Student Modeling in Orthopedic Surgery Training: Exploiting Symbiosis between Temporal Bayesian Networks and Fine-grained Didactic Analysis

Vu Minh Chieu [a], Vanda Luengo [b], Lucile Vadcard [c], Jérôme Tonetti [d]
[a] *School of Education – University of Michigan, United States*
[b] *Laboratoire LIG – UMR CNRS/UJF/INPG/INRIA/UPMF 5217, France*
[c] *Laboratoire des Sciences de l'Education – University of Grenoble 2, France*
[d] *Orthopedic Department, Grenoble Hospital, France*
vmchieu@umich.edu, vanda.luengo@imag.fr, lucile.vadcard@upmf-grenoble.fr, JTonetti@chu-grenoble.fr

Abstract. Cognitive approaches have been used for student modeling in intelligent tutoring systems (ITSs). Many of those systems have tackled fundamental subjects such as mathematics, physics, and computer programming. The change of the student's cognitive behavior over time, however, has not been considered and modeled systematically. Furthermore, the nature of domain knowledge in specific subjects such as orthopedic surgery, in which pragmatic knowledge could play an important role, has also not been taken into account deliberately. We believe that the temporal dimension in modeling the student's knowledge state and cognitive behavior is critical, especially in such domains. In this paper, we propose an approach for student modeling and diagnosis, which is based on a symbiosis between temporal Bayesian networks and fine-grained didactic analysis. The latter may help building a powerful domain knowledge model and the former may help modeling the learner's complex cognitive behavior, so as to be able to provide him or her with relevant feedback during a problem-solving process. To illustrate the application of the approach, we designed and developed several key components of an intelligent learning environment for teaching the concept of sacro-iliac screw fixation in orthopedic surgery, for which we videotaped and analyzed six surgical interventions in a French hospital. A preliminary gold-standard validation suggests that our diagnosis component is able to produce coherent diagnosis with acceptable response time.

Keywords. Student Modeling, Dynamic Bayesian Networks, Intelligent Support, Didactic Engineering, Medical Education, Computer-Based Simulations.

INTRODUCTION

Knowledge Modeling and Student Modeling

In the past thirty years, research results in cognitive science have been exploited for student modeling in problem solving, as evidenced by a significant number of cognitive approaches (Webber, 2004; Mayo & Mitrovic, 2001; Murray, 1999). Many studies have been done within the context of teaching fundamental subjects, for example, geometry (Anderson, Boyle, & Yost, 1986), (Webber, 2004), algebra (Koedinger, Anderson, Hadley, & Mark, 1997), physics (Albacete & VanLehn, 2000), computer programming language (Anderson, Farrell, & Sauers, 1984). The nature of domain knowledge and the complexity of the learner's cognitive behavior, especially in a number of specific subjects (e.g., in medical education), however, have not been considered carefully. Firstly, the *tacit pragmatic knowledge* (obtained by experience) plays an important role for both the expert teacher and the novice learner during a problem-solving process. This tacit knowledge refers to "work-related, practical know-how that typically is acquired informally as a result of on-the-job experience, as opposed to formal instruction." (Wagner, Sujan, Sujan, Rashotte, & Sternberg, 1999, p. 157). While observing a number of medical interventions in a French hospital, we realized that sometimes the expert teacher and the novice student, when confronting a specific problem, used pragmatic knowledge to elaborate an original solution to the problem encountered, which could not have been defined before. Secondly, the student's cognitive behavior we observed in those interventions is complex. A skillful learner, even a domain expert, often makes several attempts before arriving at an acceptable solution: he or she may make an error and then retry to correct the error several times. Thus, from an observer's point of view, one may need to consider a sequence of actions from the learner to be able to diagnose his or her cognitive state and behavior accurately.

A number of researchers (Kodaganallur, Weitz, & Rosenthal, 2005; Luengo, Mufti-Alchawafa, & Vadcard, 2004; Webber, 2004; Mitrovic & Ohlsson, 1999) have argued that it is important to check the consistency of the student's solution with domain constraints (i.e., local consistency checks) rather than to compare the student's solution with the domain expert's *a priori* normative solution (Ohlsson, 1992). This idea is particularly useful for building tutoring systems for the kind of specific domains mentioned in the previous paragraph, because in those domains there may have many different solutions to a given problem, some of which being elaborated in action by the domain expert. So, the first question we address in this paper is concerned with exploiting and analyzing different kinds of domain knowledge, especially tacit pragmatic knowledge, in order to build a robust domain model, which is critical for student modeling and diagnosis (Weber & Brusilovsky, 2001). Tacit pragmatic knowledge is often not explicitly explained in theoretical courses or reference books (Vadcard & Luengo, 2005). To answer the first question, we argue for a fine-grained "didactic" analysis (Pastré, 1997). Didactic (an originally francophone term) designates the study of teaching and knowledge acquisition in different subject domains. Didactic is thus different from pedagogy by the central role of the subject domain contents and by its epistemological dimension (i.e., the nature of knowledge to be taught). To some extent, didactic analysis is similar to cognitive task analysis (Clark & Estes, 1996). Both of them seek to better understand the subject being taught, so as to better devise instructional situations for students. The major difference between them is the analysis protocol: cognitive task analysis is often done by observing *highly skilled practitioners* and describing the precise activities that are required to perform a complex task, whereas didactic analysis is often performed in instructional or apprenticeship settings in which, for example, a novice learner interacts with *an expert teacher* to solve problems. Unlike cognitive task analysis, which tries to describe the problem-solving process of domain experts as completely as possible and to seek pedagogical implications from that process, didactic analysis seeks different kinds of knowledge needed for successful teaching directly from observing instructional or apprenticeship settings. Hence, didactic analysis may reveal special kinds of knowledge such as pedagogical content knowledge (Shulman, 1986) that cognitive task analysis might not be able to produce because the domain experts might not have those kinds of knowledge in mind or not reveal them explicitly in the context of cognitive task analysis. Special kinds of knowledge such as pedagogical content knowledge are useful for the design of a learning environment (Shulman, 1986).

The second question is concerned with exploiting suitable techniques in artificial intelligence to model and "diagnose" the student's knowledge or cognitive state at a given time and his or her cognitive behavior over time. The first diagnosis is important and very common in many traditional ITSs (Wenger, 1987). We believe that the second diagnosis about cognitive behavior could be also important, because it may help generate better feedback for the student. A way to do those kinds of diagnosis is to analyze the student's interactions with the interface of the learning system such as a computer-based simulation (Luengo, Mufti-Alchawafa, & Vadcard, 2004). Diagnosing the student's knowledge and cognitive behavior, however, is not easy because it is difficult to know what happens exactly in the mind of an individual when he or she is learning a concept or solving a problem (Sasse, 1991). Bayesian networks offer a useful technique for modeling under uncertainty (e.g., about students' cognitive state), and according to Mayo and Mitrovic (2001) it has been adopted in many applications, including ITSs. Considering the complexity of the learner's cognitive behavior over time (e.g., the learner's correction process while he or she is constructing a solution) in specific domains, as mentioned previously, *temporal* (or dynamic) Bayesian networks (Russell & Norvig, 2009; Ghahramani, 1998) could be an effective means.

Contribution

In this paper, we propose a new temporal-Bayesian-network-based student model in which we emphasize the importance of explicitly diagnosing the student's knowledge state and cognitive behavior over time by modeling the *temporal* dimension. It is important because it may help an ITS better "understand" how and why the student makes an error, so as to generate better feedback for him or her. The new student model is strengthened by a fine-grained didactic analysis, which permits the construction of a powerful domain knowledge model. We applied both models to build a number of key components of an intelligent learning environment for the problem area presented next about the sacro-iliac screw fixation in orthopedic surgery. The design of the learning environment has been grounded in a theory of didactic situations proposed by Brousseau (1997[1]) and based on an analysis of surgical interventions.

Context for the Example

The current research has been done in the context of the TELEOS (Technology Enhanced Learning Environment for Orthopedic Surgery) project (Mufti-Alchawafa & Luengo, 2009; Vadcard & Luengo, 2005; Luengo, Mufti-Alchawafa, & Vadcard, 2004). TELEOS research team includes computer scientists and engineers, psychologists, educators, and surgeons. The aim of the project is to exploit, analyze, and model different kinds of knowledge in orthopedic surgery, especially pragmatic knowledge, to design and build an intelligent simulation-based learning environment for professional learners (i.e., resident junior surgeons). The motivation of the project has been that in a traditional approach the student interacts with an experienced surgeon to learn operative procedures, learning materials being patient cases and cadavers. This approach principally presents a number of problems, as follows: it requires one surgeon for one student, it is unsafe for the patient, and cadavers must be available. Several authors have claimed that the introduction of computers in medical education could deal with those problems (Eraut & du Boulay, 2000), but on the condition that real underlying educational prin-

[1] This reference refers to a translation of a collection of Brousseau's work originally published in French (1978, 1982, 1988, etc.).

ciples are integrated (Lillehaug & Lajoie, 1998). In particular, the importance of individual feedback has been stressed (Rogers, Regehr, Yeh, & Howdieshell, 1998). We believe that, to provide appropriate feedback to the learner, it is crucial to build a robust student model, which is the main concern of this paper.

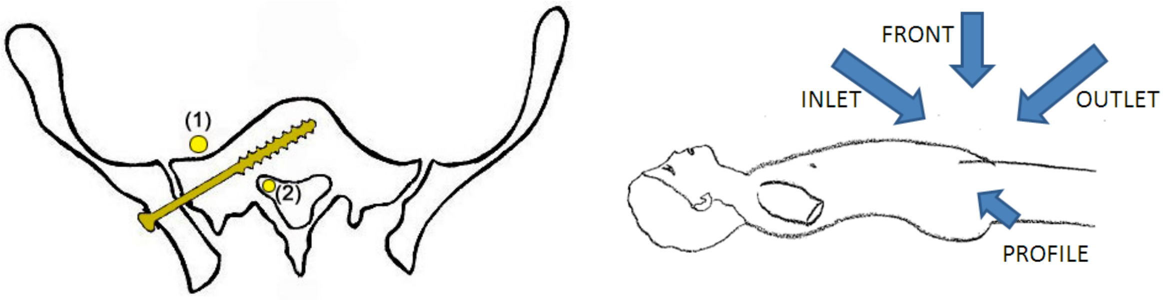

Fig. 1. Sacro-iliac screw position (left) and the four used X-rays (right).

An example for the motivation of the project is percutaneous sacro-iliac screw fixation (Tonetti, Carrat, Blendea, Merloz, & Troccaz, 2003), which allows posterior lesions of the pelvic ring of the hip bone to be fixed (Fig. 1). It could be summarized as follows. The surgeon first inserts a guide pin in the bone through the skin (percutaneously, i.e., without incision). He makes the pin progressing in the bone, taking several X-rays (Fig. 1) to validate the pin course at different steps of the progression. The four X-rays allow him or her to reconstruct a complete vision of the position of the pin, in relation to the pelvis. If he or she recognizes any problems in those views, he or she restarts the operation process, taking another pin and correcting its entry point and/or direction. During this phase (i.e., pin insertion), the surgeon can make several attempts. Once the pin's trajectory gives satisfaction, the screw fixation phase will be performed: a screw is inserted along the pin, which will make the right bones' compression or maintaining for the treatment of the fracture. Last, the pin is taken out and one suture point is made to close the pin's entry point. The main danger of the percutaneous technique is a screw course outside of the bone with risk of injury of nerves (see (1) and (2) in Fig. 1).

In France, the training of sacro-iliac screw fixation is usually organized into a theoretical session (involving the acquisition of theoretical knowledge such as definitions of concepts) and a practical session (involving the informal acquisition of pragmatic knowledge with costly one-to-one assistance of an expert, e.g., applications of concepts in a variety of real clinical cases). In earlier work (Vadcard & Luengo, 2005), we showed the importance of a bridge between these two sessions: the use of an intelligent learning environment as an intermediate phase of learning, which provides the learner with an operative dimension of knowledge before the real situation.

Structure of the Paper

Firstly, we introduce the architecture of our learning environment and the theoretical framework behind the design of the learning environment. Secondly, we present the method and results of our didactic analysis, and we show the student model and the diagnosis component. Thirdly, we discuss related work and our approach, as well as a preliminary gold-standard validation of our approach. Finally, we make conclusions and we show several promising directions for future research.

SYSTEM ARCHITECTURE

Fig. 2 shows a simulation-based architecture of our learning environment. We reused an open-source Java multi-agent platform (JADE, 2006) to implement the learning environment; this platform allowed us to easily integrate different software components developed by different members in our research team. Here are the main agents: a tracing agent (see (1) in Fig. 2) dealing with a simulation for sacro-iliac screw fixation and with actions and traces produced by the individual student while interacting with the simulation, a diagnosis agent (see (2) in Fig. 2) for student modeling and diagnosis, a didactic decision agent (see (4) in Fig. 2) to decide which feedback should be presented to the student (according to his or her current knowledge state and cognitive behavior), a clinical cases agent (see (5) in Fig. 2) to help the student examine real clinical cases, an online course agent (see (6) in Fig. 2) to direct the learner, when necessary, to suitable part(s) of a theoretical course. The knowledge model (see (3) in Fig. 2) is a key component to make the diagnosis agent and the didactic decision agent work. We detail this knowledge model in the sections about didactic analysis and about student modeling and diagnosis. We developed and subjectively (i.e., without actual students) validated the tracing agent, the diagnosis agent, the clinical cases agent, and the online course agent. We have been developing the remaining didactic decision agent and we shall integrate it into the multi-agent platform later. Although in the present paper we concentrate on student and knowledge modeling, in the following paragraphs we briefly describe all of the main agents to help the reader better understand how the diagnosis agent works and why it is important.

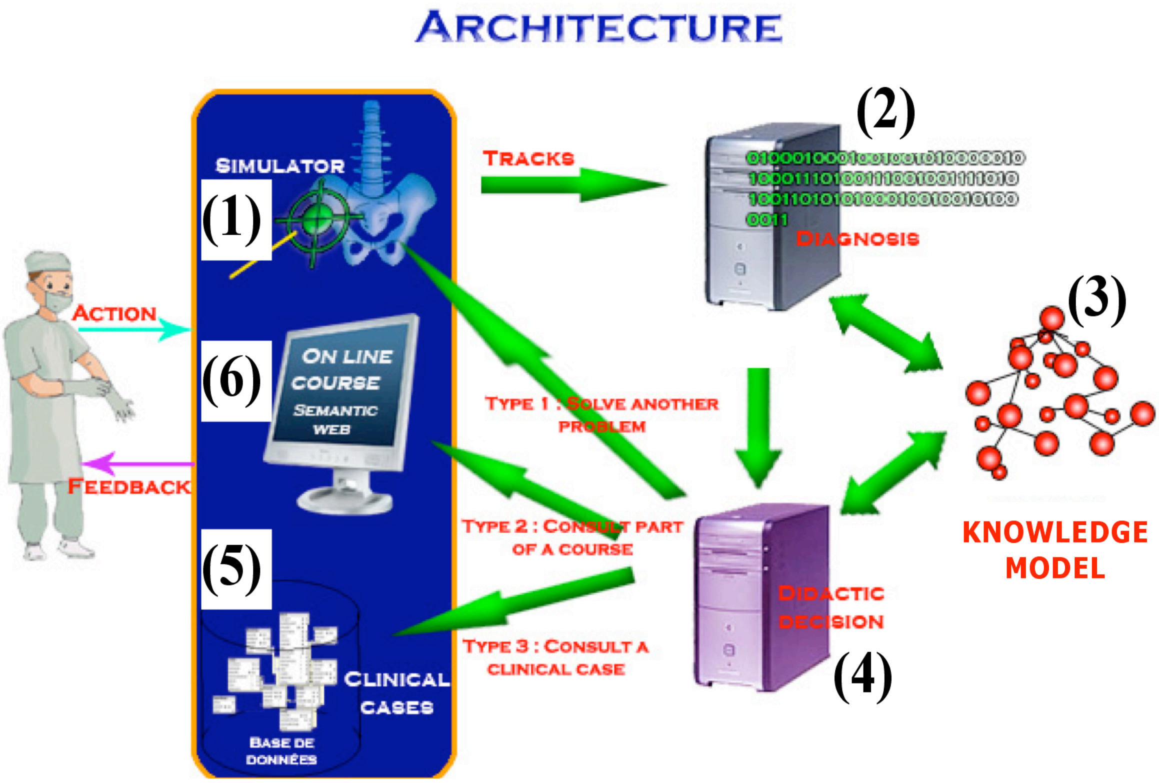

Fig. 2. Global architecture of TELEOS.

The development of the TELEOS research project has been based on the results of the VOEU (Virtual Orthopedic European University) research project (Vadcard, 2003). The latter has provided the clinicians with various tools supporting the learning of clinical skills. Those tools include a web-based 3D simulation for sacro-iliac screw fixation[1], an online theoretical course[2], and a clinical cases database. A main contribution of TELEOS to VOEU has been the research of domain knowledge modeling and student modeling for the design and development of an intelligent learning environment. The current web platform is not powerful enough to run all components of such learning environment. Hence, we developed a Java-based standalone application, except for the part of the online course. We also needed to build a new simulation component because the old version is not able to output the learner's actions and traces and it is impossible for other software agents to change parameters of the old simulation system.

Concerning the teaching and learning of sacro-iliac screw fixation, analysis of surgical interventions allowed us to determine that the most crucial phase is the pin insertion. During this phase most of the required knowledge is used: anatomy, reading of X-rays and interpretation of images, mental representation of the pelvis, requirements about the pin trajectory, ... Let us assume that a surgeon student, familiar with theoretical concepts of sacro-iliac screw fixation, uses the intelligent learning environment to develop the ability to solve various sacro-iliac screw fixation situations. He or she is presented with one of those problems in the new Java-based simulation (Fig. 3). In this simulation, 3D representations of pelvis bones and X-rays have been constructed from scanning bones of real patients. The 3D pelvis representation (the pelvis object in Fig. 3), like in real situations, is skinned so that the student cannot see the internal structure of the pelvis. The landmarks (Fig. 3) are there to help the student recognize important parts of the pelvis object. The learner has to position a pin (Fig. 3) and to introduce it in the simulated pelvis. His or her actions include positioning the pin, orientating the pin, advancing the pin, removing the pin and changing its orientation and/or entry point. At any time the learner can ask for an X-ray control; the four available orientations (inlet, outlet, lateral, and face, see Fig. 1) correspond to the orientations used by the surgeon in real situations. After the learner validates his or her solution (by clicking on a "Confirm" button shown at the bottom right of Fig. 3), the simulation component, similar to many computer-based simulations, provides the learner with some information feedback about his or her problem-solving process such as the number of attempts, the number of extra-osseous trajectories (i.e., outside of the bone with risk of injury of nerves, see also Fig. 1) validated, the number of X-rays taken. A "transparency" slider, which can be used to make the skin of the 3D pelvis representation disappear, is also provided for the learner to visualize his or her final pin course.

---

[1] See http://www-sante.ujf-grenoble.fr/SANTE/voeu/visang/exerci/intro.htm, developed by using VRML (Virtual Reality Modeling Language).

[2] See http://www-sante.ujf-grenoble.fr/SANTE/voeu/visang/vissage.htm.

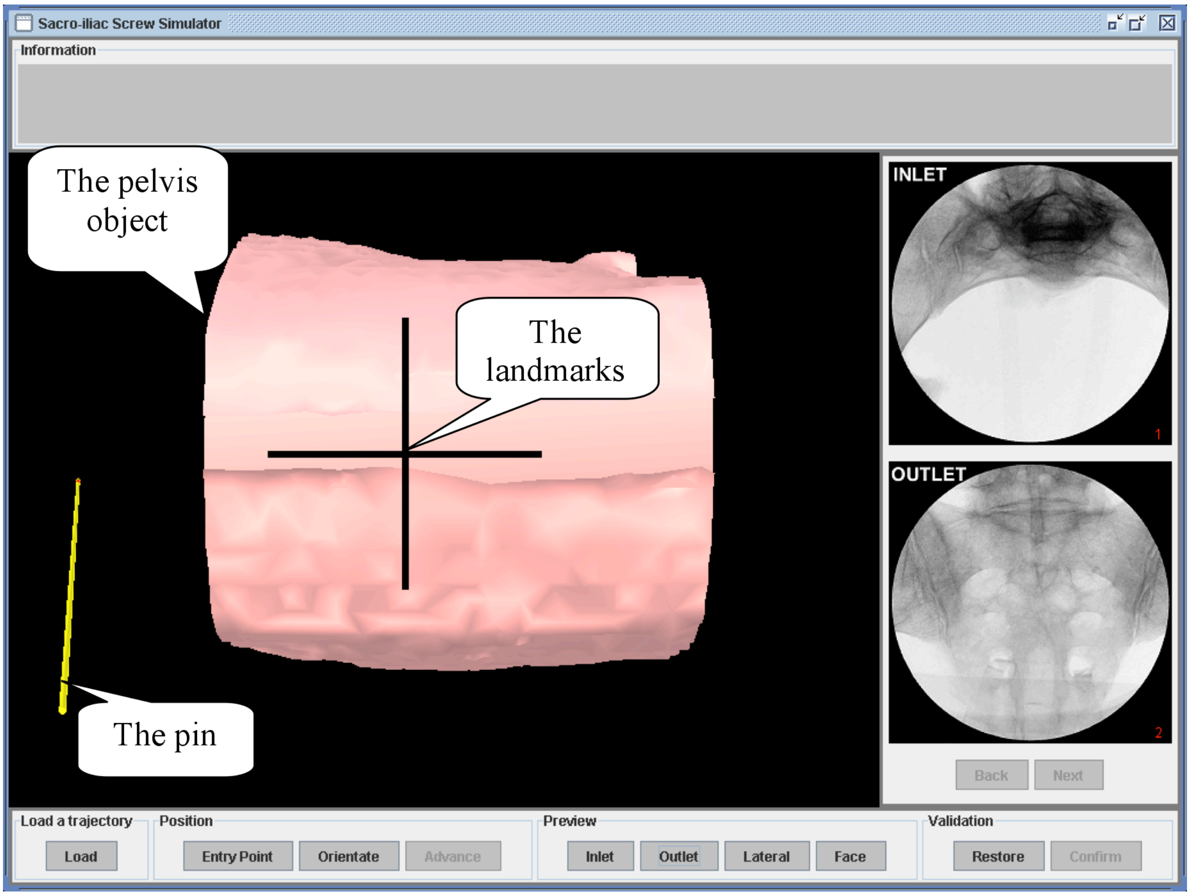

Fig. 3. A Java-3D simulation interface for sacro-iliac screw fixation.

Additionally, when necessary the system will present the student with one or several suggestions to enhance the student's learning. A suggestion can be: (1) another problem to solve, (2) web pages in the theoretical course to explore, (3) clinical cases to study. To make the system able to produce relevant suggestions for the student, firstly the tracing agent captures the student's traces whenever he or she does an action. Those traces include the student's current action (e.g., advancing the pin) and the current position of the pin, in relation to a number of critical points, lines, areas in the pelvis bone (e.g., see `(1)` and `(2)` in Fig. 1), on the four available orientations (e.g., the distance between the pin and the anterior cortex of the lateral part of the sacrum on the inlet view). The tracing agent calculates those distances by analyzing X-ray images. Then, once the student's traces are available, the tracing agent sends them to the diagnosis agent, which will analyze them, by using a domain knowledge model, to update the student model (see `(3)` in Fig. 2)—the student model, the domain knowledge model, and the diagnosis agent are the principal concerns in the present paper, and we detail those components in the following sections. Finally, the diagnosis agent sends to the didactic decision agent the current state of the student model. The didactic decision agent (see Mufti-Alchawafa & Luengo, 2009; Luengo, Vadcard, Mufti-Alchawafa, & Chieu, 2007 for more details) will analyze the updated student model, by using the domain knowledge model, to make decisions about which kind(s) of feedback should be relevant to the individual student as well as about the content for each feedback. More specifically, the didactic decision agent asks the learner to solve another problem if it detects that he or she may need to master some pragmatic knowledge. In this case, the new problem will be presented to the learner in the simulation component. Additionally, the learner may be asked to examine one or several clinical cases that illustrate the post-consequences of pin trajectories that are similar to his or her current pin course. If the agent detects that the student may need to master some theoretical knowledge, it asks him or her to study a number of theoretical concepts in the online course. We have applied semantic-web techniques to improve the online course in the VOEU project so that our system is able to redirect the learner to precise and relevant parts of the online course (see Chieu, Luengo, Vadcard, & Mufti-Alchawafa, 2007 and Luengo & Vadcard, 2005 for more details). Although feedback is often provided to the student after he or she validates a solution, in some particular cases (e.g., if the diagnosis agent detects a serious lack of theoretical knowledge by the student) the didactic decision agent may decide to encourage him or her to go to the online course immediately and to read one or several web pages related to the detected error.

THEORETICAL FRAMEWORK

The theoretical framework of our research is grounded mainly in a theory of didactic situations in mathematics (Brousseau, 1997). A number of studies in mathematics education have been grounded in Brousseau's theory. For example, Hersant and Perrin-Glorian (2005) have examined didactic situations the teacher chooses, and how he manages classroom interactions and the students' work in the classroom and at home, so as to characterize a mathematics teaching practice in secondary schools. Margolinas and associates (2005) have built a model to study how in-service teachers acquire didactic knowledge from observation and reflection upon students' mathematical activity in the classroom. Although the work of Brousseau is concerned with mathematics teaching, we have kept a number of ideas and principles, as guidelines, for the design of our learning environment.

Conception, Misconception, Knowing, "Milieu", and Problem Solving

Brousseau (1997) has contributed the notion of the *milieu* as an important element for a theory of instruction. The milieu refers to the system counterpart to the student in a learning situation. The milieu in particular is both the target of the actions of the student and the source of feedback on those actions (Vergnaud, 1981). The learning environment described previously enables the student's *interactions* with the simulation system, which in turn can generate *feedback* and prompt future actions for the student. The point we make here is that the feedback is "calculated" and presented to the student in a manner consistent with principles of Brousseau's theory, as we illustrate below.

Brousseau's work has been grounded in the perspective of learning as adaptation, stemming from Piagetian epistemology (Piaget, 1985), which states that people learn by adapting their existing cognitive structures to the feedback provided on their actions by the environment with which they are confronting. Specifically, a fundamental hypothesis of the theory is that "[errors] are not only the effect of ignorance, of uncertainty, of chance, as espoused by empirist or behaviourist learning theories, but the effect of a previous piece of knowledge which was interesting and successful, but which now is revealed as false or simply unadapted." (Brousseau, 1997, p. 82). In other words, we assume the hypothesis (Balacheff & Margolinas, 2005; Brousseau, 1997; Confrey, 1990) that a *misconception* is of interest from a learning point of view if it shares the properties of a *knowing*[2]: it has a domain of validity otherwise it would not exist as such. So, the key difference between a misconception and a knowing is that for the former there exists a refutation that is known at least to an observer. For example, in the problem of sacro-iliac screw fixation after perceiving that the pin comes too near from one precise anatomic part of the pelvis bone on the inlet view, meaning that the pin is "too low" on the inlet view, the novice surgeon may decide to remove the pin and to move the entry point upwards, which is incorrect, due to the particular orientation of this X-ray. The student's knowing behind this decision may be derived from his or her perfect understanding of classical representations where directions are conserved, but when that knowing is applied to the previous situation it is no longer valid in the current domain and becomes erroneous.

The thesis of Brousseau (1997) states that some of those knowings likely to be falsified are necessary to learning: the learning trajectory of the student may have to pass by the (provisional) construction of erroneous knowings because the awareness of the reasons why a knowing is erroneous is necessary to the construction and understanding of a new knowing—this is a key point about adaptation in learning. In other words, a knowing is first of all the result of an adaptation of the learner to his or her "milieu." As a consequence of this adaptation, any knowing has a provisional character, or rather, any knowing could be revised, its domain of validity can be modified as the result of some perturbations. The notion of knowing and the notion of belief introduced by Paiva and Self (1995) for student modeling share the same critical assumption about that adaptation. The only indicators to state whether the learner's understanding is erroneous or not in a given situation are the learner's behaviors and productions that are consequences of the knowing the learner has constructed. That is why the tracing agent in our learning environment is built to capture the learner's actions (e.g., advancing the pin, remove the pin, change the entry point) and productions (i.e., the position of the pin course), which allow the diagnosis agent, by using the domain knowledge model, to deduce the learner's knowings. Fine-grained didactic analysis of surgical interventions we describe in the next section could be important to understand those knowings and their mobilization in situation as completely as possible, and thus to be able to build a robust domain knowledge model.

The interaction between the subject and the milieu as well as the relationship between the learner's behavior and the learner's knowing are complex. For example, in our case, the surgeons often need several to a dozen of attempts (making a wrong pin course, removing the pin, and changing the entry point and/or the direction of the pin to correct the pin course) before arriving at a correct solution. The point is that sometimes they may make wrong pin trajectories because of the nature of the surgical domain (e.g., it is difficult, even for a domain expert, to make a correct pin course at the first attempt with only assistance of X-rays), but not because of their misconceptions. Therefore, it could be essential to examine those interaction and relationship in a sequence of actions or events, instead of one at a time. In other words, considering the *temporal dimension* in student modeling is a key point to build the diagnosis agent mentioned earlier. For instance, in the previous situation about sacro-iliac screw fixation, one may understand the learner's knowings better if his or her behaviors and productions at different points in time are systematically analyzed together than if those behaviors

[2] To account for the French distinction between "*connaissance*" and *"savoir*" that Brousseau frequently used in his work (in a general context, both of them are often translated as "knowledge"), these terms have been translated as "knowing" and "knowledge". The former refers to individual intellectual cognitive constructs. The latter refers to socially shared and recognized cognitive constructs, which must be made explicit.

and productions are analyzed separately. More examples and details are showed in the section about student modeling and diagnosis to better clarify the importance of that temporal dimension.

In summary, according to Brousseau's theory, the milieu for the apprenticeship must be organized to foster active learning by producing relevant feedback to the learner's actions and productions. We assume that the system can produce relevant feedback for the apprenticeship if it reacts regarding an internal validation of the learner's solution process, that is, local consistency checks of the learner's actions and productions (Ohlsson, 1992). By relevant feedback, we mean not only a variety of opportunities (i.e., an online theoretical course, a clinical cases database, a set of problems) for the student to learn, but also the adequacy of the feedback content provided to the student, for example, which problem or which part of the online course would be relevant to help the student perform an optimal adaptation to his or her milieu (i.e., the simulation system), according to his or her current knowing state and cognitive behavior. Thus, student modeling and diagnosis and domain knowledge modeling are among the most important concerns in the design and development of our intelligent learning environment.

Framework for Domain Knowledge Modeling

For didactic analysis and domain knowledge representation, we adopted the cK¢ (conception, knowledge, and concept) model that provides a computational framework for didactic research (Balacheff & Gaudin, 2010). This model facilitates the analysis of the domain knowledge to be formalized and implemented in our learning system (see (3) in Fig. 2).

The aspect of the cK¢ model that concerns our work is the formalization of a *conception*, which is the instantiation of the knowing ascribed by a subject to a situation. The cK¢ model formalizes a conception by the following quadruplet: a set of problems (P); a set of operators (R) involved in the solutions of the problems from P; a representation system (L) allowing the representation of P and R; and a control structure (Σ). The first three components are the key features identified by Vergnaud (1991, p. 145) to characterize a concept. The fourth was introduced by Balacheff and Margolinas (2005): the elements of Σ allow the subject to decide whether an action is relevant or not, or to decide that a problem or sub-problem is solved. The crucial role of control elements in problem solving has already been pointed out (e.g., Schoenfeld, 1985).

The formalization of a conception by the previous quadruplet allows a characterization of the subject-milieu system: the representation system allows the formulation and the use of the operators by the active sender (the subject) as well as the reactive receiver (the milieu). The control structure enables the expression of the means the subject uses to justify the adequacy and validity of an action, as well as the criteria of the milieu for selecting feedback (Balacheff & Margolinas, 2005).

DIDACTICAL ANALYSIS

According to the cK¢ model presented previously, the aim of the didactic analysis is to identify the quadruplet (P, R, L, Σ) and relationships among its components at an operational level, so as to build a robust domain knowledge model. An expert in didactic science and an expert surgeon, both in the TELEOS team, were working collaboratively to do and validate this didactic analysis.

As previously explained, surgical education is constituted of two main periods: the initial learning period, during which theoretical knowledge is acquired, and the professional learning period, during which experience of real situations is progressively acquired, and knowledge becomes progressively operational. The expertise, that is the efficient treatment of various situations, is transmitted in a partly implicit mode to the novice. To build a robust domain knowledge model, it is important to make a number of these elements of expertise explicit. It is not easy, however, for the expert surgeon to describe the correct process of sacro-iliac screw fixation *completely*. For example, to validate a solution in a particular case (e.g., a patient with a very hard bone), the expert surgeon sometimes uses *pragmatic* knowledge, which has not been described in any research articles or theoretical courses. Here is an explication of an expert teacher to a novice learner about one of the X-rays, taken at the mid-course of a problem-solving process:

> [The pin] is a bit too much behind... you see, it is a bit too much behind, it should have been much more by here, but the entry point is ok, **we won't modify it, but do not pass over the midline, furthermore he [the patient] has got a very hard bone, so you don't need to have a very well anchored threading**. (bold-font text emphasized by the didactic analyst)

The bold-font text in the previous dialog extract shows a part of the surgeon's professional expertise, which allows him to validate the position of the pin which should have been considered as invalid by the prescription "a bit too much behind". One of the objectives of our learning system is to help the student master that kind of pragmatic knowledge. Situating this objective in the constructivist approach of learning described earlier, it is important to integrate into our domain model these elements of knowledge and their related validity domain, so as to be able to analyze the student's actions in terms of possible used knowledge, and to produce related feedback.

The previous kind of dialogs is common in novice-learner and expert-teacher apprenticeship contexts, but not in the context of cognitive task analysis. That is why didactic analysis is particularly important in modeling knowledge of specific domains such as the sacro-iliac screw fixation. In the following sub-sections, we detail our approach for the didactic analysis; then, we present the results of the analysis process.

Approach

The approach is essentially based on an approach proposed by Pastré (1997) for didactic analysis in professional education. According to Pastré's approach, the didactic analysis process is composed of two main consecutive phases: the preparation phase and the observation and interview phase. The aim of the preparation phase is to help the didactic analyst master background knowledge of the subject domain in order to prepare as best as possible for the observation and interview phase, which in turn helps him or her collect subject domain knowledge as optimally as possible. Both phases are performed in the context of instructional situations in order to better understand how subject domain knowledge is acquired and taught.

In the preparation phase, the TELEOS didactic analyst first studied a theoretical course provided by an expert surgeon-teacher and a certain number of external resources describing the subject domain. This study helped her master the basic of declarative and procedural domain knowledge: the different anatomic parts of the pelvis and their relations, the different possible pelvis diseases and their treatments, the main steps of the required surgical interventions. Then, she observed and videotaped a number of real surgical interventions. Every operation was performed by both an expert and a resident junior surgeon. Thus, the observation of interventions provided elements of both the expertise and its transmission. This study allowed the didactic analyst to look further into theoretical concepts, to observe the course of surgical operations (some of which may not be mentioned in theoretical courses), and to witness the difficulty and the importance of certain tasks. For example, she identified the process of translating the information given by an X-ray into actions to perform on the patient as being a crucial task. It requires a complex cognitive process: coupling different views to reconstruct a mental 3D representation of the bone, identifying the correspondence between the X-rays directions and the pelvis position (see Fig. 4), considering the correspondence between the real distances and the ones displayed on X-rays.

In the observation and interview phase, the didactic analyst performed a cognitive analysis of the novice-learner and expert-teacher apprenticeship during real interventions of sacro-iliac screw fixation. In the observation process, to collect data as maximally as possible, an assisted observation protocol (Leplat, 2000) was used: a camera to take videos, a microphone to take audios, and a monitor of the operating theatre to collect taken X-rays. After collecting and studying the set of data, according to a consecutive verbalization technique (Leplat, 2000), she interviewed both the surgeon-learner and the surgeon-teacher. The interview questions were mainly concerned with the clarification of how and why to do and validate actions during the problem-solving process. This kind of interview questions helped collecting a variety of tacit pragmatic knowledge used by the learner and the teacher. Here are several examples of those questions: Why did you do this action? How did you do to accomplish this operation? What did you examine on this X-ray? What are the clues that allowed you to validate (not validate) this X-ray? The didactic analyst videotaped and analyzed six surgical interventions of sacro-iliac screw fixation, and she used two of them for consecutive verbalization in order to deepen the analysis. Each element of knowledge brought into our domain knowledge model, thanks to those verbalization processes, was carefully associated with its domain of validity (in which cases this knowledge is applicable and in which cases it is not, and why).

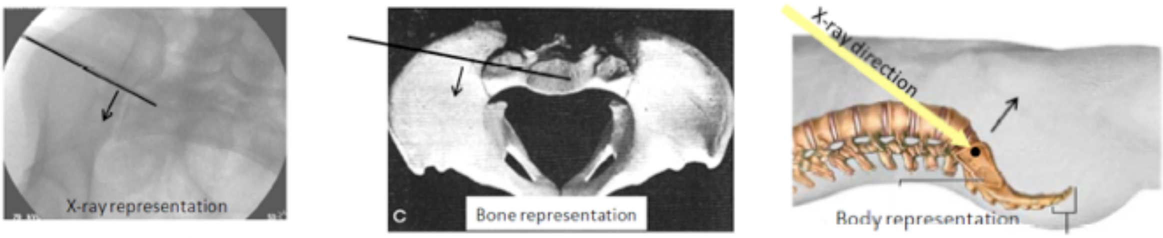

Fig. 4. Representations of a pin (two black lines and one black dot) and X-ray direction (black arrow).

Main Results

We present hereafter the results of the didactic analysis: the quadruplet (problems, operators, controls, representation system) defined earlier.

Problems (P)

In the didactic analysis process, we recognized that the teacher or learner's validation of a solution depends on the characteristics of the sacro-iliac screw situation he or she is solving, for instance, the type of the patient’s pelvis fracture, the quality of the patient’s bone, the nature of the problem-solving situation. Those characteristics are named as "didactic variables" (Brousseau, 1997).

Table 1
A subset of problems presented to the student

Problem ID	Fracture type		Bone quality			Declaration	
	Sacrum fracture	Pure disjunction	High density	Normal density	Low density	Define trajectory	Validate trajectory
P_A		X		X		X	
P_B	X			X		X	
P_C	X			X			X
P_D		X		X			X
P_E	X				X	X	
P_F		X			X	X	

Table 2
Operators used in a problem-solving sacro-iliac screw fixation

Action ID	Operator	Traces of the pin course
1	OP1: Introduce an entry point for the pin course	No information available
2	OP2: Orientate the pin	No information available
3	OP3: Advance the pin	No information available
4	OP5: Take an inlet view	The pin comes too near from the anterior cortex of the lateral part of the sacrum on the inlet view, meaning that the pin is too low on the inlet view (the position of the pin is incorrect)
5	OP6: Take an outlet view	The position of the pin is correct on the outlet view
6	OP4: Restore the pin	No information available
7	OP1: Introduce an entry point for the pin course	No information available
8	OP2: Orientate the pin	No information available
9	OP3: Advance the pin	No information available
10	OP5: Take an inlet view	The pin comes a little far from the anterior cortex of the lateral part of the sacrum on the inlet view (the position of the pin is still incorrect)
11	OP6: Take an outlet view	Take an outlet view: the position of the pin is correct on the outlet view
12	OP9: Validate the pin course	The same traces from the previous actions

The didactic variables were used to create a variety of problems (Table 1). This set of problems could be used to create cognitive conflict by the learner (i.e., to destabilize the learner's conceptions) to make learning happened effectively. For example, a problem with high-density bone may be presented to the learner to help him or her better understand exceptional cases (see the problem shown in the dialog extract above) if the system detects that he or she does not have any understanding about the importance of the quality of the patient's bone. Sometimes, the problems could be used to refine the diagnosis result about the learner's conceptions. For example, the learner could be asked to validate a particular solution so that the diagnosis component can get more evidence about how he or she under-stands the importance of the quality of the patient's bone. That refinement may help the system generate more relevant feedback for the student.

Operators (R)

The didactic analysis allowed a detailed description of the process of sacro-iliac screw fixation described previously as well as a set of operators. Table 2 shows a number of operators used in a problem-solving sacro-iliac screw scenario. We use this scenario to explain how the diagnosis component works in the next section.

Controls (Σ)

As mentioned previously, during the problem-solving process the surgeon's checking of whether an action is relevant or whether a problem or sub-problem is solved or not is associated with a set of controls. In the domain of sacro-iliac screw fixation, the didactic analyst distinguished two kinds of controls, according to their epistemological dimension: (1) theoretical (mastered in the theoretical course), and (2) pragmatic (mastered by experience). All of them are declarative knowledge (procedural knowledge, i.e., the steps to make a pin course described previously, is quite easy for any surgeon student to master, so it is not mentioned here). The didactic analyst also classified controls into four groups, according to the nature of the subject domain: (1) anatomy controls that describe the anatomical characteristics of the pelvis, (2) trajectory controls that describe the characteristics of the pin course, (3) radiography controls that describe the characteristics of the X-rays taken during the surgical operation, and (4) correspondence controls that describe relationships between the X-rays and the body. Those classifications are necessary because the modeling of the diagnosis process and the modeling of the didactic decision depend on the nature of controls (see more details in the following section). Table 3 shows a subset of about 100 controls the didactic analyst collected.

The controls, which are associated with the surgeon's decision-making, are the key elements of the difference between the expert surgeon teacher and the novice surgeon learner. Indeed, the surgeon teacher knows how and when to "use" these controls to take decisions, whereas the learner may "use" some of these controls out of their validity domain. When we say that surgeons "use" the controls we mean their understanding is associated with the controls we have identified (the formulation of their controls may be different from that of our controls, though the meaning of their controls and that of our controls are similar). For instance, after perceiving that the pin comes too near from one precise anatomic part of the pelvis bone (e.g., its anterior cortex) on the inlet view, meaning that the pin is "too low" on the X-ray representation, the expert uses Σ14 (see Table 3) to decide to correct the entry point downwards, in relation to the body lying position (see Fig. 4). The novice may not have that control in mind (and thus does not correct the pin course) or may have a correct understanding of that control (and acts as does the expert) or may have an incorrect understanding of that control in the mentioned context (and therefore corrects the entry point upwards, which is invalid).

To decide which control(s) to be used in a given problem-solving situation, the surgeon (learner or teacher) needs to examine the situation, principally by taking X-rays, to determine its characteristics regarding the domain constraints. We use the term "situation variables" (SVs) to describe the characteristics of a given problem-solving situation. Table 4 shows a couple of examples of SVs. The value of a SV can be evaluated by the surgeon only when he or she does a relevant operation, for example, "take an inlet view" to identify the value of SV1 and SV10.

Table 3
Examples of controls in sacro-iliac screw fixation

Control ID	Type	Content
Anatomy		
Σ8bis	Theoretical	IF the pin is down the anterior cortical bone of the iliac wing on the inlet view, THEN it can hurt the lumbo-sacral trunk
Σ9bis	Theoretical	IF the pin is up the sacral canal on the inlet view, THEN it can hurt the S1 roots and the cauda equine
Σ38	Theoretical	IF the pin is in front of the dense lines on the lateral view, THEN it can hurt the sacral canal
Trajectory		
Σ7	Theoretical	IF the pin is well positioned, THEN it is behind the dense lines on the lateral view
Σ8	Theoretical	IF the pin is well positioned, THEN it is up the anterior cortical bone of the iliac wing on the inlet view
Σ30	Pragmatic	IF the pin would become extra osseous by being pushed in S1, 1cm after the median line, THEN it can be stopped at the median line
Σ65	Pragmatic	IF the screw is well anchored, THEN its extremity lies in S1, 1cm after the median line
Σ67	Pragmatic	In case of a disjunction: IF the pin would become extra osseous, THEN it can be stopped just 1cm after having reached S1
Radiography		
Σ6	Theoretical	IF the X-ray is a good outlet THEN the sacral plate must be visible
Σ92	Pragmatic	IF the pin is correctly positioned on the inlet view, THEN the pin trajectory can be valid, but not sure (i.e., one should not validate the pin course in this case, but should look for more evidence to do so)
Correspondence		
Σ14	Pragmatic	IF the pin touches the anterior cortex of the lateral part of the sacrum on the inlet view THEN it is too ventral on the body of the patient
Σ15	Pragmatic	IF the pin touches the posterior cortex of the lateral part of the sacrum on the inlet view THEN it is too dorsal on the body of the patient
Σ20	Pragmatic	IF the pin touches the first anterior sacral foramen on the outlet view THEN it is too caudal on the body of the patient

Table 4
Examples of situation variables

Situation Variable ID	Description
SV1	The pin touches the anterior cortex of the lateral part of the sacrum on the inlet view.
SV10	The pin touches the posterior cortex of the lateral part of the sacrum on the inlet view.

Representation System (L)

Although we used a simple but systematic representation system to represent, for example, operators in the simulation component and to name various variables in the diagnosis component (see the next section), the representation system is not a main concern in our current research. In the future, however, it could be useful (e.g., for didactic engineering) to add to the knowledge model a representation as an attribute to each control used in the model: gestural, mental 3D representation, 2D imagery, kinesthetic sensation, sound, etc.

Conclusion

In the model presented previously, the solving process of a problem P can be described as a succession of operators (i.e., actions in our case) whose consequences on the problem are validated by controls, which are used in a certain system of representation. The model may be used to describe the solving process of more complex problem-solving situations we envision for future research. For example, the progression of the pin may be sometimes controlled by the surgeon by hearing the sound of the surgical motor he or she uses to insert the pin: a shriller sound is interpreted as a progression in a dense part of the bone. Then, the surgeon's knowledge of the pelvis anatomy allows him or her to know the exact location of the pin extremity. Most of the time, he or she will confirm this information by taking an X-ray. This action provides him or her with a validation of this step of the problem-solving process in another system of representation (2D image).

STUDENT MODELING AND DIAGNOSIS

The control structure previously presented is the most important element for validation, a key aspect of problem solving (Schoenfeld, 1985). Therefore, the objective of our diagnosis component is to diagnose the student's understanding about the controls after each of the actions he or she performs during his or her interaction with the simulation component of our learning system. The diagnosis result will be used to make didactic decision about relevant feedback to be provided to the student (see the previous section that describes the architecture of our learning environment).

Diagnosing the learner's understanding about a certain control is difficult. For example, in the scenario shown in Table 2, after the learner does Action 4, it is difficult to diagnose his or her understanding about the related controls (e.g., see Σ14 in Table 3): the learner may not have these controls in mind and makes such a pin course randomly, or he or she may have correct understanding of these controls but makes an incorrect pin course for another reason (even an expert may need several attempts to arrive at a correct pin course), or he or she may have incorrect understanding of these controls and therefore makes such an incorrect pin course. In other words, it is uncertain to diagnose the learner's knowing state in that kind of situations. Bayesian networks provide an effective mean for modeling under uncertainty (Henze & Nejdl, 2001; Mayo & Mitrovic, 2001).

In addition, considering the temporal dimension in student modeling and diagnosis is important because it may help model the student's knowing state and cognitive behavior more completely. For instance, in the previous scenario, the student makes an incorrect solution in the first course of actions (Action1 - Action 5). Then, he or she makes a good correction behavior in the second course of actions (Action 6 – Action 11). The student, however, validates an incorrect pin course (Action 12). If not considering the temporal dimension, the system may focus only on the error the student makes in Action 12, diagnose that he or she has an incorrect understanding about Σ14 with high probability, and generate feedback regarding that error. But if taking into account the temporal dimension, the system should concentrate on both the student's correction behavior in the second course of actions and his or her error in Action 12, diagnosing that he or she has an incorrect understanding about Σ14 with average probability, and produce feedback with respect to both that cognitive behavior and that error. We believe that the feedback in the latter case could be better because it may help the learner focus on his or her good correction behavior and hence arrive at a correct solution by an optimal path.

Temporal Bayesian networks (Russell & Norvig, 2009; Ghahramani, 1998) could be exploited to model such a temporal dimension. In the following sub-sections, we first introduce necessary background on temporal Bayesian networks, then we present the student model in our learning system, and finally we show a *meta-model* as well as details of the temporal Bayesian network used in the student diagnosis component of our learning system.

Temporal Bayesian Networks

A Bayesian network is a graphical model that researchers use to encode probabilistic relationships among variables of interest. More specifically, a Bayesian network is a directed, acyclic graph whose nodes represent random variables and whose arcs encode conditional dependencies. If there is an arc from node X to another node Y, X is called a parent of Y and Y is called a child of X. A node without parents is called a root node or an unconditional node. A node with parents is called a conditional node, which is attached with a conditional probability table (CPT) that quantifies the effect of its parents on the conditional node. The probabilistic computation in Bayesian networks is grounded in the rules of probability (Pearl, 2000, Chapter 1), which in turn are based on Bayes's theorem or law. This law relates the conditional and marginal probability of events X and Y, where Y has a non-vanishing probability, as follows:

$$P(X \mid Y) = \frac{P(Y \mid X)P(X)}{P(Y)}$$

Where:

- P(X) is the marginal probability of X. It is also called the prior probability in the sense that it does not consider any information about Y.
- P(X|Y) is the conditional probability of X, given Y. It is also called the posterior probability because it is derived from the specified value of Y.
- P(Y|X) is the conditional probability of Y, given X.
- P(Y) is the marginal probability of Y, and it acts as a normalizing constant.

Intuitively, Bayes' law in the previous form describes the way in which one's beliefs about observing X are updated by having observed Y. That is why researchers in the field also name Bayesian networks as belief networks (Reye, 2004).

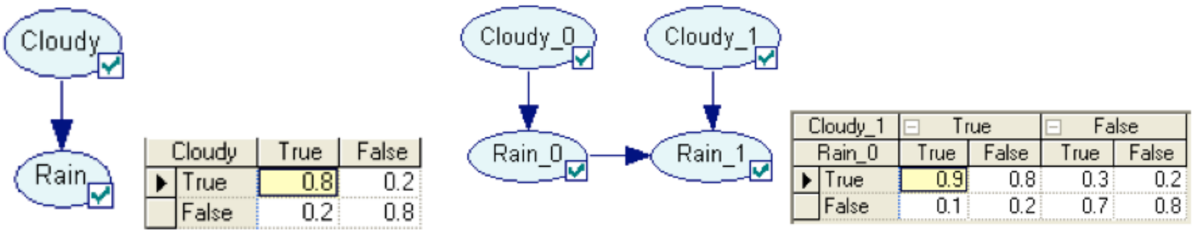

Fig. 5. A simple Bayesian network (left) and a simple temporal Bayesian network (right).

Fig. 5 shows our example of a simple Bayesian network: "Cloudy" is an independent random variable (a root or unconditional node), which has two possible values (True and False). "Rain" is a dependent random variable (a conditional node), which also has two possible values (True and False). “Cloudy” node is called a parent of “Rain” node and “Rain” node is called a child of “Cloudy” node. “Rain” node is labeled with a CPT (see Fig. 5), which means that if it is cloudy then it probably (80%) rains and that if it is not cloudy then it probably (80%) does not rain. On the basis of this Bayesian network, one can calculate the probability of the fact "it rains" from the probability of the fact "it is cloudy", for instance, if the probability of "it is cloudy tomorrow" is 70% then the probability of "it rains tomorrow" is 62%.

A temporal Bayesian network is a Bayesian network in which stochastic processes are modeled, that is, temporal dimension is taken into account (Russell & Norvig, 2009; Ghahramani, 1998). In the previous example, if one wants to consider the fact "if it rains today, it will probably rain tomorrow", he or she may build the following model (Fig. 5): "Cloudy_0" and "Rain_0" represent variables in the previous day and "Cloudy_1" and "Rain_1" represent variables in the following day. The CPT attached to variable "Rain_1" is redefined by considering its two parents: "Cloudy_1" and "Rain_0". If one knows, for instance, that the probability of the fact "it is cloudy tomorrow" is 70% and the probability of the fact "it is cloudy the day after tomorrow" is 50%, the present model allows him or her to calculate the probability of the fact "it rains the day after tomorrow" to be 56%.

Student Model

The diagnosis component in our learning system aims at diagnosing the student's use of controls during his or her problem-solving process. As mentioned in an example presented previously about control Σ14, for each control we consider three states about the student's use of that control:

- `BPV`: This state stands for "brought into play validly". It means that the student uses the control and his or her understanding about the control is correct, so he or she may know when and how to use it correctly, as expected by the surgeon teacher.
- `BPI`: This state stands for "brought into play invalidly". It means that the student has a misunderstanding about the control in a specific situation or context. More specifically, he or she may have correct understanding about the control in another context, but he or she wrongly applies the control in the given situation.
- `NBP`: This state stands for "not brought into play". It means that the student does not use the control, probably because he or she does not have it in his or her mind.

To model the student's knowing state, we use a *control vector* (*ΣV*), based on a model proposed by Henze and Nejdl (2001). Each component of the vector is a conditional probability, describing the diagnosis system's estimation that a student *S* has understanding about a control *Σ*, on the basis of all observations (evidences) ***E*** the system has about *S*:

$$\Sigma V(S) = (P(\Sigma_1|E), P(\Sigma_2|E), \ldots, P(\Sigma_n|E))$$

Table 5 shows a part of the student model, initialized for every learner at the beginning of the problem-solving process. Because there is no evidence about the student's use of controls at the beginning, for each control the probability is equally distributed for the three states BPV, BPI, and NBP (the value for NBP is set to be 0.34 in order to satisfy the rule that the sum of the three values must be equal to 1).

Table 5
A part of the initialized student model

Control ID	**Probability**		
	BPV	BPI	NBP
Σ14	0.33	0.33	0.34
Σ15	0.33	0.33	0.34

Student Diagnosis

The main task for building the diagnosis component in our learning system is to create a temporal Bayesian network. In this section, we show both an approach we applied to build the temporal Bayesian network and a meta-model of this network. Showing those elements in detail could help the reader understand why and how they may be reused for student diagnosis in other subjects than sacroiliac screw fixation and in other instructional contexts than simulation-based learning systems.

Meta-model

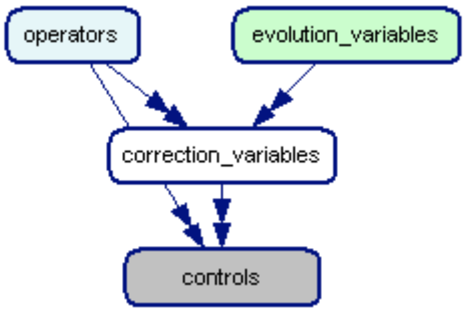

Fig. 6. A meta-model of the temporal Bayesian network for student diagnosis.

Fig. 6 shows the meta-model of our temporal Bayesian network in which we emphasize the temporal dimension. This meta-model has the following sub-models:

- The `operators` sub-model contains nodes representing the operators identified in the previous didactic analysis (see Table 2).
- The `evolution_variables` sub-model contains nodes representing the "evolution" variables. Briefly, this sub-model is mainly used to monitor the evolution of the characteristics of the student's pin course throughout his or her construction of a solution.

- The `correction_variables` sub-model contains nodes representing the "correction" variables. This sub-model is principally used to monitor the student's correction process (i.e., correction behavior) throughout his or her construction of a solution (e.g., see Action 6 – Action 11 in Table 2).
- The `controls` sub-model contains nodes representing the controls identified in the previous didactic analysis, for example, Σ14 (see also Table 3).

The student diagnosis in our learning system is "control-oriented" (i.e., it diagnoses the student's knowledge about controls), thus the approach used to build the temporal Bayesian network is also "control-oriented." In other words, for each of the controls (e.g., Σ14), we applied the same process to build the diagnosis model for that control. Fig. 7 illustrates the diagnosis model for Σ14: `take_inlet` and `validate_pin_course` are operator nodes in the `operators` sub-model, `evolution_distance_0`, `evolution_distance_1` are evolution nodes in the `evolution_variables` sub-model, `correction_distance` is an correction node in the `correction_variables` sub-model (for this specific example, "distance" is used to indicate the distance between the pin and the anterior cortex of the lateral part of the sacrum on the inlet view and to replace "distance_pin_anterior_cortext_on_inlet" in long variable names in Fig. 7), and `sigma14_0` and `sigma14_1` are control nodes in the `controls` sub-model. By convention, the variables ending with "_1" are used to represent the current state of the student's pin course, whereas the variables ending with "_0" are used to represent the previous state of the student's pin course (we give more details about this in the next sub-sections). To compute the value of the evolution variables, a set of IF-THEN rules, which represents domain constraints identified in the phase of didactic analysis, is used. We present the four sub-models and those rules in detail in the following sub-sections.

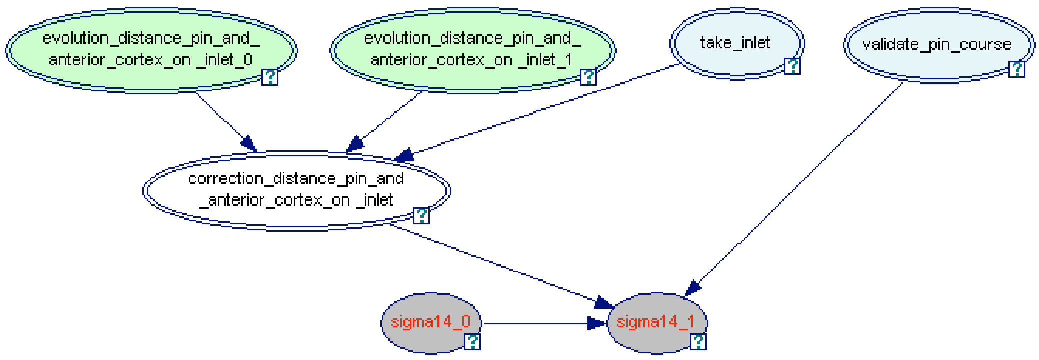

Fig. 7. A part of the temporal Bayesian network for modeling the diagnosis of Σ14 (sigma14).

Evolution Variables

In the phase of didactic analysis, the didactic analyst identified relationships among controls, operators, and situations variables (see also Table 4), for instance, the situation variable related to Σ14 is SV1: "the pin touches the anterior cortex of the lateral part of the sacrum on the inlet view." We use, however, a variable such as "distance between the pin and the anterior cortex of the lateral part of the sacrum on the inlet view" to model the evolution and the correction of the student's pin course.

The scenario presented in Table 2 suggests that it is important to take into account the values of the same situation variable at different points in time in order to model the temporal dimension. Thus, for each of the situation variables related to a control, we define two intermediate variables to model the evolution of the situation variable, for example, Fig. 7 shows two intermediate variables created for Σ14: `evolution_distance_0` and `evolution_distance_1` (we explain more about why those two variables are useful in the sub-section about correction variables). In the temporal Bayesian network, those variables are represented by deterministic nodes (a particular kind of nodes in Bayesian networks, see GENIE, 2006), which take one of the following five values, for example, for variable `evolution_distance_1`:

- `correct`: When the distance between the learner's pin *at present* and the anterior cortex of the lateral part of the sacrum on the inlet view is *correct*, according to the domain constraints (but not to an ideal solution by the expert surgeon).
- `incorrect`: When the distance between the learner's pin *at present* and the anterior cortex on the inlet view is *incorrect* and the system has no information about that distance *at a previous time* (e.g., see Action 4 of the scenario shown in Table 2). By previous time, we mean the most recent moment the system updates the student model (updating is performed after every operation by the student with the simulation component).
- `incorrect_good_way`: When the system has information about that distance both at a previous time and at present, and the distance at present is incorrect, but there is evidence that the learner's behavior toward the error is in a good way (e.g., see Action 10 in Table 2).
- `incorrect_bad_way`: Similar to `incorrect_good_way`, but the learner's behavior toward the error is in a bad way.
- `no_information`: This is a value by default. This value is useful, for instance, for initializing the Bayesian network at the beginning of the problem-solving process.

IF-THEN Rules

To compute the value of the previous evolution variables ending with "_1" at present, the diagnosis component uses a set of IF-THEN rules, which was formulated on the basis of the domain constraints collected in the phase of didactic analysis. Table 6 shows several rules related to variable `evolution_distance_1`. The set of IF-THEN rules was created by considering the temporal dimension explained in the previous sub-section (see the situation variables ending with "_0" in the IF clause in Table 6).

Table 6
A part of the IF-THEN diagnosis rules
(distance is in millimeter, suffix "_1" means "at present" and suffix "_0" means "at a previous time")

IF Clause	THEN Clause
(distance_pin_and_anterior_cortex_on_inlet_1 > 4)	evolution_distance_1 = correct
(distance_pin_and_anterior_cortex_on_inlet_1 <= 4)	evolution_distance_1 = incorrect
((distance_pin_and_anterior_cortex_on_inlet_1 <= 4) AND (distance_pin_and_anterior_cortex_on_inlet_0 < distance_pin_and_anterior_cortex_on_inlet_1))	evolution_distance_1 = incorrect_good_way
((distance_pin_and_anterior_cortex_on_inlet_1 <= 4) AND (distance_pin_and_anterior_cortex_on_inlet_0 > distance_pin_and_anterior_cortex_on_inlet_1))	evolution_distance_1= incorrect_bad_way

Correction Variables

Although the temporal dimension has been partly taken into account in the set of IF-THEN rules, the evolution variables could not allow the system to model the learner's correction behavior *completely*, for example, to model the following situation: At a previous time, the learner made a pin course in which the distance between the pin and the anterior cortex is 3mm (i.e., incorrect). At present, to correct a certain error, which may not be the error related to that distance, the learner makes another pin course in which the distance between the pin and the anterior cortex is unchanged (i.e., 3mm). In this situation, the system should diagnose that the learner has not made any correction regarding that distance yet. Evolution variables ending with "_0" (e.g., `evolution_distance_0` for Σ14) are thus created to detect that kind of specific situations. The value of `evolution_distance_0` is copied from that of `evolution_distance_1` computed at the previous time.

Therefore, to model the student's correction behavior toward a certain error as completely as possible, deterministic nodes (e.g., `correction_distance`, see Fig. 7) are created. Furthermore, the learner, while interacting with the simulation component, needs to do certain operations in order to examine the evolution of the pin course. For instance, to see the change of the distance between the pin and the anterior cortex on the inlet at the previous time and at present, the learner needs to take an inlet view (see Actions 4 and 10 in Table 2). Thus, node `correction_distance` has three parents: two evolution nodes described earlier and one deterministic binary node representing whether the respective operation is done or not by the student (see `take_inlet` in Fig. 7). This correction node can take one of the following five values:

- `correct`: Similar to the value `correct` of the "evolution" variables.
- `no_correction`: For example, the distance between the pin and the anterior cortex on the inlet view is unchanged (see the example described in the previous paragraph).
- `good_way`: Similar to the value `incorrect_good_way` of the "evolution" variables.
- `bad_way`: Similar to the value `incorrect_bad_way` of the "evolution" variables.
- `no_information`: Similar to the value `no_information` of the "evolution" variables.

Table 7 shows a part of the definition of this node. The point we make here is that *this definition is the same for every correction node*.

Control Nodes

After creating the "correction" variable related to Σ14, two chance nodes (e.g., `sigma14_0` and `sigma14_1`, see Fig. 7) are created. The former represents the cumulative diagnosis result of the respective control until the previous time and the latter represents that at present. As described in the previous sub-section about student model, control nodes have three outcomes: BPV (brought into play validly), BPI (brought into play invalidly), and NBP (not brought into play). Node `sigma14_0` has no parents, and at the beginning of the problem-solving process, the outcomes of this node are equally distributed (see Table 5). Node `sigma14_1` has three parents: `sigma14_0`, `correction_distance`, and a deterministic binary node `validate_pin_course` that represents whether the learner's current operation is the validation of the pin course or not. We take into account operator `validate_pin_course` because we consider that validating an incorrect solution is different from making an incorrect pin course and doing an operation other than validating the pin course, for instance, restarting to correct errors (in the latter case the learner's knowing state and cognitive behavior should be diagnosed with more positive result than those in the former case, see more details in the following sub-section about evaluation). Table 8 presents a part of the CPT of node `sigma14_1` we subjectively filled out. *This same CPT is used for every control node in the same group* (see four groups in Table 3: anatomy, trajectory, radiography, and correspondence).

Table 7
A part of the definition of correction nodes

take_inlet	yes				
evolution_distance_ 0	incorrect				
evolution_distance_ 1	correct	incorrect	incorrect_good_way	incorrect_bad_way	no_information
good_way			X		
bad_way				X	
correct	X				
no_correction		X			
no_information					X
take_inlet	yes				
evolution_distance_ 0	correct				
evolution_distance_ 1	correct	incorrect	incorrect_good_way	incorrect_bad_way	no_information
good_way					
bad_way		X	X	X	
correct	X				
no_correction					
no_information					X

Table 8
A part of the definition of "control" nodes in the correspondence group

correction_distance	good_way					
validate_pin_course	yes			no		
sigma14_0	BPV	BPI	NBP	BPV	BPI	NBP
BPV	0.4	0.1	0.1	0.6	0.4	0.4
BPI	0.5	0.8	0.5	0.2	0.4	0.2
NBP	0.1	0.1	0.4	0.2	0.2	0.4

Diagnosis Process

At the beginning of the learner's problem-solving process, every node in the Bayesian network is set to a value by default (e.g., `no_information` for “evolution” nodes). The functionality of the diagnosis agent can be summarized in the following four steps:

1. After receiving the student's action and traces provided by the tracing agent, the diagnosis agent uses the set of IF-THEN rules to calculate the values of the "evolution" variables. The diagnosis agent always keeps two values for each trace: one for the current time and the other for the previous time.

2. For the Bayesian network, the diagnosis agent sets "True" for the value of the operator corresponding to the student's action and uses the results computed in Step 1 to set the values for the "evolution" nodes.
3. The diagnosis agent updates the Bayesian network and sends the diagnosis result (e.g., the probabilities about the three states of the controls) to the didactic decision agent, which in turn will generate feedback for the student.
4. To make the Bayesian network function correctly for the student's next action (i.e., to consider the temporal dimension), after updating the Bayesian network the diagnosis agent copies the values of the nodes ending with "_1" (e.g., `evolution_distance_1`, `sigma14_1`) to the values of the nodes ending with "_0" (e.g., `evolution_distance_0`, `sigma14_0`).

Table 9
A part of the diagnosis result of the scenario shown in Table 2
(BPV = brought into play validly, BPI = brought into play invalidly, NBP = not brought into play)

Operator	**Pin traces**	**"Evolution" variables**	**Part of diagnosis result**			
			Σ	**BPV**	**BPI**	**NBP**
OP1: Introduce an entry point	None	all variables: no_information	Σ14	0.33	0.33	0.34
OP2: Orientate the pin	None	all variables: no_information	Σ14	0.33	0.33	0.34
OP3: Advance the pin	None	all variables: no_information	Σ14	0.33	0.33	0.34
OP5: Take an inlet view	distance_pin_and_anterior_cortex_on_inlet=1	evolution_distance_pin_and_antrior_cortex_on_inlet_1=*incorrect*	Σ14	0.20	0.20	0.60
OP6: Take an outlet view	distance_pin_and_sacral_foramen_on_outlet=6	evolution_distance_pin_and_sacral_foramen_on_outlet_1=*correct*	Σ14	0.20	0.20	0.60
OP4: Restore the pin	None	all variables: no_information	Σ14	0.20	0.20	0.60
OP1: Introduce an entry point	None	all variables: no_information	Σ14	0.20	0.20	0.60
OP2: Orientate the pin	None	all variables: no_information	Σ14	0.20	0.20	0.60
OP3: Advance the pin	None	all variables: no_information	Σ14	0.20	0.20	0.60
OP5: Take an inlet view	distance_pin_and_anterior_cortex_on_inlet=3	evolution_distance_pin_and_anterior_cortex_on_inlet_1=*incorrect_good_way*	Σ14	0.44	0.24	0.32
OP6: Take an outlet view	distance_pin_and_sacral_foramen_on_outlet=5	evolution_distance_pin_and_sacral_foramen_on_outlet_1=*correct*	Σ14	0.44	0.24	0.32
OP9: Validate the pin course	distance_pin_and_anterior_cortex_on_inlet=3 distance_pin_and_sacral_foramen_on_outlet=5	evolution_distance_pin_and_anterior_cortex_on_inlet_1=*incorrect* evolution_distance_pin_and_sacral_foramen_on_outlet_1=*correct*	Σ14	0.23	0.57	0.20

Evaluation

The model described previously is not based on the construction of expert solutions, but on a set of domain constraints representing the expertise of the domain. The approach we used is similar to an expert system approach (Russel & Norvig, 2003; Mayo & Mitrovic, 2001), especially in the engineering of IF-THEN rules, but we examined and modeled the temporal dimension explicitly and systematically. For the time being the validation of our approach is based on methods used in expert centric approaches and decision-theoretic expert systems. We validated several computational aspects related to the possibility to produce a cognitive diagnosis: we defined a gold standard validation (Russell & Norvig, 2009) in which we identified a set of scenarios with a set of correct input and output pairs.

The input is the actions and the traces of the pin course while the learner is interacting with the simulation component, and the output is the diagnosis produced by the diagnosis component (e.g., see Table 9). Our research team, including computer scientists and didactic experts, worked collaboratively to specify and validate the testing scenarios. The validation we performed with 6 scenarios and about 20 controls suggests that our diagnosis agent is able to produce coherent diagnosis with an acceptable response time (say, less than five seconds when running on a Dell PC with a 2.4Ghz processor and a 512Mb RAM). For example, the diagnosis result shown in Table 9 may be interpreted, as follows: After Action 4, because the learner made an incorrect distance between the pin and the anterior cortex and he or she took an inlet view, the outcome NBP (not brought into play) of the related control (i.e., Σ14) is increased. After Action 10, because it seems that the learner corrected the pin course in a good way, the outcome BPV (brought into play validly) of Σ14 is increased. After Action 12, however, because the learner validated an incorrect solution, the outcome BPI (brought into play invalidly) of Σ14 is increased.

Discussion

In the Bayesian network, we modeled the learner's behavior only at two points in time: the current time and the most previous time (each point in time corresponds to an action of the learner while interacting with the simulation component). The learner's behavior at other points in time (i.e., before the most previous time) is taken into account when the Bayesian network is updated at the most previous time (see Step 4 in the previous sub-section about the diagnosis process). More specifically, the control nodes ending with "_0" cumulatively represent the diagnosis result of the learner's knowings from the beginning of the problem-solving session to the most previous time. A technical metaphor for that process could be a recursive formulas to compute the sum of 1 to N: `F(1) = 1, F(N) = F(N-1) + N` where N > 1, in which computing F(N) is based on the value of F(N-1) and the value of N; so, the metaphor of F(N) is the control nodes ending with "_1", the metaphor of F(N-1) is the control nodes ending with "_0", and the metaphor of N is the action and the traces of the pin course at present. In other words, modeling the temporal dimension of the sequence of actions and traces in the Bayesian network could also be understood as a means to incrementally construct the student model.

In addition to the diagnosis result of the current state of the learner's knowings represented in the probabilities of control nodes, the diagnosis component also explicitly provides *the diagnosis result of the learner's cognitive behavior over time*, which is represented, for example, by correction nodes. The former diagnosis result has been proved to be critical in many traditional ITSs (Wenger, 1987).

The latter diagnosis result about cognitive behavior, we believe, could be also important. On the one hand, it may help the didactic decision agent generate better feedback for the learner. For instance, in the previous example the diagnosis agent detects that the learner validated an incorrect pin course, and it also recognizes that he or she corrected the pin course in a good way before the validation, it may therefore ask the didactical decision agent to generate feedback by focusing on the learner's correction behavior before on his or her validation mistake—these choices may be driven by parameterizable pedagogical hypotheses. On the other hand, it may help the system recognize when and how learning happens within a process of solving a problem and across multiple processes of solving different problems, so as to improve the student model and to generate better feedback for the student. For instance, let us return to the previous example, after several error-feedback interactions between the learner and the system, he or she is asked to solve another problem, he or she makes an incorrect pin course, then corrects the pin course in a good way several times until the pin course is correct, and finally validates the pin course. In this case, the system may recognize that learning (how to do to correct a wrong pin course) happens, probably with the assistance of the previous feedback generated by the system for the student. So, the system would continue to use the same effective strategy to generate that kind of feedback in similar cases.

The IF-THEN rules and the temporal Bayesian network presented earlier may not be the only way to take into account the temporal dimension in student modeling and diagnosis. One may use, for instance, more IF-THEN rules and less deterministic nodes in the Bayesian network than we did to arrive at the same solution. The point is that the temporal dimension should be systematically modeled in both the IF-THEN rules and the Bayesian network.

There may be certain dependencies among control nodes in the proposed Bayesian network, for example, if one knows the student's cognitive state about a certain control, he or she may be able to deduce the student's cognitive state about the controls that are "cognitively related" to that control. We believe that the consideration of this kind of dependencies in the engineering of the proposed Bayesian network could improve its effectiveness in terms of diagnosis. We shall take into account that kind of dependencies in future research.

RELATED WORK AND DISCUSSION

In this section, we discuss several cognitive approaches for student modeling and diagnosis that are closely related to ours: a conception-based approach, a constraint-based approach, and a Bayesian-network-based approach. The main contribution of this paper to those existing approaches is a new temporal-Bayesian-network-based model for student modeling and diagnosis, in which in addition to diagnosing the learner's current knowing state we emphasize the importance of explicitly and systematically diagnosing the learner's cognitive behavior expressed by a sequence of actions and productions he or she performed within a problem and across problems.

The interested reader is referred to the work of Wenger (1987) and of Webber (2004) for reviews of other cognitive approaches such as the overlay approach (Clancey, 1983; Burton & Brown, 1982), the buggy approach (Conati, Gertner, & VanLehn, 2002; Brown & Burton, 1978), and the model tracing approach (Koedinger, Anderson, Hadley, & Mark, 1997; Anderson, Boyle, & Yost, 1986; Anderson, Farrell, & Sauers, 1984).

In this paper, the simulation for sacro-iliac screw fixation is used only as an example to illustrate and evaluate our proposed model. That is, this paper goes beyond the example of medical education and computer-based training simulations. The reader who is interested in those fields is referred to a brief review of technology in medical education (Lajoie, Faremo, & Wiseman, 2001) and a review of simulation-based learning systems, including simulation-based ITSs (Joab, Guéraud, & Auzende, 2005).

Conception-based Approach

The conception-based approach (Webber, 2004) is also grounded in the theoretical framework presented previously, especially the cK¢ model. This approach has been used for student modeling and diagnosis in the Baghera project (Soury-Lavergne, 2003), which provides a platform to help students learn doing proofs in geometry. The approach is an emergent, multi-agent, and bottom-up approach, in which there are two main levels: the micro level and the macro level. The former consists of a complex network of conception elements (i.e., problems, operators, representation system, and controls), based on the ck¢ model. The latter represents the set of conceptions the student may hold, for example, central symmetry, orthogonal symmetry, and parallelism symmetry in the domain of reflection. Thus, the macro level corresponds to an abstraction (in terms of knowledge) of what is represented in the micro level. More specifically, the major role of the macro level is to observe and interpret the final state of the agents in the micro level in terms of diagnosis result.

During the student's problem-solving process, the agents in the micro level will use group-decision-making strategies (a spatial voting mechanism) and coalition formation (Sandholm, 1999) to update the representation of the diagnosis result at the current time, and one or more agents in the macro level will interpret that representation in terms of conceptions held by the student at the current time. Those conceptions are associated with utility values (e.g., 20 for central symmetry, 11 for orthogonal symmetry, and 10 for parallelism symmetry). A Tutor agent in the macro level will choose the conception(s) with the largest utility (i.e., central symmetry in the example) and provide the student with relevant feedback regarding those conceptions. For example: (1) presenting the student with a new activity to reinforce his or her correct conceptions, (2) confronting the student with more complex situations, (3) presenting examples or counterexamples to the student, (4) promoting interactions between the student and peers or the teacher.

The main difference between the conception-based approach and our approach, as mentioned previously, is the modeling of the temporal dimension. Indeed, the conception-based approach analyzes the student's actions and productions and updates the student model only at the moment when the student validates a solution, and it does not analyze each step of the problem-solving process and model the student's cognitive behavior, as does our approach. The conception-based system is able to produce a diagnosis only at the final stage of the problem-solving process, whereas our system evolves throughout the student's interaction with the simulation component, thanks to the temporal Bayesian network, and in consequence our system is able to provide feedback to the student at any step of the problem-solving session. Thus, we believe that our approach could get closer to the notion of "milieu" proposed by Brousseau (1997) than the conception-based approach.

There is another noteworthy difference: our diagnosis component focuses on controls, but not on conceptions. In principle, using a control-based diagnosis result could generate more fine-grained feedback than using a conception-based diagnosis result. For example, by targeting "fine-grained" controls the system is able to suggest the student to explore a precise part of a web content page related to the targeted controls, whereas by targeting conceptions, which are usually general (e.g., central symmetry), the system may be able to suggest only a chapter or a section for the student. Obviously, because controls are a part of the conception network, it may not be difficult to switch from conception focus to control focus in the conception-based approach.

Constraint-based Approach

Student modeling and diagnosis of a constraint-based ITS (Ohlsson, 1992, 1994) are principally based on a database of domain constraints, which models correct evaluative knowledge of the subject being taught—evaluative knowledge is used to evaluate outcomes of one's action as desirable or undesirable. In the domain of Lisp programming, for example, here is a simple constraint (Mitrovic & Ohlsson, 1999, p. 239): "*If the code for a Lisp function has N left parentheses, there has to be N right parentheses as well (or else there is an error).*" This constraint can be used to evaluate the learner's action of constructing a Lisp expression.

During the problem-solving session, the constraint-based tutor analyzes the learner's solution with respect to the set of constraints, and sometimes to the ideal solutions of the given problems, to detect the constraint(s) the learner may have violated. The student modeler of the tutor records the history of each constraint throughout the learner's problem-solving process, for example, how often the learner satisfied / violated the constraint. After the learner submits a solution, the tutor always concentrates on one violated constraint, if any, to generate feedback for the learner—if multiple violated constraints are identified, the tutor will select the constraint with the largest number of violations. Feedback may be organized into five levels of detail: right/wrong, error flag, hint, partial solution, and complete solution. The right/wrong feedback is often used after the first attempt of the learner, whereas the flag feedback and the hint feedback are often used after several unsuccessful attempts. The learner can view the partial solution or the complete solution if he or she wants.

Although the theoretical framework of the constraint-based approach and that of ours are different, both share several common important points. Firstly, the constraint-based approach concentrates on modeling evaluative knowledge, which is somewhat similar to control knowledge modeled in our approach, except for our explicit modeling of tacit pragmatic knowledge. Secondly, both approaches are not mainly based on the construction of expert solutions, but on the set of domain constraints. In other words, in principle the diagnosis component in a tutor applying either approach cannot be executed to solve problems. Both approaches, however, are able to detect multiple correct solutions by the learner, some of which may be different from expert solutions. This feature is critical because "[the tutor needs] not be thrown off track by correct but creative or unusual solutions" (Mitrovic & Ohlsson, 1999, p. 253). Thirdly, targeting constraints or controls as the object of the diagnosis could be promising because the tutor is able to provide the learner with fine-grained feedback directly concerned with a constraint or control or a group of constraints or controls, for example, a constraint-based SQL-Tutor contains about 400 constraints, each of which may be associated with different feedback. Additionally, it would be not too difficult for the ITS designer to build and maintain the diagnosis component using either approach, for example, to add or remove constraints in the constraint-based model or controls in our model.

Beside the main difference between the two approaches about the explicit modeling of the temporal dimension, as the difference between the conception-based approach and ours, there is another technical difference: we use Bayesian networks as a complementary tool to the set of IF-THEN rules to diagnose the cognitive state of the learner about the use of controls.

Bayesian-network-based Approach

The work of Mayo and Mitrovic (2001) provides a detailed review on the use of Bayesian networks for student modeling and diagnosis. In this sub-section, we give only an overview of this approach. Sometimes, the distinction between the Bayesian-network-based approach and other approaches is not clear-cut, because there are ITSs, for example the CAPIT tutor (Mayo & Mitrovic, 2001), that use Bayesian networks as a complementary tool to improve an existing student model.

According to Mayo and Mitrovic (2001), it is possible to classify Bayesian-network-based student models into three categories, regarding the techniques researchers have used to build those models, though this distinction is not well-defined: (1) expert-centric (Gertner & VanLehn, 2000; Miselvy & Gitomer, 1996), (2) efficiency-centric (Murray, 1998; Reye, 1998), and (3) data-centric (Mayo & Mitrovic, 2001; Stern, Beck, & Woolf, 1999). In the expert-centric category, one or several experts specify the complete structure and CPTs of the Bayesian network, either directly or indirectly. So, the student model is an unrestricted product of domain analysis. In the efficiency-centric category, on the other hand, researchers partially specify or restrict the structure as well as the CPTs of the Bayesian network, and then domain knowledge is "fitted" to that network. Generally, the restrictions are chosen in such a way that can maximize several aspects of efficiency of that network (e.g., the evaluation time). In the third category, the system applies a number of techniques in machine learning to "learn" from available data (e.g., from data about interactions between the student and the system), so as to adjust the structure and/or the CPTs of the Bayesian network on the fly.

There is a significant difference between the approach we used to build our Bayesian network and the previous approaches: we use dynamic Bayesian networks (Russell & Norvig, 2009; Ghahramani, 1998) to model the *temporal dimension* explicitly and systematically. Indeed, all of the evolution variables, the correction variables, and the control variables in our Bayesian network are created in such a manner that models the learner's cognitive behavior and state by considering his or her sequence of actions and productions deliberately. In the future, we shall apply techniques in the previous approaches to make our network more efficient and effective in terms of diagnosis, for example, to consider "cognitive" relationships among control nodes, to use machine-learning techniques to improve CPTs.

CONCLUSIONS

Our main affirmation in this paper is that an appropriate use of dynamic Bayesian networks for modeling the temporal dimension together with fine-grained didactic analysis could be an effective way for student modeling and diagnosis, especially modeling the learner's cognitive behavior in complex domains such as orthopedic surgery. On the one hand, in those domains pragmatic knowledge plays an important role, particularly in better understanding the learner's cognitive behavior during a problem-solving process. Such knowledge, however, is not completely reported in standard instructional materials such as textbooks, and it is not easy to reveal it from practitioners. Therefore, we argue for a fine-grained didactic analysis to understand the nature of knowledge being taught in problem solving as completely as possible. Indeed, this paper has showed that the didactic analysis is useful in implementing a robust domain knowledge component for student diagnosis. On the other hand, dynamic Bayesian networks provide significant help in modeling the temporal dimension as completely as possible. It is also not difficult to integrate IF-THEN rules, which represent part of domain expertise, into temporal Bayesian networks. We believe that modeling both the student's knowing state and his or her cognitive behavior over time may help the system provide him or her with more relevant and fine-grained feedback, and consequently improve the student's learning outcomes.

In the future, we shall carry out empirical studies by using both quantitative and qualitative methods in order to know the full extent of the impact of our diagnosis component as well as of our learning environment on the student's learning (Rieber, 2005). For example, we shall compare diagnosis results of the system with those of expert teachers for a number of real students' problem-solving sessions (Webber, 2004). We may also use thinking-aloud methods (Leplat, 2000; Lewis, 1982) to compare the system's diagnosis results with what the learners speak aloud.

We shall also look further into the following three directions: (1) a method for the development of an easy-to-use authoring tool to help educational researchers in didactic engineering and reengineering, regarding our framework; (2) a method for automating the construction of IF-THEN rules and of the temporal Bayesian network from the results of the didactic analysis; and (3) the possibility to reuse our approach for other domains such as mathematics teacher education in which we are interested because recent studies in mathematics teaching have also indicated the importance of tacit knowledge (e.g., knowledge of student and content, knowledge of content and teaching, specialized content knowledge, see Ball, Thames, & Phelps, 2008) in the practice of teaching mathematics.

ACKNOWLEDGMENTS

We would like to thank the three anonymous reviewers for their valuable and constructive comments. We wish to thank the TCAN research program from CNRS (*Centre National de la Recherche Scientifique*, France) for partly supporting the current research work (the TELEOS project). We would like to thank Philippe Merloz, head of the Orthopedic Surgery Department of CHU (*Centre Hospitalier Universitaire de Grenoble*), and Laurent Couarraze, a master student in computer science, for their valuable assistance in our didactic analysis and prototype development respectively. We also wish to thank organizations that offer open-source tools and platforms, for example, GENIE (2006), JADE (2006), JENA (2006), so that we have been able to develop our prototype.

REFERENCES

Albacete P.L., & VanLehn, K. (2000). The Conceptual Helper: An Intelligent Tutoring System for Teaching Fundamental Physics Concepts. In G. Gauthier, C. Frasson, & K. VanLehn (Eds.), *Intelligent Tutoring Systems* (pp. 564–573). Berlin: Springer (Lecture Notes in Computer Science, Vol. 1839).

Anderson, J. R., Boyle, C. F., & Yost, G. (1986). The Geometry Tutor. *The Journal of Mathematical Behavior*, 5–20.

Anderson, J.R., Farrell, R., & Sauers, R. (1984). Learning to Program in LISP. *Cognitive Science*, 8, 87–129.

Balacheff, N., & Gaudin, N. (2010). Modeling Students' Conceptions: The Case of Function. *CBMS Issues in Mathematics Education,* 16, 207-234 (shortened and edited version of Balacheff, N., & Gaudin, N. (2002). *Students Conceptions: An Introduction to a Formal Characterization*. Grenoble, France: Les cahiers du laboratoire Leibniz, n°65. Retrieved from http://www-leibniz.imag.fr/LesCahiers).

Balacheff N., & Margolinas C. (2005). cK¢ Modèle des Connaissance pour le Calcul de Situation Didactiques. In A. Mercier & C. Margolinas (Eds.), *Balises pour la Didactique des Mathématiques* (pp. 1-32). Grenoble, France: La Pensée Sauvage.

Ball, D. L., Thames, M., and Phelps, G. (2008). Content Knowledge for Teaching: What Makes It Special? *Journal of Teacher Education*, 59(5), 389–407.

Brousseau, G. (1997). *Theory of Didactic Situations in Mathematics; Didactique des Mathématiques 1970-1990* (N. Balacheff, M. Cooper, R. Sutherland, & V. Warfield, Eds. & Trans.). Dordrecht, The Netherlands: Kluwer Academic Publishers.

Brown, J.S., & Burton, R. (1978). Diagnostic Models for Procedural Bugs in Basic Mathematical Skill. *Cognitive Science*, 2, 155–192.

Burton, R., & Brown, J.S. (1982). An Investigation of Computer Coaching for Informal Learning Activities. In D. Sleeman, & J. Brown (Eds.) *Intelligent Tutoring System*s. Orlando: Academic Press.

Chieu, V.M., Luengo, V., Vadcard, L., & Mufti-Alchawafa, D. (2007). A Framework for Building Intelligent Learning Environments in Ill-defined Domains. *International Workshop on AIED Applications in Ill-defined Domains*, *held in conjunction with The 13th International Conference on Artificial Intelligence in Education*, Marina Del Rey, CA, July 9-13, Web publication (www.cs.pitt.edu/~collinl/AIED07/, no page numbers).

Clancey, W.J. (1983). GUIDON. *Journal of Computer-Based Instruction*, 10(1), 8–14.

Clark, R.E., & Estes, F. (1996). Cognitive Task Analysis. *International Journal of Educational Research*, 25(5), 403–417.

Conati, C., Gertner, A., & VanLehn, K. (2002). Using Bayesian Networks to Manage Uncertainty in Student Modeling. *J. of User Modeling and User-Adapted Interaction*, 12(4), 371–417.

Confrey, J. (1990). A Review of the Research on Students Conceptions in Mathematics, Science, and Programming. In C. Courtney (Eds.) *Review of Research in Education* (pp. 3–56). American Educational Research Association (Vol. 16).

Eraut, M., & du Boulay, B. (2000). Developing the Attributes of Medical Professional Judgement and Competence. *Cognitive Sciences Research* (paper 518). Retrieved March 25, 2007 from http://www.cogs.susx.ac.uk/users/bend/doh.

Gertner, A., & VanLehn, K. (2000). Andes: A Coached Problem Solving Environment for Physics. In G. Gauthier, C. Frasson, & K. VanLehn (Eds.), *Proceedings of Fifth International Conference on Intelligent Tutoring Systems*, Springer-Verlag, pp. 133–142.

Ghahramani, G. (1998). Learning Dynamic Bayesian Networks. In *Adaptive Processing of Sequences and Data Structures, Lecture Notes in Artificial Intelligence*, pp. 168–197.

GENIE (2006). *Graphical Network Interface*. Retrieved March 25, 2007 from http://genie.sis.pitt.edu.

Henze, N., & Nejdl, W. (2001). Adaptation in Open Corpus Hypermedia. *International Journal of Artificial Intelligence in Education*, 12, 325–350.

Hersant, M., & Perrin-Glorian, M.J. (2005). Characterization of an Ordinary Teaching Practice with the Help of the Theory of Didactic Situations. *Educational Studies in Mathematics*, 59 (1-3), 113-151.

JADE (2006). *Java Agent Development Framework*. Retrieved March 25, 2007 from http://jade.tilab.com.

JENA (2006). *A Semantic Web Framework for Java*. Retrieved March 25, 2007 from http://jena.sourceforge.net.

Joab, M., Guéraud, V., & Auzende, O. (2005). Les Simulations pour la Formation. In M. Grandbastien & J.M. Labat (Eds.), *Environnements Informatiques pour l'Apprentissage Humain* (pp. 287–310). Paris : Hermès.

Kodaganallur, V., Weitz, R. R., & Rosenthal, D. (2005). A Comparison of Model-Tracing and Constraint-Based Intelligent Tutoring Paradigms. *International Journal of Artificial Intelligence in Education*, 15, 117–144.

Koedinger, K.R., Anderson, J.R., Hadley, W.H., & Mark, M.A. (1997). Intelligent Tutoring Goes to School in the Big City. *International Journal of Artificial Intelligence in Education*, 8, 30–43.

Lajoie, S.P., Faremo, S., & Wiseman, J. (2001). Identifying Human Tutoring Strategies for Effective Instruction in Internal Medicine. *International Journal of Artificial Intelligence in Education*, 12, 293–309.

Leplat, J. (2000). *L'Analyse Psychologique de l'Activité en Ergonomie : Aperçu sur son Evolution, ses Modèles et ses Méthodes*. Toulouse : Octarès.

Lewis, C. (1982). *Using the "Thinking Aloud" Method in Cognitive Interface Design* (IBM Research Rep. No. RC9265[#40713]). Yorktown Heights, NY: IBM Thomas J. Watson Research Center.

Lillehaug, S. I., & Lajoie, S. P. (1998). AI in Medical Education: Another Grand Challenge for Medical Infor-

matics. *Journal of Artificial Intelligence in Medicine*, 12(3), 1–29.
Luengo, V., Vadcard, L., Mufti-Alchawafa, D., & Chieu, V.M. (2007). Conceptions and Bayesian Network for an Adaptive Orthopedic Surgery Learning Environment. *The Second International Workshop on Personalisation for e-Health*, *held in conjunction with The 11th International Conference on User Modelling*, Corfu, Greece, June 25-29, pp. 39–46.
Luengo, V., Mufti-Alchawafa, D., & Vadcard, L. (2004). The Knowledge like the Object of Interaction in an Orthopaedic Surgery-Learning Environment. . In J.C. Lester, R.M. Vicari, & F. Paraguacu (Eds.) *Intelligent Tutoring Systems* (pp. 108–117). Berlin: Springer (Lecture Notes in Computer Science, Vol. 3220).
Margolinas, C., Coulange, L., & Bessot, A. (2005). What Can the Teacher Learn in the Classroom? *Educational Studies in Mathematics*, 59 (1-3), 205-234.
Mayo, M., & Mitrovic A. (2001). Optimising ITS Behaviour with Bayesian Networks and Decision Theory. *International Journal of Artificial Intelligence in Education,* 12, 124–153.
Mislevy, R., & Gitomer, D. (1996). The Role of Probability-Based Inference in an Intelligent Tutoring System. *User-Mediated and User-Adapted Interaction*, 5, 253–282.
Mitrovic, A., & Ohlsson, S. (1999). Evaluation of a Constraint-Based Tutor for a Database Language. *International Journal of Artificial Intelligence in Education,* 10, 238–256.
Mufti-Alchawafa, D., & Luengo V. (2009). *Design Implementation and Computer Validation of Didactical Decision Model in a Learning Environment for Orthopaedic*. Paper presented at the 2nd International Workshop on Intelligent Support for Exploratory Environments, in conjunction with the 14th International Conference on Artificial Intelligence in Education, Brighton, The United Kingdom.
Murray, T. (1999). Authoring Intelligent Tutoring Systems: Analysis of the State of the Art. *International Journal of Artificial Intelligence in Education,* 10, 98–129.
Murray, W. (1998). A Practical Approach to Bayesian Student Modeling. In B. Goettle, H. Halff, C. Redfield, & V. Shute (Eds.) *Proceedings of the Fourth International Conference on Intelligent Tutoring Systems*, Springer-Verlag, pp. 424–433.
Ohlsson, S. (1994). Constraint-Based Student Modeling. In J. E. Greer, & G. McCalla (Eds.) *Student Modeling: The Key to Individualized Knowledge-Based Instruction* (pp. 167–189). Berlin: Springer-Verlag.
Ohlsson, S. (1992). Constraint-based Student Modeling. *International Journal of Artificial Intelligence in Education*, 3(4), 429–447.
Pastré, P. (1997). Didactique Professionnelle et Développement. *Psychologie Française*, 42(1), 89–100.
Pearl, J. (2000). *Causality: Models, Reasoning, and Inference*. New York, NY: Cambridge University Press.
Piaget, J. (1985). *Equilibration of Cognitive Structures*. Chicago, IL: University of Chicago Press.
Reye, J. (2004). Student Modelling Based on Belief Networks. *International Journal of Artificial Intelligence in Education*, 14, 63–96.
Reye, J. (1998). Two-Phase Updating of Student Models Based on Dynamic Belief Networks. In B. Goettle, H. Halff, C. Redfield, & V. Shute (Eds.), *Intelligent Tutoring Systems* (pp. 274–283). Berlin: Springer (Lecture Notes in Computer Science, Vol. 1452).
Rieber, L.P. (2005). Multimedia Learning in Games, Simulations, and Microworlds. In R.E. Mayer (Eds.), *The Cambridge Handbook of Multimedia Learning* (pp. 549–567). New York, NY: Cambridge University Press.
Rogers, D.A., Regehr, G., Yeh, K.A., & Howdieshell, T.R. (1998). Computer-assisted Learning versus a Lecture and Feedback Seminar for Teaching a Basic Surgical Technical Skill. *American Journal of Surgery*, 175(6), 508–510.
Russell, S., & Norvig, P. (2009). *Artificial Intelligence: A Modern Approach* (3rd ed.). Upper Saddle River, NJ: Prentice Hall.
Sandholm, T.W. (1999): Distributed Rational Decision Making. In G. Weiβ (Eds.): *Multiagent Systems: A Modern Introduction to Distributed Artificial Intelligence* (pp. 201–258). Cambridge, MA: MIT Press.
Sasse, M.A. (1991). How to T(r)ap Users' Mental Models. In M.J. Tauber & D. Ackermann, *Mental Models and Human-Computer Interaction 2* (pp. 59–79). Amsterdam: Elsevier Science Publishers.
Schoenfeld, A. (1985). *Mathematical Problem Solving*. New York, NY: Academic Press.

Paiva, A., & Self, J. (1995). TAGUS - A User and Learner Modeling Workbench. *User Modeling and User-Adapted Interaction*, 4(3), 197-226.

Shulman, L. S. (1986). Those Who Understand: Knowledge Growth in Teaching. *Educational Researcher,* 15(2), 4-14.

Soury-Lavergne, S. (2003). *Baghera Assessment Project, Designing a Hybrid and Emergent Educational Society*. Grenoble : Les cahiers du laboratoire Leibniz (http://www-leibniz.imag.fr/LesCahiers).

Stern, M., Beck, J., & Woolf, B. (1999). *Naïve Bayes Classifiers for User Modeling*. Center for Knowledge Communication, Computer Science Department, University of Massachusetts.

Tonetti, J., Carrat, L., Blendea, S., Merloz, P., & Troccaz, J. (2003). Iliosacral Screw Placement With Ultrasound-Based Navigation versus Conventional Fluoroscopy. *Techniques in Orthopaedics*, 18(2), 184-191.

Vadcard, L. (2003). VOEU Pedagogical Strategy. *Final report, VOEU project, IST 1999-13079.*

Vadcard, L., & Luengo, V. (2005). Réduire l'Ecart entre Formations Théorique et Pratique en Chirurgie : Conception d'un EIAH. In P. Tchounikine, M. Joab, & L. Trouche (Eds.) *Environnements Informatiques pour l'Apprentissage Humain* (pp. 129–139). Paris : Institut National de Recherche Pédagogique.

Vergnaud, G. (1991). La Théorie des Champs Conceptuels. *Recherches en Didactique des Mathématiques*, 10(2/3), 133–170.

Vergnaud, G. (1981). Quelques Orientations Théoriques et Méthodologiques des Recherches Françaises en Didactique des Mathématiques. *Recherches en Didactique des Mathématiques*, 2(2), 215–231.

Wagner, R.K., Sujan, H., Sujan, M., Rashotte, C.A., & Sternberg, R.J. (1999). Tacit Knowledge in Sales. In R. J. Sternberg & J. A. Horvath, *Tacit Knowledge in Professional Practice: Researcher and Practitioner Perspectives* (pp. 155–182). Mahwah, NJ: Lawrence Erlbaum Associates.

Webber, C. (2004). From Errors to Conceptions – An Approach to Student Diagnosis. In J.C. Lester, R.M. Vicari, & F. Paraguacu (Eds.) *Intelligent Tutoring Systems* (pp. 710–719). Berlin: Springer (Lecture Notes in Computer Science, Vol. 3220).

Weber, G., & Brusilovsky, P. (2001). ELM-ART: An Adaptive Versatile System for Web-based Instruction. *International Journal of Artificial Intelligence in Education*, 12, 351–384.

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