Teaching Expertise is at The Core of ITS Research

Monique Grandbastien

International Journal of Artificial Intelligence in Education (1999), 10, 335-349

Teaching Expertise is at the Core of ITS Research

Monique Grandbastien, *LORIA – Université Henri Poincaré Nancy1, Campus Scientifique – BP 239, 54506 – VANDOEUVRE Cedex – France*
Monique.Grandbastien@loria.fr

Based on an invited talk presented at the ITS 98 conference

Abstract. To design efficient, flexible and user-adaptable learning environments, we need to embed a great deal of knowledge into them. This paper emphasises the teacher's knowledge, which encompasses all the expertise of the teacher or of the trainer. Thus it is broader than that which is usually called tutoring knowledge. We start with a description of several facets of teaching expertise and of various implementations that have been described in the ITS literature. Then we describe in more detail some prototypes that were built by our research team during the past ten years. We focus on the benefits of having practitioners on the design teams in order to provide us with their teaching expertise. We provide a framework for organising this knowledge into categories based on different viewpoints. Using this framework, we discuss the acquisition and modelling problem; in our view commonly shared ontologies in the domain have not yet been fully developed. Finally, we propose architectures suitable for reflecting teacher's knowledge, moving from the traditional, but old "four modules model" to more up to date proposals such as agent models and environments open to teachers. Our conclusion is that much has been accomplished, but the context is rapidly changing, especially in terms of communication and co-operation. Thus major research initiatives need to be undertaken to face these new challenges. Our task becomes LARGER, not SMALLER.

INTRODUCTION

The necessary updating of professional skills, the existing dissatisfaction with many educational systems and the aspiration for personal development call for more individualised education and training. Network and Web technologies are now available to provide education and training nearly anywhere at any time. What we need now is to design efficient, flexible and user-adaptable learning environments, the expression ITS refers to such environments in this paper. The design of ITS requires us to embed a great deal of knowledge into them. People often classify this knowledge into four categories: domain, tutoring, student and interface, based on the ITS architecture popular in the eighties. This paper emphasises teaching knowledge, which encompasses all the expertise of the teacher thus it is broader than what is usually called tutoring knowledge.

From a knowledge engineering viewpoint, ITS belongs to the category of complex knowledge-based systems. Consequently, they share all the features of these systems. One of the main features is that the performance of such systems mainly relies on the quality, quantity and appropriateness of the different kinds of knowledge embedded in the system. ITS can be characterised by the tasks they are supposed to support and by the expertise they should contain. First the task or process they are supposed to support is learning; learning is a crucial process for everyone, but it is not yet well understood. Second, the expertise that should be embedded into the systems is "owned" by teachers and trainers, which means that this expertise is spread all over the world and also that it is personal and very different from one teacher to another one. So, teaching expertise and learning processes are key components of ITS research and development. Although learning is very important, I will not address directly the learning process in this paper, I will rather focus my presentation on the teaching expertise. Indeed I rely on the experience gained through two decades of close relations with teachers using computers and through the design of several prototypes of educational software.

Much thought has been given to teaching expertise by ITS developers, even if this expertise is not enough identified as such and provided in an explicit declarative way. This paper does not aim at providing a complete overview of the work already done on embedding teaching expertise in tutoring systems. Rather it reflects a personal view of some key ideas in the field. This view was developed through a professional history in both research and teaching activities. It is certainly influenced by a computer science approach as my background is in computer science, mainly in artificial intelligence. But, more important, my professional experience includes more than twenty years of close collaboration with teachers in many different contexts. I was in charge of introductory courses in the use of computers in education for teachers, of the management of projects aimed at using computers in the classrooms, of the design of computer science courses for beginners in secondary education, etc… I was soon convinced that computers must be brought into the schools with the co-operation of the teachers. When testing educational pieces of software, I observed that teachers are willing to use good products that remain open to their own creativity. A teacher needs professional tools that he uses as a craftsman. These observed needs have been the main thread of the work developed in the LORIA research team on computers and education, work from which most of the following examples are borrowed.

The paper is organised as follows. The first part is devoted to the analysis of teacher's knowledge in several existing prototypes. First, I provide a general view on what I call teaching expertise and on how teaching expertise was viewed in ITS literature. Next I describe the teaching expertise embedded in two systems that were built in our research team, ECSAI and Calques 3D. Once we have a better understanding of what teaching expertise is, two questions arise in the context of developing computer-based tutoring systems. Which part of this expertise can we represent and include in our systems? With which models, methods and tools should we develop those systems, as far as embedding teaching knowledge is concerned? Some answers to these two questions are provided in the second part of the paper. I conclude by discussing the major research efforts and sharing initiatives that could in my view help in understanding and representing teaching knowledge, for improving education and training, either computer-based or not.

WHAT IS TEACHING EXPERTISE?

Learning, and especially learning at school, is a situation that has been experienced by everyone, at least in the developed countries. Some people conclude from that experience that they would be able to teach since they had enough technical background in the domain to be taught. They do not realise that subject matter knowledge is not enough to become a teacher, just as the expertise embedded into the MYCIN system was not sufficient to turn it into a tutoring system. However many observations have shown that teaching is a very difficult task which requires specific skills (Leinhardt & Greeno, 86).

The many facets of teaching expertise

Teaching expertise has many facets. The teacher organises the subject to be taught in several lessons and each lesson includes several topics. The topics may be combined into learning objectives. For each topic the instructor makes the choice of appropriate presentations and examples, followed by exercises and assessment activities. For those choices, he relies on what he knows about his students and about the context (how can I keep their attention for the last lesson in the day?) and on his previous experience. During a lesson, the teacher adapts his presentation to the reactions and questions of the students; he relies a lot on their feedback and on observation of their behaviour. When they are involved in personal or group activities, he provides help on request and also without request, when he feels that providing advice is suitable.

Several observations can be inferred from the above description. Above all the teaching activity is made of professional expertise plus strong human involvement, This teaching expertise is not yet well formalised. Teaching expertise includes domain specific knowledge (topics in a lesson, examples, exercises, etc…) and domain independent knowledge (general pedagogical rules, adaptation to contextual constraints such as managing the last lesson in the day). Between domain specific knowledge and domain independent knowledge we find a wide range of tutoring expertise that could be applied in several domains. Representations for understanding images in medical diagnostic imaging as described in RUI (Direne, 97) which elaborates upon previous works on visual recognition provide a first example; simulation-based environments for discovery learning (Joolingen, & de Jong, 96) have also been used for a large variety of target domains. Last, but not least, existing teaching expertise was mainly built without computers in the schools. So this existing teaching expertise is certainly a good starting point for building computer-based environments, but a new expertise is needed to assist the pioneer teachers who integrate educational software into their lessons.

These different facets of teaching expertise need to be more deeply analysed and described, in order to decide which of them could be, at least partly, implemented in a computerised environment. Such an analysis requires a joint effort from people working in several complementary domains including subject didactics, cognitive psychology, educational sciences, educational sociology, knowledge engineering and educational software developers. Some of these facets have already been implemented in existing prototypes and provide us with an overview of the subset of teaching expertise than can be embedded in computer-based systems.

Teaching expertise in previous research and prototypes

It has been acknowledged from the very beginning of ITS research that a knowledge-based system without specific tutoring knowledge cannot be used successfully for learning purposes. A well known example is provided by the GUIDON (Clancey, 79, 87) and NEOMYCIN (Clancey, 86) projects that were built on the MYCIN expert system for training students to learn from the expertise embedded into MYCIN to diagnose infectious diseases. Research in Artificial Intelligence has often been experimental; models and architectures mostly come from close observation and analysis of working prototypes. This has been, and continues to be true in the field of ITS. The analysis of the first prototypes led to a four-module architecture: domain, tutoring, student model and interface. The first three were presented in (Hartley, & Sleeman, 73) and the whole list has not been changed in more than twenty years (Sleeman, & Brown, 82), (Kearsley, 87), (Lawler, & Yasdani, 87), (Wenger, 87). If the so called “tutoring module” or “instructional module” is entirely devoted to teaching expertise, the other three modules also include teaching expertise. In the domain module, we can find the choice of the set of concepts, their organisation in a network of specific dependencies, the presentation of the subject matter, the examples that are provided; all require a lot of teaching experience. The student model mostly relies on student features that have been chosen by teachers. Last but not least, the interface module relies on teaching expertise for the design of student machine dialogues, the availability of explanations and more generally for all feedback to students’ requests.

As stated in the introduction, a review of how teaching expertise is represented in an explicit way or partially hidden in ITS research is not the focus of this paper. However, recent reviews of existing ITS prototypes have been made from other viewpoints and we can benefit from what they report about teaching expertise. (Shute, & Psotka, 95) search for answers to questions such as “What does the term (ITS) means?” and “What are important milestones and issues across the 20+ year history of ITS? As far as teaching knowledge is concerned, they emphasise the diagnosis of errors and misconceptions of the learners based on a dynamic assessment of a student’s performance as well as appropriate remediations. They conclude that progress will come from research about effective and efficient ways to represent, utilise and communicate domain knowledge, represent an individual evolving knowledge state (for both declarative knowledge and procedural skills) and instruct the material most effectively for a particular learner.

(Murray, 99) proposed an in-depth survey of ITS research from the authoring viewpoint. The creation of environments to facilitate the design and development of ITS has been a continuing thread in ITS research. So the analysis of the related productions give an interesting feedback on the way in which authors (that could be teachers) could encode their expertise with the provided tools. Murray identifies models of instructional contents and teaching strategies as main features of ITS systems. Then he categorises existing authoring systems according to their major differentiating factors. One factor is "tutoring strategies" which include "when and how to give explanations, summaries, examples and analogies; what type of hinting and feedback to give; and what type of questions and exercises to offer to the student". Some systems such as COCA (Major & Reichgelt 92; Major 95) focussed on implementing teaching strategies and on providing teachers with sophisticated parameters to activate one of these strategies. Another factor is the curriculum sequencing based on instructional units that are related by prerequisites, part and other relationships. The fine-grained cognitive student models that can be authored in "model tracing tutors" (Anderson & Pelletier, 91) provide another example of teaching expertise.

A deeper analysis through two case-studies

We propose to come to a better understanding of teaching expertise by taking a closer look at two systems that were built by our research team. The first one is a generic environment aimed at building an ITS, the second one is a micro-world that permits students to explore dynamic 3D geometry. In both cases, we will address the following questions:

Where is teacher's knowledge hidden?
How was it acquired, how is it represented?
What kinds of architectures have been used?
How can end-users (teachers) include their own expertise?

ECSAI: A GENERIC ENVIRONMENT FOR DESIGNING ITS

General overview

ECSAI[1] (Gavignet, 94; Gavignet & Grandbastien, 92; Grandbastien, & Gavignet, 94) is a generator of learning environments. According to the categorisation introduced by (Murray, 99) for authoring environments, ECSAI would mainly fall into the first category devoted to "Curriculum Sequencing and Planning", where "intelligent sequencing" of instructional units is at the core of the system, although it provides features from other categories as well. It includes a formal model for describing the subject matter to be taught, tools to index and populate a database of "learning units" and a dynamic linkage mechanism to create a courseware program adapted to each learner's behaviour. It belongs to a wide category of authoring systems where the domain to be taught must be divided into topics organised in a network. It is not oriented towards a specific set of domains. It can be used as soon as the author is able to build such a model of the domain. Then the author has to provide a set of elementary pieces of software called "learning units" dealing with that domain and to fill each unit's profile in terms of topics, prerequisites and results, links with other units and other data. At that stage, the environment is populated for a given domain, we call it ECSAI-X.

ECSAI-X is a learning environment for students in a domain X (e.g. biology, solving algebraic equations, past participle in French). A student is asked for the learning objectives. Then ECSAI-X computes the first learning unit that is to be presented, displays it to the student, updates its student model according to the student behaviour during his interaction with the unit and then presents the next unit until the objective is reached or there is no more available units. Experiments with ECSAI have been conducted with chemistry teachers and with biology teachers.

[1] ECSAI is a French acronym which stands for » environment for the design of interactive learning systems »

Although the ECSAI model was published in 1991 (Gavignet, 91), it is still completely up to date in the way it describes and links the many documents that begin to be stored in several "knowledge pools". It could for instance fit with the approaches described in (Forte et al., 97) in ARIADNE, a project funded by the European Union for indexing educational resources or with the IEEE P1484 standardisation initiatives for indexing and retrieving learning material through the Web. It exemplifies a wide set of authoring systems based on the same principles. So it is worthwhile to further explore the analysis of the underlying model and to observe where teaching knowledge occurs. The main features of the model are described in the "teacher's viewpoint" section while the dynamic linkage mechanism is presented in the "learner's viewpoint" section.

Building an ITS with ECSAI: the teacher's viewpoint

The authoring process with ECSAI can be divided into several steps. First, the author creates or imports from available documents a set of learning units, which are separate pieces of software. This set is the database of physical units that will be invoked at runtime. The author then creates the model of the domain; the subject matter domain must be described through a network of elementary topics. As far as a domain network is concerned, two main questions occur. What is the suitable level of granularity for each topic? Which kinds of links are allowed between the topics? ECSAI provides very precise answers to both questions. As far as the level of granularity of concepts to be taught is concerned, ECSAI proposes the idea of "evaluation items". In other words, the list of concepts that is used in an ECSAI-X system includes the concepts, which can be, associated with learning objectives and evaluation processes. Of course, the learning units may deal with many other concepts, but only the evaluation items are included in the domain model. Semantic relations between items are represented by semantic links in the domain network. There are two types of links. The first type represents the hierarchical organisation of the domain, e.g. a wheel is a component of a vehicle and the second type represents pedagogical constraints such as "this item must be learned before that one". Figure 1 enlightens the co-operation between several formal models and a database of learning units in ECSAI.

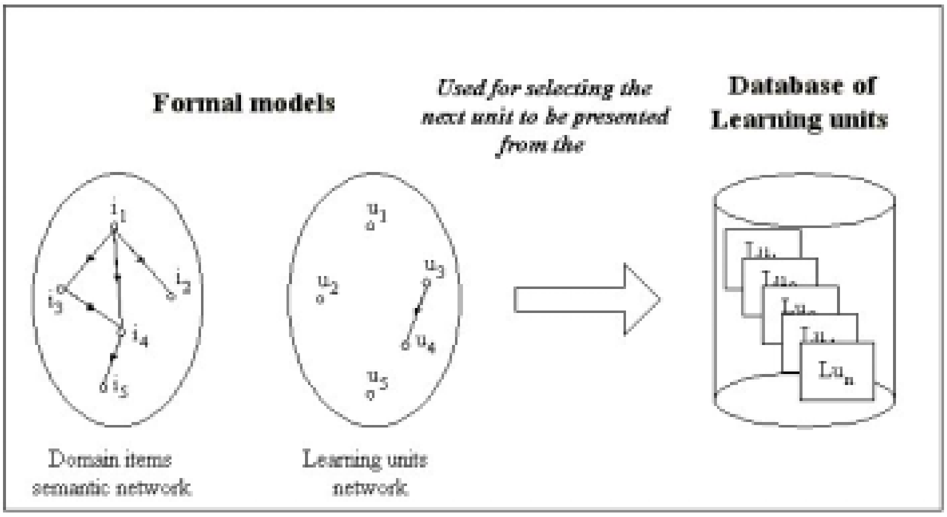

Figure 1: ECSAI viewed as a set of models and a set of physical learning units

Once the domain network is completed, the final step consists in describing each learning unit in the model. A unit description includes a label, a type, a free content description, pre-conditions and post-actions. The label is used to refer to the given unit in other parts of the model. The type of unit can be presentation, example, exercise, test, etc. The content description is a text used when the author is browsing through the available set of unit models. The pre-conditions set up the constraints that must be satisfied in order to run the unit, these constraints include other units that must have been already completed by the student as well as on the learner's level on a given list of items. For instance, unit number 14 could be proposed only if unit number 11 has been previously presented and if the learner's level is "good" on item 17 and "medium" on item 18. The post-actions describe the changes that the system will operate in the learner model when the unit has been completed. For instance the level becomes "good" or remains "medium" on item 18, depending on the results obtained in completing the learning unit. Figure 2 illustrates the formal description of a learning unit in ECSAI where tokens model the learner's level for every item.

Pre-conditions	**Content**	**Post-actions**
Previous unit: U15 Item 17: tokens = 3 Item 18: tokens = 2	Label: U14 Type: Exercise Item used: i3, i4, i5, i8	Item 5: tokens = 1 Item 18: token = +1 depending on results

Figure 2: An example of formal description of a learning unit in ECSAI

The learning unit descriptions are also organised into a network. Some of the links are static ones, for instance the author can set up a "followed by" link between two units to indicate that the second one must always be presented immediately after the first one. But the ECSAI engine dynamically creates most links from the unit pre-conditions and from the data stored about the student. Indeed, even if the author is not asked to build a learner model, the system includes an overlay model in which the performance of the learner is represented by tokens attached to the domain items. This simple learner model is updated throughout the session. The initial model can be empty or filled in through a questionnaire.

Using an ITS built with ECSAI: the learner's viewpoint

After the learner has been identified by the system, he is asked for his learning objectives based on choices from a given menu. The objectives are specified in terms of elementary items as defined in the domain model and levels of competence for these items. For instance, one can ask for learning "past participle" with the level "advanced". From the given learning objectives and from other initial data, which both form the initial state of the learner model, the system determines the first learning unit to be displayed.

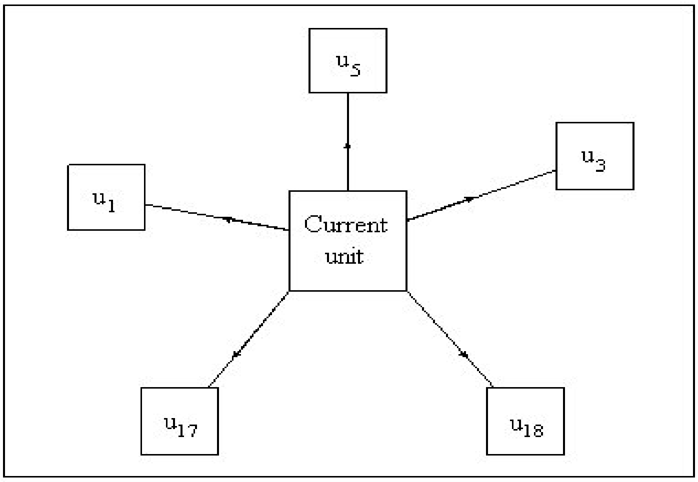

Figure 3: Units directly linked to the current unit at runtime

The choice is made according to an algorithm included in the system. Units for which the pre-conditions have been satisfied by the learner model are ranked according to additional criteria and the first one is proposed. After the unit is completed, the learner model is updated according to the post-actions of the unit, and the next unit to be displayed is computed. The system stops either when the learning objectives are attained or when no other unit can be displayed. Figure 3 shows a set of units at runtime; several types of links are used to link the current learning unit to other units, which can be selected as next unit; they have to be ranked in order to compute the next unit that will be displayed.

Summary: Where is teaching knowledge?

Let us now come back to the main questions we need to answer: Which teaching expertise is embedded in an ITS built with ECSAI and where is it hidden? I would answer there is teaching expertise everywhere, so let us look at it step by step.

We first talked about a set of learning units. Each unit includes domain-related expertise, for instance appropriate examples, clear definitions, graphical displays as well as attractive interactions. Then we discussed about the domain model. Determining the domain items on which this model will rely requires other capabilities from the designer at a higher modelling level than the design of each unit. Next we talked about describing each learning unit in the domain model. Here you have not only to think about the abstract model of the domain to be taught but also to the dynamic links that will result from the units specified at runtime. We have identified three steps where the author has to bring his teaching expertise into the system. But another part of the teaching expertise is “wired” into the kernel. It includes the model that is proposed to describe the domain and the units; it also includes the criteria used to dynamically build the learning path for each user. As ECSAI is intended to be a domain-independent framework, this “wired” teaching expertise is mostly generic. The domain-dependent expertise is located within each learning unit. It also occurs for designing the domain model and for using it in the learning unit descriptions.

Menu-driven interfaces allow the author to design the domain model and to describe the learning units he intends to use with his students. A teacher can easily enrich a given ECSAI-X environment. Keeping the domain semantic network, he can add new learning units by describing them using the appropriate interface and remove existing ones. After any addition or removal, he can shift from the authoring mode to the run-time mode in order to check the effect of the proposed modifications. So, environments designed with ECSAI remain open to teachers in the sense that teachers can change the set of learning units that will be used for students. If they possess authoring skills, they can even bring their own learning units into the system. But we cannot expect that all teachers are going to author new learning units and there is no other global parameter to set up global teaching strategies as those provided in COCA for example. We foresee to provide a more explicit representation of the control process that is still hardwired in the ECSAI engine.

CALQUES 3D: A DOMAIN-DEDICATED MICRO-WORLD

General overview

Calques 3D is a micro-world, i.e. a type of Interactive Learning Environment designed for constructing, observing and manipulating 3D geometrical figures. It intends to reduce the difficulties observed in teaching spatial geometry in secondary schools.

It allows an intuitive and adaptable access to environment features, intuitive because it has to be used by students who do not have preparation, adaptable because it allows teachers to decide with respect with their own pedagogy which primitives and operations will be made available to their students. The basic ideas underlying the design of Calques 3D result from P. Bernat professional experience in teaching mathematics as well as in designing educational software and in training teachers.

In Calques 3D, geometrical figures are constructed from a set of elementary objects (points, lines planes, etc.) and a set of construction primitives (intersection, parallel, perpendicular, etc.). Once a geometrical figure has been constructed, students are expected to use the observation and exploration facilities provided by the environment. To help the student observing the figure, Calques 3D facilities include:

- Materialisation of the spatial system of reference (e.g.; axes, floor, etc.)
- Several perspectives (cavalier, vanishing point)
- Modification of the observer's point of view
- Display of visual feedback on objects (e.g. projections of points' co-ordinates on a plane) as shown in Figure 4.

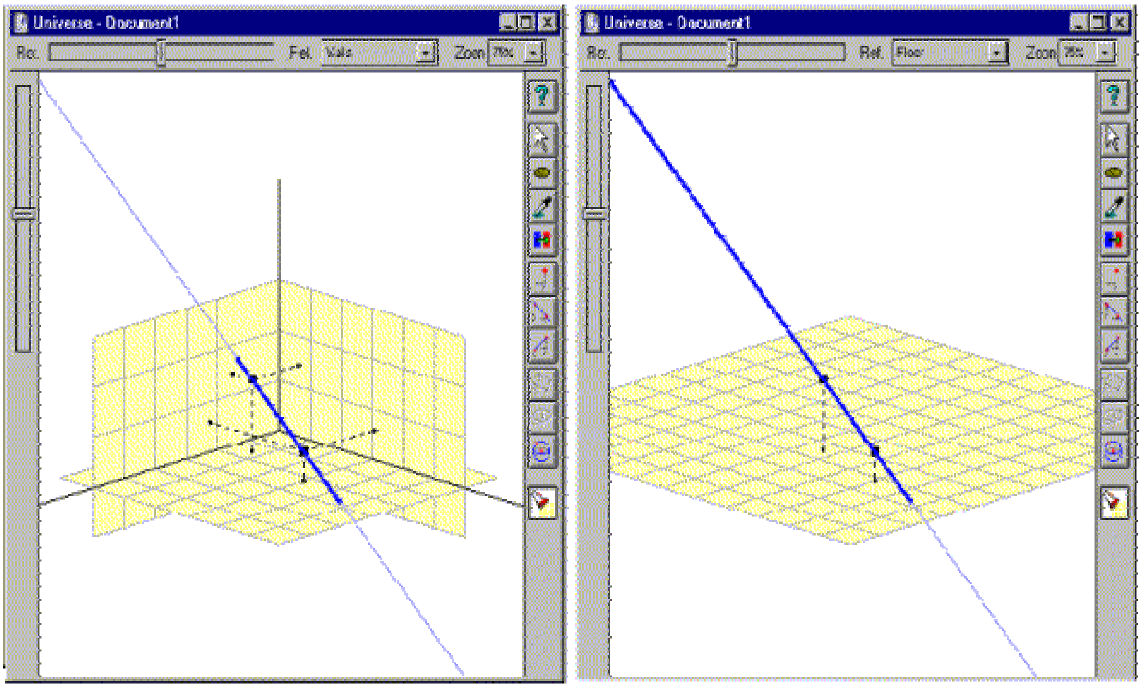

Figure 4: Snapshot of Calques 3D environment

Sometimes, these facilities do not provide enough visual information to the learner; indeed it remains a plane representation of a spatial figure. Two other features can help the students:

- Dynamic deformation of figures by directly dragging base-points. It provides the user with another way to explore the figure. This exploration is realised through an interface based on an extension of the direct manipulation concept, as provided for instance in Geometer's Sketchpad or Cabri-geometry, for spatial geometry.
- Synchronised tracings: Problems in 3D geometry often result in complex figures. To overcome this complexity, Calques 3D allows the user to extract geometrical objects in separate synchronised tracings, as illustrated in Figure 5

An incremental design process keeping teachers in the loop

When launching the Calques 3D project, our long-term research aim was to develop materials and techniques that would allow us to capture the knowledge those teachers have about "how and why they teach concepts in particular ways". We intended to use this knowledge to provide them with environments that they could adapt to their teaching needs. Since teachers have no experience in teaching 3D geometry using a dynamic geometry environment, we adopted an incremental design process to capture a part of the experience of a team of teachers. We started by giving them a first version of Calques 3D and we asked them to describe the sequences of activities they planned for their students. Then, every two months, we provided them with a version taking into account the modifications we agreed upon during the previous meeting and we discussed the ways they used Calques 3D as well as new functions that could help them in achieving their teaching objectives. A more detailed description of this process can be found in (Van Labeke & al., 99).

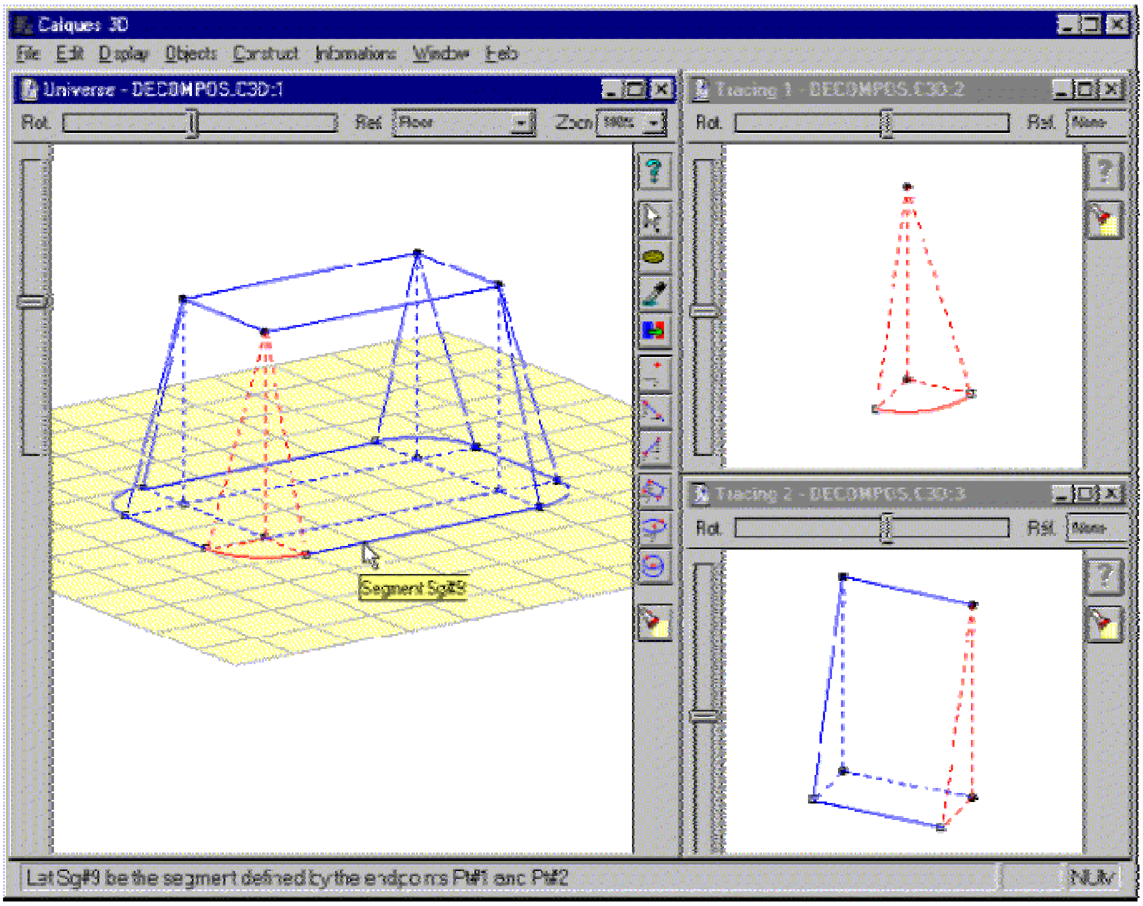

Figure 5: A complex figure and its associated tracing windows

Towards teaching expertise capture

In Calques 3D teaching expertise is related to the teaching of spatial geometry. But it is multi-faceted and includes:

- Sequences of activities planned for the students.
- Basic operations provided for constructing figures.
- Visual units provided for comprehension (they were suggested by the teachers of our team from their previous experiences with paper-based figures, but the final implementation results of several modifications that were made after discussions).
- Presentation of geometric objects which depends on the didactical situations the teacher wants to realise (e.g. a cube may be represented as a wireframe, i.e. all the edges are visible, or as a hidden faces cube for using the cube as a solid object).

Part of the expertise was expressed through “forms” in which teachers were asked to describe the sequences of activities that they provide to their students. These sequences include non computer-based activities, which are as informative as computer-based ones for capturing teacher’s expertise; moreover, supporting non computer-based activities was already noticed as an important requirement from teachers in (Major & Reichgelt, 92). From these descriptions, we proposed menus allowing teachers to shape the Calques 3D environments to some of their ways of teaching. Next step for us would consist in automatically providing a teacher with the environment configuration he needs according to the described sequence of activities. Other teacher's requirements were expressed during the bimonthly discussions.

Our experience in designing Calques 3D with a team of practising teachers from several secondary schools representing various kinds of education (general secondary education, professional training) reinforced us in our conviction of keeping teachers and trainers in the design loop. It also exemplifies the many facets of teaching knowledge and underlines the need for organising it, first to better implement it into future ITS and second to allow reusing it in further developments. The next paragraph proposes several views for describing teaching knowledge.

ORGANISING TEACHING KNOWLEDGE

The previous paragraphs have shown that teaching knowledge is embedded everywhere in existing prototypes, but most implementations are still beginning from scratch instead of building on existing material; few papers are really comparing their proposal with previous ones, mainly because there is no common framework for such comparisons. We have also shown that there are many categories of teaching knowledge. Now we need to organise the extensive but very unordered data that is available on that topic. Starting from the four ITS modules and taking into account the importance of domain dependant knowledge as well as specific computational constraints, we propose several views to organise teaching knowledge. The first one uses the domain, tutoring and student model classification. The second one aims at enhancing domain related pedagogical expertise and raising it to the same level of importance that was given to general tutoring knowledge up to now. The third one is a novel categorisation that is necessary if we want to take into account computational constraints.

Domain, tutoring and student knowledge

This view does not require more comments as it has been widely used. Nevertheless, domain knowledge and tutoring knowledge include very different features, which must be developed in tandem. Knowledge about the student could also be divided into static knowledge that is provided to the system before interacting with the ITS and dynamic knowledge which is inferred through the analysis of the interactions between the system and the learner.

General tutoring knowledge versus domain related pedagogical expertise

Most teaching expertise explicitly available in ITS frameworks, is domain independent while most working prototypes are strongly domain dependent. The main reason for that is the goal to build generic environments that could be used in different educational settings for a large number of subjects. Such a focus on generality is for instance to be found in the FITS project (Ikeda & Mizoguchi, 94) where the authors explain that formulating problems and constraints at an appropriate level of abstraction should allow to attain high generality and efficiency. They confirm that tutoring strategies require domain specific data and facts, but that by domain-independent tutoring strategy, they mean that the mechanism should be domain-independent, not the data used.

When you work with teachers, especially with secondary teachers, you observe that they express their teaching expertise in close relation with the domain in which they are teaching. So the tools that they need and will most likely accept and integrate into their daily practice should be domain-related. In my view development should be domain related. It is up to researchers and designers to decide whether there are some common features that can be reused from one domain to another. At least each teacher should be provided with a domain-related interface adapted to his own needs. Moreover, more domain-related teaching expertise should be modelled and made widely available in order to avoid re-inventing the wheel for each new design. For this task we need to invent and provide models at the appropriate level of abstraction.

Taking computational constraints into account

There is a growing demand for visual representations using the powerful graphical displays that are currently available. Teachers are asking for specific colour and other visual attributes at the same time they are defining the feedback they that suggest after a student request. In other words, features belonging to the domain are often mixed with requests related to the presentation of the subject matter and with visual constraints. To help teachers categorise their requirements and developers implement them at the right level, we propose a four level knowledge model as shown in Figure 6. The levels include domain, conceptual, presentation, visual, they are described in (Bernat & Morinet-Lambert, 95).

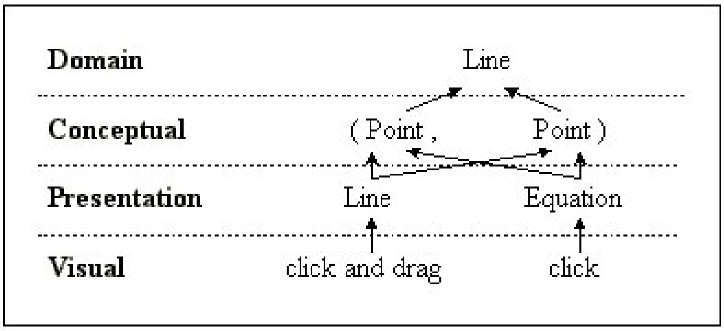

Figure 6: A line through the four level knowledge model

The domain level is a theoretical level that allows one to describe the subject matter to be taught independently from any symbolic-level representation. It is similar to Newell's knowledge level (Newell, 82). Teachers are describing their activities at this domain level when they use natural language, gestures and manipulations of currently used devices such as figures or experimental settings. The conceptual level defines a realisation of the domain, based on design choices, concerning concepts and relations. The represented knowledge is reified at the presentation level. At this level we consider the external view of domain concepts. At this level multiple representations of domain knowledge are required because teachers use different representations in different contexts. For instance a line can be presented as a graphical line or as an algebraic equation. Finally, the visual level is the graphical interface level where one makes implementation decisions such as whether to use direct manipulation.

Other views of teaching expertise

We have discussed generic expertise and subject-matter dependant expertise, presentation expertise and domain-based expertise, traditional classroom expertise and computer-based teaching expertise. Teachers who are involved in authoring educational material do not make clear distinctions between presentation expertise (the kind of display they want) and production expertise (the way they build it with existing tools), but production skills are also part of teaching expertise. Another leading thread for analysing teaching expertise is interaction or tutoring dialogue including how to address errors. Handling learner's mistakes and providing appropriate remediation requires far more expertise than only scaffolding the problem solving process. Whatever the view, we need methods and tools to acquire, model and represent this expertise.

ACQUIRING, MODELLING, REPRESENTING TEACHING EXPERTISE

Eliciting teaching expertise

Making expertise available is acknowledged as being a bottleneck in the implementation of any knowledge-based system. The tight interrelation existing between professional expertise and human involvement that characterises teaching activity makes the endeavour even harder for teaching expertise. All existing techniques must be used, including traditional expertise transfer with the help of knowledge engineers. Murray provides an analysis of the knowledge acquisition problem, including general teaching expertise in (Murray, 97). Among others, the DOCENT project (Winne & Kramer, 88) intended to provide teachers with means for expanding their knowledge about teaching. But, for designing those domain-related environments that enable teachers to adapt learning material to their local constraints, we suggest a bottom-up approach and a spiral development process. Let teachers start with using running pieces of educational software that are currently successful in the domain or for the purpose for which they were designed. Let us observe the activities that are set up, let us extract common features, take into account the needs of the instructor and implement them into a larger framework embedding the given tool and providing a set of additional functions. Such a process, which incorporates feedback, allows us to focus on teaching practice and maintains the teacher-user at the heart of the software development loop. The results of the observations should be carefully recorded and more often reported in research papers. From a knowledge management viewpoint, this step should produce a “knowledge book”.

Modelling teaching knowledge

Once a significant set of teaching expertise becomes available, it is necessary to represent it. A first level of expression should use educational ontologies. Indeed, in Knowledge Engineering, an ontology is a system of primitive vocabulary and concepts, which aims at defining and structuring the domain. It is used for further formalisation and implementation of system components. General instructional ontologies have already been proposed, with a focus on task ontologies in (Mizoguchi & al., 97). The IEEE P1484 initiative and the ARIADNE project (Forte & al, 97) propose similar approaches. We share the view expressed there about the need of conceptualisation and standardisation for knowledge representation in ITS. But in our view, the development of subject matter related teaching ontologies is as important as the development of general instructional ontologies. Subject matter ontology will on one hand borrow concepts from more general ones either generic or domain related and will on the other hand represent the local expertise. Several ontologies will model several viewpoints that have to be taken into account for describing teaching knowledge, they will enhance conceptualisation, standardisation, sharing and reuse.

For the development of Calques 3D we would have liked to rely on ontologies about teaching geometry in secondary schools. Using the different views we have suggested for describing teaching expertise, such ontologies could include geometrical concepts ontologies, visualisation ontologies, learners activities ontologies, etc. Making available the ontologies on which Calques 3D relies would provide the teachers with a “declarative representation of what the system knows” as suggested by Mizoguchi. Moreover, representing the national curriculum in geometry with the same ontology would show whether the proposed environment is curriculum compliant or not and satisfy a request of many teachers.

In the same way, ontologies could be used to describe the learner model and the interaction modules. I am convinced that the use of ontologies will allow a better view of the expertise embedded into systems.

Architectures and representations

Teaching knowledge based on shared ontologies should then be implemented within educational pieces of software by using all available knowledge representation techniques. Production rules have been often used for representing domain problem solving capabilities as well as teaching strategies. In the same way, students' abnormal behaviours were represented through "mal-rules". Murray and Woolf (Murray, & Woolf, 92) suggested PANs (parameterised action networks) which are similar to procedural flowcharts and provide a more manageable representational formalism than production rules. Other attempts include van Marcke's GTE (Van Marcke, 92) which relies on teaching expertise represented in terms of instructional tasks, instructional methods and instructional primitives. Object technology is also available and currently used. Other possible paradigms include teaching through case-based reasoning, especially for professional training on the workplace.

The embedded knowledge must be organised according to a global architecture. The four modules architecture has been replaced by more sophisticated schemas. More recently, agent-based architectures have been proposed (Kearsley, 93), they are more flexible and enhance modular developments as well as modules reusability. Among the many kinds of agents that can be used in an ITS, more attention has been paid to pedagogical agents, with the aim of providing templates for such agents. Another interest of the agent architecture is to consider human agents as well as artificial agents and thus to represent "artificial companions" as well as human learners and tutors.

The representations and architectures used up to know aimed at implementing a single learning environment. But the context is changing with the availability of Web-based learning environments. Since the prospect of developing a complete learning environment is mostly unrealistic, one promising direction is to encourage the development of small mutually compatible components, so that any individual author's investment is limited. Thus new architectures should support platforms where independent components should co-operate to fulfil the learners needs. Ideally, standards should be promoted which allow components to be developed independently, with reasonable expectation that the resulting components interact properly when assembled. Communication and co-operation between software agents at the knowledge level is a crucial issue in computer-based educational software design (Macrelle, & Desmoulins, 98). Moreover, deploying ITS on the Web requires a higher level of adaptativity, given the wide variety of potential learners. In recent conference proceedings (AIED 97, ITS 98) several authors have proposed milestones through conceptual frameworks to describe these new educational settings

CONCLUSIONS AND FUTURE TRENDS

Teacher's knowledge becomes the core problem for the design of ITS; thus it is crucial to make this teacher's knowledge more explicit and to embed it into computer-based learning environments for improving their performances and flexibility. My claim about teaching expertise is twofold: first, it is much more than what is called tutoring knowledge in many existing ITS prototypes, second an equal attention should be paid to general teaching expertise and to subject matter teaching expertise. Moreover, there is an increasing need to fully involve practising educators in the design and evaluation of the systems. Teacher participation in tutor design will greatly enhance practicality of the tutor and the relevance of the instructional content, and it should at the same time allow teaching expertise capture.

But we must realise that in practice many teachers will not have the resources and motivation to be involved in design, as only a few teachers write textbooks. However all should be given the opportunity to customise a computer-based environment to their local constraints and strategies. As this opportunity will not be always used, the availability of a default mode is crucial.

Even if we involve more teachers in such incremental design processes, we need to acquire knowledge and to get feedback from many others. In the same way as companies are organising knowledge management, the education community should encourage the creation of knowledge networks for sharing teaching techniques and strategies and disseminating success stories among peers and for teacher training. User-friendly environments should be provided for describing teaching tasks and learning activities. Descriptive frameworks would obviously differ from existing authoring environments. From such descriptions we could learn a lot on teaching knowledge. The Web provides a potential technical support for such initiatives, we should provide the models and guidelines that enable the process to start.

Furthermore, research is needed to better understand the cognitive teaching skills and to model teachers' activities. Studies can be conducted according a large number of views. For instance, Baker and Lund (Baker & Lund, 99) propose to model the teacher's activity based on viewing it as a complex explanation process. By analysing teachers reflecting on their teaching dialogues, they show the kind of student knowledge teachers attempt to acquire and how they acquire it. The Human Communication Research Centre in Edinburgh (UK) is investigating the impact on learning of seeing other learn – what they call vicarious learning (McKendree & al., 1998). Therefore, they study vicarious dialogues and intend to provide logical models as a framework for better understanding educational communication.

Distance learning and lifelong learning are becoming major issues. Computer Supported Collaborative Environments have been given increasing interest in recent conferences. New models and architectures are needed to implement such environments including human companions as well as software agents. Among these new agents, the design of animated agents needs the capture and sequencing of space, gesture and mimic behaviour. A new kind of teaching expertise is arising there, but is it still teacher's expertise? As we are developing more artificial artefacts, we must not forget the importance of human factors and of social interactions among students and teachers.

Acknowledgements

The author gratefully acknowledges the contribution of all the past and present members of the LORIA research team in Computers and Education, as well as R. Aiken for his valuable comments and his review of the final English version of the paper.

References

Anderson, J.R. & Pelletier, R.(1991), A development system for model tracing tutors. in *Proceedings of the International Conference of the Learning Sciences*, Evanston, IL.

Bernat P., Morinet-Lambert J. (1995) Spécificités et modélisation de l'interaction dans un EIAO, *Quatrièmes Journées EIAO de Cachan*, Eyrolles, Paris, France, 208-220

Clancey, W.J. (1979) Tutoring rules for guiding a case method dialog. *International Journal of Man-Machine Studies*, 11(9), 25-49

Clancey, W.J. (1986) From GUIDON to NEOMYCIN and HERACLES in twenty short lessons, *AI Magazine*, 7(3), 40-60

Clancey, W.J. (1987) *Knowledge-Based Tutoring: The GUIDON Program*, MIT Press, Cambridge, Massachusetts

Clancey, W.J. (1992) Guidon-Managed Revisited: A Socio-Technical Systems Approach, in *Proceedings of ITS 92*, 21-36, Springer Verlag

Direne, A. (1997), Designing Intelligent Systems for Teaching Visual Concepts, *International Journal of Artificial Intelligence in Education*, 8, 44-70.

Forte, E., Wentland, M., & Duval E. (1997) The ARIADNE Project (part 1 & 2): Knowledge Pools for Computer-based and Telematics-supported Classical, Open and Distance Education, *European Journal of Engineering Education*, 22(1&2), 61-73 & 153-165

Gavignet E. (1991) Environnement de construction de système d'apprentissage: Une modélisation de la connaissance pédagogique, Thèse de l'université de Nancy 1.

Gavignet, E. & Grandbastien, M. (1992), Représentation des connaissances dans un générateur de systèmes d'EIAO, in *Proceedings of ITS 92*, Lecture Notes in Computer Science n°608, Springer Verlag

Gavignet E. (1994) Instructional expertise in ECSAI, *in Proceedings of the International Conference CALISCE'94*, Dessales J.L. ed., TELECOM Paris, France.

Grandbastien M. & Gavignet E. (1994) ECSA: An Environment to Design and Instantiate Learning Material, *in Design and Production of Multimedia and Simulation-based Learning Material*, de Jong T. & Sarti L. (Eds). Kluwer Academic Publishers.

Hartley, J.R. & Sleeman, D.H., (1973), Toward Intelligent Teaching Systems, *International Journal of Man-Machine Studies*, 5, 215-236

Ikeda, M. & Mizoguchi, R. (1994). FITS: A Framework for ITS - A computational model of tutoring. *Journal of Artificial Intelligence in Education* 5(3), 319-348

Joolingen, W.R.van & Jong, T. de, (1996) Design and implementation of a simulation-based discovery environments: the SMISLE solution, *Journal of Artificial Intelligence in Education*, 7(3/4), 253-27

Kearsley, G. (1987) *Artificial Intelligence and Instruction*, Addison & Wesley.

Kearsley, G. (1993) Intelligent agents and instructional systems: implications of a new paradigm, *Journal of Artificial Intelligence in Education* 4, (4), 295-304

Lawler, R. & Yasdani, M. (1987) (Eds.) *Artificial Intelligence and Education*, Norwood, Ablex.

Leinhardt, G., Greeno, J.G. (1986), The cognitive skill of teaching, *Journal of Educational Psychology*, 78(2), 75-95.

Lund K. & Baker, M. (1999). How teachers determine what students know: collaborative interpretation of students' computer–mediated collaborative problem-solving interactions, in *Proceedings. of AI-ED 99* (to appear), IOS Press.

Macrelle, M. & Desmoulins, C. (1998) Macro-Definitions, a Basic Component for Interoperability between ILEs at the Knowledge Level: Application to Geometry, in *Proceedings of ITS 98*, Lecture Notes in Computer Science n°1452, Springer Verlag

Major, N. & Reichgelt H (1992) COCA: A Shell for Intelligent Tutoring Systems, in *Proceedings of ITS 92*, Lecture Notes in Computer Science n° 608, Springer Verlag.

Major, N. (1995). Modelling Teaching Strategies. *Journal of Artificial Intelligence in Education*, 6(2/3), 117-152.

McKendree, J., Stenning, K., Mayes, T., Lee, J., & Cox, R. (1998) Why observing a dialogue may benefit learning, *Journal of Computer Assisted Learning*, 14(2), 110-119

Mizoguchi, R, Ikeda, M, & Sinitsa, K. (1997) Roles of Shared Ontologies in AI-ED Research, Intelligence, Conceptualisation, Standardisation and Reusability, in *Proceedings of AI-ED 97 conference*, IOS Press.

Murray, T. (1997) Expanding the knowledge acquisition bottleneck for intelligent tutoring systems. *International Journal of Artificial Intelligence in Education.* 8(3-4), 222-232.

Murray, T. (1999) Authoring Intelligent Tutoring Systems: An analysis of the state of the art, *International Journal of Artificial Intelligence in Education*, 10(1),

Murray, T., Woolf, B.P. (1992) Tools for Teacher Participation in ITS Design, in *Proceedings of ITS 92*, Lecture Notes in Computer Science n°608, Springer Verlag

Newell, A. (1982) The Knowledge Level, *Artificial Intelligence*, 18(1), 87-127

Ohlsson, S. (1987) Some Principles of Intelligent Tutoring. In Lawler & Yazdani (Eds.), *Artificial Intelligence and Education*, Volume 1. Ablex: Norwood, NJ, pp. 203-238.

Sleeman, D. & Brown, J.S. (Eds) (1982) *Intelligent Tutoring Systems*, Academic Press, London.

Shute, V.J. & Psotka, J. (1995) Intelligent Tutoring Systems: Past, Present, and Future, in Jonassen D. (Ed), *Handbook on Research on Educational Communications and Technology*, Scholastic Publications

Van Marcke, K. (1992) Instructional Expertise, in *Proceedings of ITS 92*, Lecture Notes in Computer Science n°608, 235-243, Springer Verlag

Van Labeke, N., Aiken, R., Morinet-Lambert, J, & Grandbastien, M. (1999) IF "What is the Core of AI & Education?" is the Question, THEN "Teaching Knowledge" is the Answer, in *proceedings of AI-ED 99 conference* (to appear), IOS Press.

Wenger, E. (1987). *Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge*, Morgan Kaufmann.

Winne, P. H. & Kramer, L.L. (1988). "Representing and Inferencing with Knowledge about Teaching: DOCENT." an artificially intelligent planning system for teachers Proceedings of ITS88, Montreal, Canada