CREAM-Tools: An Authoring Environment for Curriculum and Course Building in an Intelligent Tutoring System

Roger Nkambou[1], Gilles Gauthier[2], Claude Frasson[1]

[1] Université de Montréal, Département d'informatique et de recherche opérationnelle
Case postale 6128, succursale Centre-ville, Montréal (Québec), H3C 3J7, Canada
E-mail: {nkambou, frasson}@iro.umontreal.ca
[2] Université du Québec à Montréal, Département d'informatique
Case postale 8888, succursale Centre-ville, Montréal (Québec), H3C 3P8, Canada
E-mail: gauthier.gilles@uqam.ca

Abstract. The main goal of the third stage of the SAFARI project is the delivery by an intelligent tutoring system (ITS) for a complete course. To achieve this goal, a subject-matter model, called CREAM, that can support course generation and delivery has been proposed. The acquisition of such a knowledge model requires to enable the designer with dedicated tools and methods. The purpose of this paper is to present the authoring environment (CREAM-Tools) we developped to support course and curriculum construction using the CREAM approach. This authoring environment consists of a course generation kit, of building tools and methodologies. Curriculums and courses produced with this environment are directly usable by an ITS. We also show ways other modules in an intelligent tutorial system can exploit the resulting curriculums and courses.

1 Introduction

Instructional design automation is becoming a major concern in instructional design theory [18, 4, 8, 19]. The instructional design process is often considered tedious and it needs expertise which is difficult to acquire. Its automation generally consists in providing the designer with a toolkit. Systems such as ID Expert [9], ID_2 [8], IDE [17], ISD Expert [19], the GAIDA system [4] and AGD [15] have been specified to that effect. These systems offer a set of tools to be used in the instructional design process. However, the authoring environment varies according to the model to be produced and how this one will be exploited in the teaching process. In AGD, the product is a course specification that can be used by a human teacher. In ID Expert, the product is a set of recommendations represented as teaching transactions.

A subject-matter representation model in an ITS (CREAM[1]) has been proposed [11]. In this model, a curriculum is represented and organized according to three models: the domain-knowledge, pedagogical and didactic models. These three models are combined in a transition network structure called CKTN (Curriculum Knowledge Transition Network).

[1] Curriculum REpresentation and Acquisition Model

The purpose of this paper is to present the environment of course and curriculum conception according to the CREAM approach. This environment provides the user with building tools, intelligent support tools, course generation kit, and curriculum and course building methodologies.

After a brief presentation of CREAM, the curriculum and course conception environments are presented. For each of them, the associated approaches are defined. We finally show ways other ITS modules can exploit the resulting curriculums and courses in a teaching/learning process.

2 The CREAM Approach to Subject-matter Modeling

We consider the curriculum in an ITS as a structured representation of the subject matter in terms of capabilities, instructional objectives and didactic resources (learning materials). Achievement of instructional objectives contributes to the acquisition of capabilities. This achievement is supported by didactic resources through learning activities (exercises, demonstrations, problems, simulations...).

We have created a rich and structured environment intended to support course generation, instructional planning and the learning process. Other approaches to curriculum modeling exist [11]. They generally do not encompass the three main views of the subject matter: domain knowledge, pedagogical and didactic.

CREAM considers these three aspects. This is achieved through a network of capabilities, of instructional objectives related to these capabilities and of didactic resources supporting the accomplishment of instructional objectives. Using these three knowledge structures, we construct what we call the **pedagogical structure** of the curriculum (CKTN). The CKTN contains particular relations between the elements of the three models (capability model, instructional objectives model and didactic resources model).

In the capability model, capabilities are linked together according to several types of links: analogy, generalization, abstraction, aggregation, misconception.

In the pedagogical model, an instructional objective is linked to the necessary resources for its realization, to other objectives, to prerequisite capabilities and capabilities towards the acquisition of which it can contribute. The association of teaching material (resources) to an objective considers the type of capability we want the student to acquire [3]. A *prerequisite* link from a capability C to an objective O expresses the fact that C is a precondition to the realization of O. A prerequisite link is characterized by its nature (*mandatory*, *optional*) and by the minimum mastery level required on the source capability to enable the selection of the objective (*entry level*). This entry level is specified by using values taken from an evaluation vocabulary. Several evaluation vocabulary exist in the literature [5, 8]. For example, using Klausmeier's vocabulary [5], a capability of type "concept" can be acquired at one of the following levels: recognize, identify, classify or generalize. A *contribution* link qualifies the way in which the realization of an objective contributes to the acquisition of a capability: it can be a *strong*, *moderate* or *weak* contribution. Several objectives can contribute to the acquisition of the same capability.

CREAM has been used to construct a curriculum related to the use of a BAXTER infusion pump. This curriculum supports a course which included the following learning activities:

- Activities related to theoretical aspects of the infusion pump (component explanations, infusion concept...). Didactic resources supporting this part of the course are given by HTML documents.
- Activities concerning the free manipulation of a Baxter pump by a nurse (through the execution of tasks).
- Activities related to motivation and help, notably activities where the nurse can be shown by the system how to execute a specific task.
- Activities regarding problem solving in which the nurse is facing a situation where she is required to program an infusion with the help of a the system.

3 The Authoring Environment

We have developed a toolkit (CREAM-Tools) for supporting designers in the curriculum and course building processes. CREAM-Tools possesses a variety of tools and methodologies for curriculum and course design including a course generating system, graphical editors, knowledge browsers, visualization and simulation tools, intelligent supporting tools and multimedia authoring tools. It has been developped under VisualWorks (Smalltalk) and constitutes a complete authoring system.

3.1 Curriculum Building Environment

The goal of this environment is to enable the designer to define, create and organize curriculum component objects. Therefore, the result is a curriculum object offering an interface accessible to other ITS components.

Curriculum Building Process.
Two approaches have been defined to support the construction of a curriculum: *content-driven and course-driven.*

In the content-driven approach, the curriculum construction begins with domain knowledge specification. After this stage we define instructional objectives, and then specify useful resources for the acquisition of the specified domain knowledge. The main advantage of this approach is that it leads to a curriculum with a "strong content" permitting the construction of several courses and pedagogical views. However, this approach is costly for a large domain as we are sometimes constructing needless details for a given course.

In the course-driven approach, the curriculum contruction begins with a course already in the designer's mind. This means that the general instructional objectives are known. Their refinement constitutes the main part of curriculum objectives specification stage. These specifications lead to low-level objectives permitting to figure out knowledge (capabilities) involved in the subject matter. We then reach the stage of resource specifications aimed at defining the material to be used. The advantage here is that it enables a good organization of the instructional objectives, but could lead to "poor content".

However, the curriculum development does constitute an iterative process. For instance, when the subject-matter content stage precedes the instructional objective specification one, new capabilities could be identified at the later and integrated to the subject-matter content.

Toolkits.

In the curriculum workshop (figure 1), we have two types of tools: Building tools and intelligent support tools.

Building toolkit : They are essentially *browsers*, graphical editors and multimedia edition tools. They allow the designer to make up curriculum objects (building of capability-model objects, objective-model objects and learning-material objects; definition of objectives and capabilities, creation of learning material). Learning materials are created by using special tools intended for this purpose (demonstration (figure 2a), problem, and multimedia editors). These learning materials are stored in a database reachable during the authoring process. The designer can test a learning material to verify if it corresponds to his need. CREAM-Tools also permits the creation of resources that can be used through a WWW client (figure 2b). For example, to create a multiple-choice question, an interface allows the designer to specify attributes (questions and possible answers). Our system generates a test object for use in a Web page displaying the test (figure 2c).

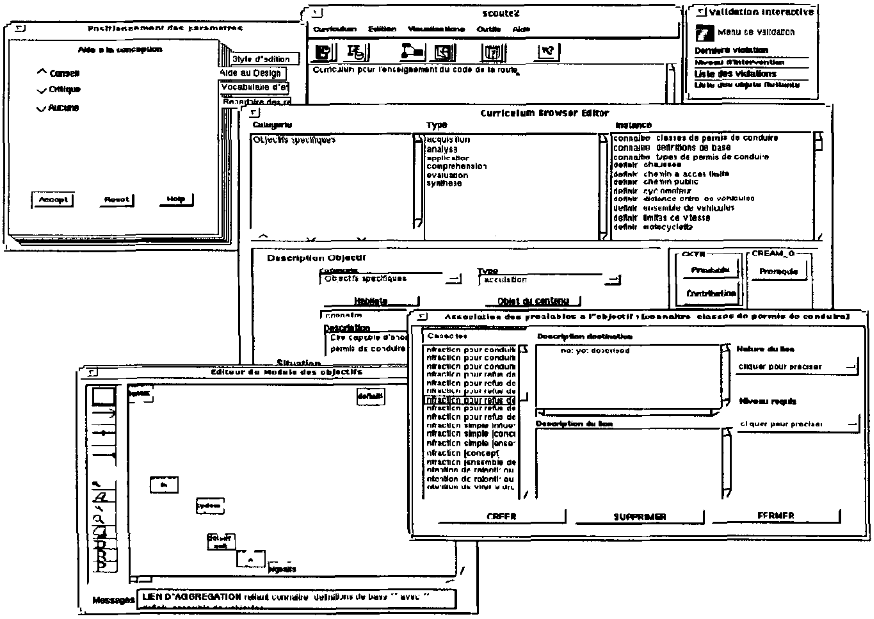
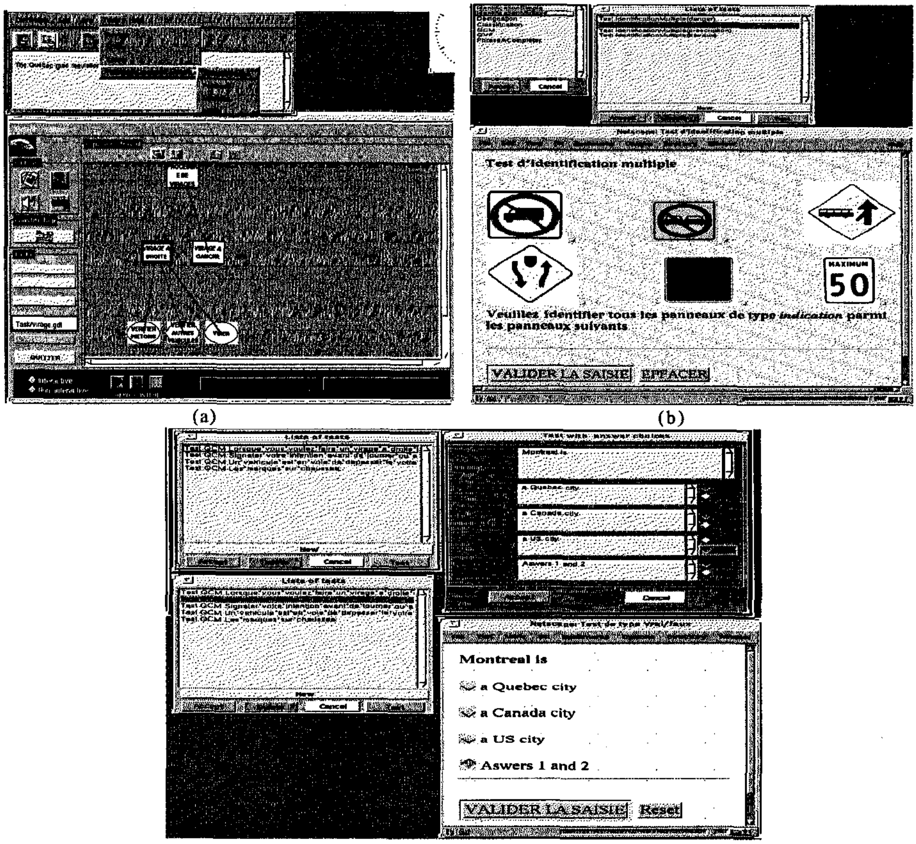


Fig. 1. An overview of the curriculum building environment

Intelligent assistant tools : Two intelligent tools are available in the curriculum building environment. One concerns the automatic generation within the curriculum (cf. 3.1.3), and the other one is an expert system for curriculum validation [13]. The later contains a knowledge base with about sixty rules extracted from instructional design theory [18, 4, 8, 19]. These rules support advising during the design process, or critic of the designer's work. Therefore, the system is able to give gradual hints (if the user so desires) at any moment of the building process. For example, a hint could be an error signal or suggestion. In critical mode, there is an intervention of the system only after the user's request. This intervention is formulated as general comments.



(a) (b)

(c)

Fig. 2. Some didactic resources building tools (demonstration and test editors).

Some Generative Issues.

We have examined some generative aspects of the design of a curriculum, using the CREAM approach:

- **Automatic generation of an objective model parallel to the capability model.** To do this, we have studied different classifications of knowledge involved in subject matter description [2, 8, 16]. Gagné's classification is the one used by CREAM. It has been used to define the behavior (by using the Bloom's taxonomy [1]) associated with each capability type. Then, when the capabilities model is defined, we can generate an objective associated with each capabilty in this model. For example, we derive the objective *define(infusion-concept)* from the capability named *infusion(defined-concept).* Links between objectives are generated from capabilities links. Thus, aggregation links will be interpreted in the objective models as constitution links.

- **Automatic generation of prerequisites links between objectives.** This is done by a reasoning about prerequisites and contribution links in the CKTN. If an objective O_1 produces a capabilty required by an objective O_2, then a prerequisite link is created between O_1 and O_2.

3.2 Course Generating Environment

The CREAM model can serve many educational purposes and can be exploited using the set of tools we have developed. In this section, we describe the course generation environment that allows the designer to generate a course, given some parameters. The course generation process starts with a specification of objectives or capabilities and performs a traversal of the corresponding curriculum-knowledge structures to obtain a description, in terms of objectives to be realized, of the course which will permit to achieve the desired goals.

The Course Concept.

In the teaching field, a course is a sequence of instruction periods, each one dealing with a particular subject and aimed at the evolution of a target-public's knowledge. In our model, we define a course as a structured set with three categories of objectives: global, specific and terminal objectives. A *global objective* being a statement that describes globally all the changes (cognitive, affective and psychomotor) that one wishes to induce in students' behavior during a course; a *specific objective* describes a set of behavior that the learner should be able to demonstrate. These objectives are specified in terms of the capabilities the learner should acquire; a *terminal objective* is, in our context, an objective that describes a precise performance the student should achieve. Generally, a specific objective is composed of several terminal objectives.

This set of objectives is centered on a well-defined educational purpose and can tackle various themes related to it.

If we put together some of the parts that result from our study of the different representations of the course concept proposed by [6, 7, 20], we may consider that a course is composed of three main parts:

- A *descriptive part*: title, description (pedagogical purpose description), and the set of general objectives. A general objective being a very abstract description of the course goals, most of the time a text summarizing them;
- A *course-graph* derived from the curriculum and containing instructional objectives linked together by curriculum links;
- A *structural part*: the structure of the different levels of objectives (course structure) and course themes.

Course Generation Approaches.

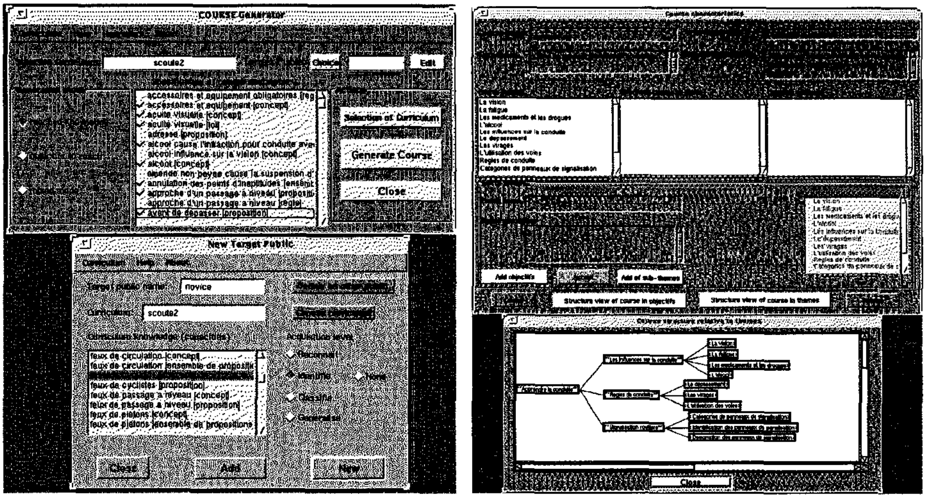
We put forward two main approaches for a course generation: Construction of a course from the set of objectives to be achieved, or from a specification of the knowledge (capabilities) a student should master after taking the course. This specification is called KTT (Knowledge to transfer).

The first approach is relatively straightforward: a set of objectives is specified and sent to the CREAM course generator. It builds a course graph from the curriculum structures using this set of objectives. The graph is then sent to the structuring process that determines the global, specific and terminal objectives and creates automatically a course structure (which can be further edited with the course editor). A structuration algorithm has been developed for this purpose and will also be used for the second generation approach. It analyzes the way the objectives are linked in the course graph in order to classify them in different objective categories. For example, all objectives without prerequisites will be considered as global. After the structuration, we can visualize the course structure and observe that it is composed of several learning hierarchies [3].

The second approach consists in generating a course from a domain knowledge specification (a set of capabilities). This set represents the knowledge the student should master after taking the course we want to generate. Obviously, all the capabilities in this set must belong to the curriculum which is to be concerned by our course. It is from this set that our generator will go through the CKTN in search of the objectives which will permit to acquire the given capabilities.

Course Generation Toolkit.

Fundamental objects for our course generation and for the whole system have been implemented: a curriculum object, a CKTN object, a course object... They are used extensively in the generation process. The algorithm developed for the generation requires: a CKTN, a knowledge specification (KTT), an optional target group and the description of the heuristic to be used.



(a) (b)

Fig. 3. Some tools of the course building environment

We created a target public editor (see figure 3a) to let the designer create his target groups and use them afterwards to generate his courses. Our editor presents him a set of knowledge units related to the desired subject matter and he has to select the acquisition level of his public on the capabilities he is including in his target group. The editor also considers the different acquisition levels as well as the acquisition vocabulary chosen by the designer and kept in a global variable of the system. So, the system adapts the terms presented to the user to qualify the different acquisition levels to the right vocabulary.

With this editor, the instructional designer can mark the different states of knowledge of his target group. He can also modify one of the existing target groups.

A CKTN object is automatically defined when a new curriculum object is created (with our system's curriculum editor). Its prerequisite links and capabilities are marked according to the target group knowledge, to produce a DynCKTN on which the generator will now reason on.

We can also already remove from the KTT the knowledge that the student possesses according to the marking, since it is not useful to include in the course the objectives for its acquisition.

Course Generation with Heuristic Reasoning.
After the optional specification of the target public, the specification of the KTT and the choice of the heuristic, the automatic generation begins. Three possible heuristics for a course generation are now implemented: *general inclusion*, *random choice*, and *complexity evaluation*. Details of these heuristics are given in [12].

The generated set of objectives is used to construct the course graph which is then displayed in the course editor to allows its modification by the designer if he wishes. A window of statistics also appears to give some information about the generation: The specified KTT, the number of objectives chosen and which ones, the KTT's capabilities already possessed by the student (according to the target group) and the capabilities which require the course to have a prerequisite. After that, the course graph is passed to the structuring process which generates a course structure. A structuring environment (figure 3b) is also opened to allow the designer to specify the descriptive part of the generated course, and different themes to be considered during the delivery process.

4 Using Course and Curriculum Objects in an ITS

We built an ITS using curriculums and courses produced by CREAM-Tools. In this ITS, curriculum object is used to support inferences in the student model [14]. The student model is an overlay on the curriculum. The planner uses the curriculum and course objects to decide the focus of the instruction, according to the student knowledge. A student interface has been design to support student activities. This interface offers metacognitives features based on the curriculum model [10].

5 Conclusion

A curriculum and course models in an ITS have been proposed. They have been experimented through the construction of a complete course and its delivery. An authoring system has been proposed to help build up these models. Curriculums and courses produced from this system are directly usable by other ITS component. They can also be used to support the decision making in the teaching-learning process (by the planner and the tutor) or student modeling. To point out how our models allow large scale ITS development, we have set up some applications using them (clinic examination in an intensive care unit curriculum, Quebec road regulation curriculum...). The prototype is actually functional.

Acknowledgments

We thank the MICST (Ministère de l'industrie, du commerce, de la science et de la technologie du gouvernement du Québec) for providing financial funds for the SAFARI project. We also thank M.-C. Frasson, E. Madja, H. Boulhama, D. Lupascu, N. Zalida and T.J. Pelleu for their collaboration.

References

1. Bloom, B. S. (1969). *Taxonomie des objectifs pédagogiques.: domaine cognitif.* Montréal, Éducation nouvelle.
2. Gagné, R.M. & Briggs, L. (1979). *Principles of Instructional Design.* (2nd edition). Holt, Rinehart and Winston, NY.
3. Gagné, R. M. (1985). *The Conditions of Learning* (4th edition). Holt, Rinehart and Winston, New York.
4. Gagné, R. M. (1993). Computer-Based Instructional Guidance. In: Spector, J.M., Polson, M.C. and Muraida, D.J. (Eds.): *Automating Instructional Design: Concept and Issues.*; pp. 133-146, ETD. Englewood Cliffs, N.J.
5. Klausmeier, H.J. (1990). Conceptualizing. In B. F. Jones and L. Idol (Eds.): *Dimensions of Thinking and Cognitive Instruction*; pp. 93-138; LEA, NJ.
6. Koehorst, A., M. (1994) Curriculum Graphs As A Backbone For Computer Supported Design And Delivery Of Education. In *Proceedings of the 7th International Conference on Systems Research, Informatics and Cybernatics*; pp. 31-36.
7. Lewis, R., Goodyear, P., & Boder, A. (1992) Just-in-time Open Learning: The background and objectives of a project supported by the European Commission under the DELTA programme; *Occasional Paper NL/1/92*, Neurope Lab.
8. Merrill, M. D. (1993) An Integrated Model for Automating Instructional Design and Delivery. In: J. M.Spector, M. C. Polson, and D. J. Muraida (Eds.): *Automating Instructional Design: Concept and Issues*; pp. 147-190, ETD. Englewood Cliffs, NJ.
9. Merrill, M.D. (1987). An expert system for instructional design. *IEEE-Expert* 2 (2) 25-37.
10. Nkambou, R., Gauthier, G., & Frasson, C. (1996). An Intelligent Tutoring System Interface Supporting Metacognition. In *Proceedings of the 13th International Conference on Technology and Education*, Orlando, NO.
11. Nkambou, R. & Gauthier, G. (1996). Un modèle de représentation du curriculum dans un STI. In *Proceedings the Third International Conference on Intelligent Tutoring System*, LNCS 1086, Springer-Verlag, Berlin.
12. Nkambou, R., Frasson, M. C., & Frasson, C. (1996). Generating Courses in an Intelligent Tutoring System. In *Proceedings of the IAE-AIE'96 Conference*, Japan.
13. Nkambou, R., Gauthier, G., Frasson, C. & M. Antaki (1995). Integrating Expert System in an Authoring System For Curriculum and Course Building. In *Proceeding of the 7th International Conference On Artificial Intelligence and Expert Systems Application*; pp. 485-490, San-Fransico, CA.
14. Nkambou, R., Lefebvre, B., & Gauthier, G. (1996). A Curriculum-Based Student Model For Intelligent Tutoring System. In *Proceedings of the Fifth International Conference On User Modelling*; pp. 91-98, Kailua-Kona, Hawaii.
15. Paquette, G., Crevier, F., Aubin, C., & Frasson, C., (1994). Design of a Knowledge-Based Didactic Engineering Workbench. In *CALISCE 94*, Paris.
16. Paris, S.G. & Winograd, P. (1990). How Metacognition Can Promote Academic Learning and Instruction. In B. F. Jones and L. Idol (Eds.): *Dimensions of Thinking and Cognitive Instruction*; pp 15-51, LEA, NJ.
17. Pirolli, P. & Russell, D.M. (1991). Instructional design environment: technology to support design problem solving. *Instructional Science*, 19 (2), 121-144.
18. Spector, J.M., Polson, M.C., & Muraida, D.J. (1993). *Automating Instructional Design: Concepts and Issues.* ETD. Englewood Cliffs, N.J.
19. Tennyson, R. D. (1993). A Framework for Automating Instructional Design. In: J. M.Spector, M. C. Polson, & D. J. Muraida (Eds.): *Automating Instructional Design: Concept and Issues*; pp. 191-214, ETD. Englewood Cliffs, NJ.
20. Webster, J. G. (1994) Instructional Objectives and Bench Examinations in Circuits Laboratories. *IEEE Transactions On Education* 37 (1) 111-113.