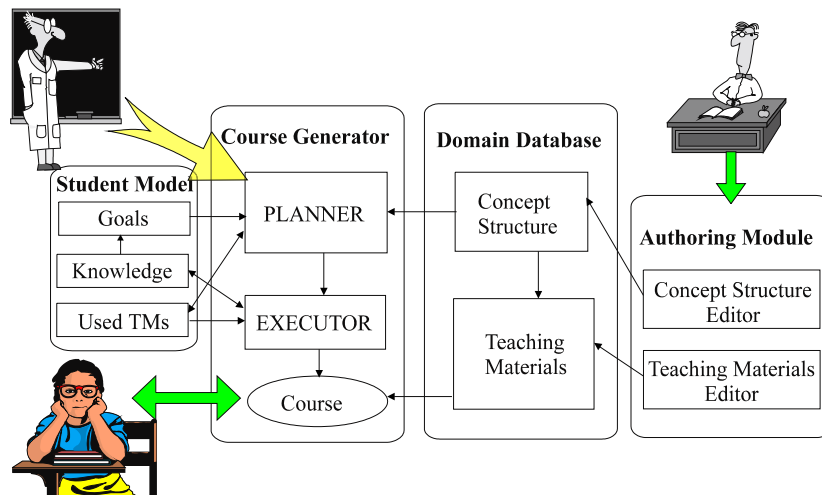


Figure 2: Architecture of the DCG.

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

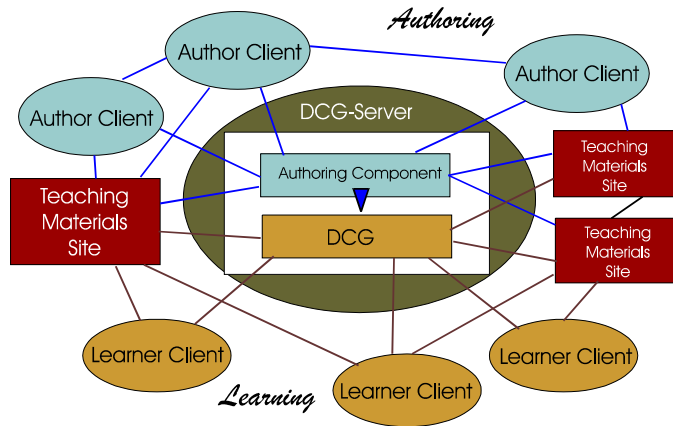


Figure 3: Client-Server Architecture of the DCG on the WWW.

2.2 Authoring with the DCG on the WWW

Authoring consists of two parts (see Figure 4):

- creating/modifying a concept structure for a given domain,
- providing links from each concept / relation to appropriate html-files on the WWW that can be used as teaching materials for this concept.

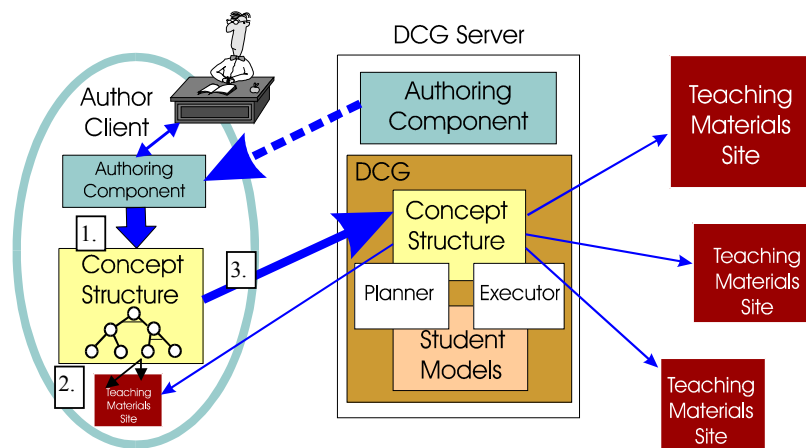


Figure 4: Authoring with DCG on WWW.

A special graphical editor for concept structures allows the representation of concepts (see Figure 5), inter-relating them with various types of semantic relations (e.g. abstraction, aggregation, analogy, temporal, causal, etc.). A set of "standard" links: the five types mentioned above is provided, but the author can define new ones, if he/she believes they are appropriate for structuring the domain. For this purpose he/she has to give a name

to the relation, and define the graphic properties for visualization of the relation in the AND/OR-graph representing the concept-structure.

The screenshot shows a dialog box titled "Knoteneigenschaften" with a close button (X) in the top right corner. The dialog is divided into several sections:

- Name:** A text field containing "Knoten 5".
- Beschreibung:** An empty text area.
- Aspekt:** A dropdown menu currently showing "Hauptaspekt".
- Kostenstatistiken:** A section with two columns: "Anzahl:" and "Kosten:".

	Anzahl:	Kosten:
bei Erfolg	0	0
bei Mißerfolg	0	0
- Beziehungen:** A section with a label "Gerichtet : Bestandteil" and a list box containing:
 - Von: - Knoten 1
 - Zu: - Knoten 2
 - Knoten 3
 - Knoten 4

At the bottom of the dialog, there are three buttons: "ULM anhängen" (with a green checkmark icon), "Neuer Aspekt" (with a yellow dashed border), and a group of three buttons: "OK" (with a green checkmark icon), "Abbruch" (with a red X icon), and "Hilfe" (with a question mark icon).

Figure 5: Editing of Nodes.

Every concept and relation is 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/reasons and one concept/reason can be presented with one or more teaching materials. For the creation of the teaching and testing materials the author can use any html-editor or reuse existing html-sources. The creation of testing materials requires from the author to assign the text of the question / problem and the right answer using a template JAVA-procedure that carries out the interaction with the learner and the answer evaluation.

2.3 Learning with the DCG

The learner sends a request for a course to the DCG-site stating the desired domain and learning goal. The DCG sends him / her a preliminary test to initialize his / her student model 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 learner's knowledge on the concepts and updates the student model locally at the learner's site. In case of the learner'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 there a copy of the Student Model and requests from the Planner a new course-plan, appropriate for the new state of knowledge of the learner. The learning process is presented in Figure 6.

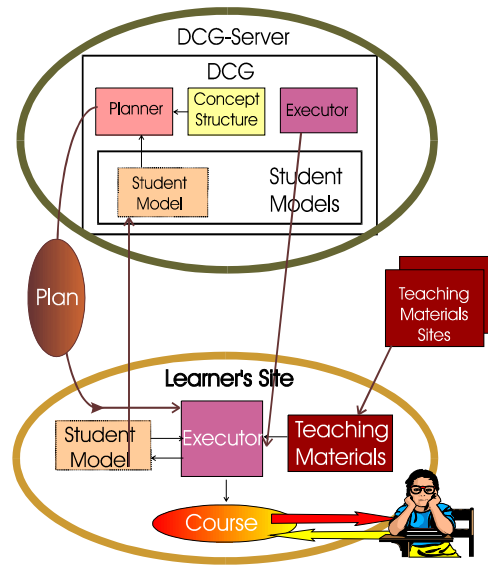


Figure 6: Learning with the DCG on WWW.

3 Components of the DCG on the WWW

As mentioned before, because of the client-server architecture adopted, some of the components of the DCG (see Figure 2) 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 Materials).

3.1 The Domain Concept Structure

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.). 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. For different domains different semantic relations are important, for example, in technical domain the most often occurring type of relation is aggregation. 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 by pointing to a goal-concept and assigning the types of relations with respect to which the plan should be made. For every domain the system offers a set of possible meaningful goals in a menu from which the learner can select. A major difference between the DCG and other approaches for adaptive navigation support (Nykänen, 1997), (Brusilovsky et al, 1996), (Specht et al, 1997), (Nakabayashi et al, 1997) lays in the higher expressive power of AND/OR graphs which allow representing not only prerequisite, but many different types of relations between concepts. This allows the use of AI-planning techniques to generate alternative courses according to links with different

semantic and in this way ensures a wide variety of different teaching goals and possible courses for achieving these goals.

3.2 The Teaching and Testing Materials

The teaching materials are html-files which can be distributed on different sites in the WWW. At the 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 / wrong answer contributes to the overall score of the concept/link in the student model.

3.3 Student Model

There are two instances of the Student Model. One is dynamic and exists on the client (learner's) site. A copy of this instance is stored on the DCG-server every time the learner ends a session or re-planning is required. It is used for recording statistics of learner 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 learner wishes so. The current state of learner 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. An estimation is built using a simple formula from the number and difficulty of successfully solved test-items related to this concept.

3.4 Planner

On the basis of the AND/OR graph representing the domain concept structure, the learning goal (the goal concept and the type of relationships to be taken into account) and the set of initial nodes (the Student Model representing the learner's current knowledge), the Planner generates a plan of the course. The planning algorithm is a modification of a well known algorithm for graph planning AO* (Nilsson, 1980). The plan is represented as a sub-graph of the concept structure consisting of concepts and relations that have to be taught during the course. It is linearized from left to right and as result a list of concepts to be taught is obtained. A more pedagogically justified way of linearization by using teaching rules based on a teaching task-decomposition, described in (Vassileva, 1995), is going to be implemented in the DCG on WWW in the near future.

3.5 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 from it by searching in the WWW for the appropriate teaching materials (html-files linked to the concept), loading them on the learner's site and presenting them to the learner (see Figure 6). When the learner wishes to be tested on his/her knowledge of the concept, the Executor selects a test-item, presents it to the learner, accepts and evaluates his/her answer and correspondingly updates 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 back to the DCG site, so that it can be taken into account in the re-planning.

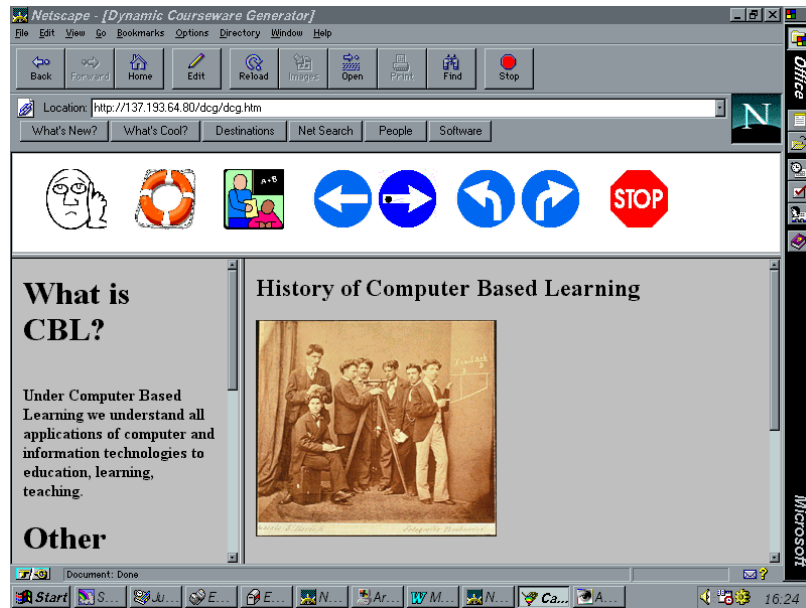


Figure 7: The Interface of the DCG.

4. Implementation and Prospective for Using the DCG on the WWW

The DCG has already been evaluated as a stand-alone authoring tool in several domains and its advantages have been shown (Vassileva, 1995). The DCG on the WWW has been experimentally implemented on three local servers (Netscape Servers and a Microsoft IIS) at the Federal Armed Forces University. The first domain which was created with the DCG covered the material of the first two lectures of a course on Computer Based Learning Systems (CBL) given by the first author at the University (see Figure 7). The goal of the experimental implementation of the DCG on WWW was to see how such system can be integrated in university teaching. DCG in WWW was experimentally used by twelve students taking the course of CBL. Our experience showed that the DCG on WWW 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 all 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 co-operate in editing domain concept structures and relating teaching materials to the concepts. It also allows a reuse of teaching materials and domain structures. Libraries of often used “basic” concepts can be developed. Domain representations in the DCG could be shared by lecturers teaching the same subject at different universities, where everyone can make his / her extensions and modifications and use the structures and materials for preparing lecture courses (also not computer-based) by recombining them 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 their own (not necessarily correct) knowledge structures. For this reason often students are left to plan and design a lesson themselves, for example, by organizing the lectures as seminars presented by students. This enables them to create their 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 (sub-domain). The student / team has to review 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.

Acknowledgement: Thanks to three anonymous reviewers for their helpful comments

References

- Brusilovsky P. (1996) Methods and Techniques of Adaptive Hypermedia, *User Modeling and User Adapted Interaction*, **6** (2-3), 87-129.
- Brusilovsky P., Schwarz E., Weber G. (1996) ELM-ART: An Intelligent Tutoring System on World Wide Web, *Proceedings of ITS'96, Lecture Notes in Computer Science No 1086*, 1996, Springer: Berlin.
- Diessel Th., Lehmann A., Vassileva J. (1994) Individualized Course Generation: A Marriage Between CAL and ICAL. *Computers and Education*, **22**, No.1/2, 57-64.

Nakabayashi K., Maruyama M., Koike Y., Kato Y., Touhei H., Fukuhara Y. (1997) An Architecture of an Intelligent Tutoring System on the WWW, to appear in *Proceedings AIED'97*, Kobe, Japan, 18-22 August 1997.

Nilsson N. (1980) *Principles of Artificial Intelligence*, Tioga Press: Palo Alto, CA.

Nykänen O. (1997) User Modeling in WWW with Prerequisite Graph Model. *Proceedings of the Workshop on Adaptive Systems and User Modeling on the WWW, 6th International Conference on User Modeling, UM'96*, Chia Laguna, Sardinia; also available at: <http://zeus.gmd.de/UM97/ws5.html>

Specht, M., Weber G., Heitmeyer S., Schöch V. (1997) AST: Adaptive WWW-Courseware for Statistics. *Proceedings of the Workshop on Adaptive Systems and User Modeling on the WWW, 6th International Conference on User Modeling, UM'96*, Chia Laguna, Sardinia; also available at: <http://zeus.gmd.de/UM97/ws5.html>

Vassileva J. (1992) Dynamic Courseware Generation within an ITS-shell Architecture. *Proceedings of ICCAL'92, International Conference on Computer Assisted Learning, 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. *Proceedings of ICCE'95 - International Conference on Computers in Education*, Singapore, AACE: Charlottesville, 290-297.