Constructive and Collaborative Learning Environments What Functions are Left for User Modelling and Intelligent Support?

Martin Mühlenbrock, H. Ulrich Hoppe
Dept. of Mathematics / Computer Science
University of Duisburg, 47048 Duisburg, Germany
Email: muehlenbrock@informatik.uni-duisburg.de

Abstract. With constructive and collaborative learning environments being current trends in tutoring systems, the function of user modelling and intelligent support is not as well-defined as in traditional intelligent tutoring systems. For practical solutions, there tend to remain only some niches for the application of intelligent components. To investigate the practicability of intelligent functions, we have build a framework architecture that allows for the implementation of self-contained constructive and collaborative learning environments. These learning environments are open in such a way that intelligent components can be added as agents wherever this seems reasonable. By means of two sample applications we have investigated the question of how much intelligent support is tractable and appropriate in these environments.

1 Introduction

By progressing from single-user intelligent tutoring systems to multiple-user collaborative and constructive learning environments, the role of the system as an intelligent component in the tutoring process appears not as clear cut as before. In these more flexible learning environments, where the students have a greater scope for following their own goals and their own understandings, the role of the student model is more subtle but not non-existent [9]. The availability of human support and peer-to-peer cooperation can help to avoid existing problems in generating adequate, personally meaningful feedback to learners, leaving the actual tutoring to human-human interaction of different types [4]. The traditional role of intelligent tutoring systems of mainly identifying misunderstandings and remediation may be filled more flexible in human-human communications and may lead to ultimately more successful repair strategies [10]. As a consequence, the idea of intelligent systems has shifted towards intelligent components. However, the necessary and sufficient functionality of these more or less ubiquitous intelligent components is still not very well defined. Two extreme positions are conceivable and have also been investigated in practice. The first one, the renunciation of any intelligent component, is being widely put into practise in a huge number of modern 'multimedia' learning environments. The second one, the resurrection of single-user intelligent tutoring systems as personified, communicating and collaborating agents in a group of learners, seems to be even more demanding and critical as the traditional counterpart.

The development of an intelligent tutoring system is a hard and complex process. Several sophisticated ingredients are necessary which have to be combined in a reasonable way to come up with a system that is really capable of helping the users in learning something. First, an explicit and descriptive domain model is needed. It forms the basis of the intelligent tutoring system, since the learners' knowledge will be related to the distinguishable components of the domain model. Fortunately, in many domains, especially in basic scientific areas, there are models or theories that describe phenomena in a consistent way. However, in many interesting yet controversial fields there are no such analytic models that could form an adequate basis for a tutoring system. Second, a cognitive model of learning is needed, that relates new or contradictory information to the learner's knowledge revision processes. As research has shown, even novice learners are no blank slates but bring at least a fragmentary commensense model to the learning task. Unfortunately, besides from more or less hypothetical cognitive models in narrow domains, such as e.g. explanations for the day/night cycle, we do not have a clear theoretical basis that describes the influence of new knowledge on pre-existing conceptual structures in a general way. Third, a discourse model is needed, since the system itself as well as humans, such as a tutor or fellow learners in a group setting, communicate with the learner. Although this area of research has made considerable progress, we still lack a descriptive and operational model for many aspects. Fourth, all three kinds of models that have been mentioned above have to be integrated in combination in a tutoring system to provide a descriptive model of tutoring and hence incorporate an intelligent approach to tutoring.

Taking all this into account, the development of personified, communicating, and collaborating agents certainly is a very interesting and demanding field of research, that can only be tackled concentrating on some small and possibly isolated aspects. Further investigations may lead to better insights and may help to improve the necessary theories and models. This is rather an interdisciplinary research approach which draws heavily on new results in educational psychology to built computerised simulations, which then stipulate further research. However, this approach does, at least currently,

ECAI 98 Workshop Programme.

not lead to practical solutions, i.e. constructive and collaborative learning environments that can really be used. Hence, the middle course between the above mentioned extremes is to drastically simplify the demands in the capabilities of tutoring systems and incorporate them as intelligent subparts in learning environments. Within those environments, there are three important issues that determine each other, namely collaboration, construction, and intelligence (see figure 1). Few learning environments have ever addressed these three issues to a conceivable extent at the same time, and there seems to be a certain degree of mutual exclusiveness in them.

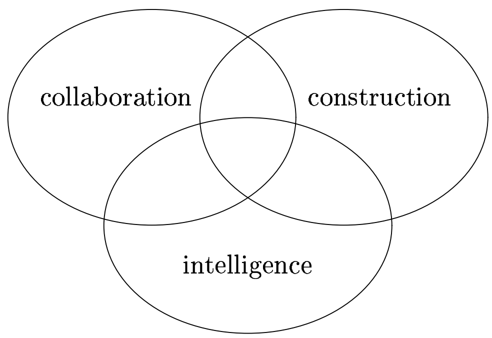

Figure 1. Important issues in learning environments

In order to investigate the interplay of these issues in practical solutions, in particular the practicability of intelligent functions, we have build a framework architecture that allows for the implementation of self-contained constructive and collaborative learning environments. These learning environments are open in such a way that intelligent components can be added as agents wherever this seems reasonable. By means of two sample applications we have tackled the question of how much intelligent support is appropriate in these environments beyond other tasks such as the provision of system connectivity and interface design. Above all, two questions are of particular interest, i.e. (1) whether the task of deriving learner or user models from ongoing group activities [7] is subsumed by single-user modelling approaches and (2) whether there is a general tendency of externalising functions formerly being internal as subprocedures in intelligent tutoring systems and thereby turning CSCL systems into CSCW systems.

2 Framework System

The collaborative learning environments for constructive tasks are based on shared workspaces using two-dimensional visual representations. We have developed a framework system for constructing shared workspace environments as networks of cards for different purposes [2]. Figure 2 shows a sample CardBoard application in the domain of mechanics. To introduce variables, constants, or operators, the learners select cards, connect the operands, and arrange the mathematical formulae. The term card is used as a general denominator for any basic object in the workspace environment. The users work with cards in private or shared workspaces. Private workspaces are only accessible by single users. With drag & drop operations, cards can be freely moved or copied from one workspace into another. Shared workspaces support collaborative problem solving. During a collaborative session, the content of shared workspaces is synchronised by broadcasting and multiplying actions on objects according to the approach of shared user interface objects [12]. A general groupware construction kit called MatchMaker supports the development of distributed applications. Its library implements classes of user interface objects with built-in protocols for synchronisation (coupling). A fixed communication protocol allows for distributing components over heterogeneous system platforms.

The diagrammatic structures or so-called visual languages in the user interfaces are primarily used as a medium of communication and representation (external memory) in cooperative work and collaborative learning. The user interfaces are inherently constructive, since for each workspace, the user can select and even design a specific visual language from a set of predefined language specifications. This will generate a palette for the corresponding card set and a new workspace. Each workspace will be initialised with a single language to avoid ambiguities and inconsistencies. The distinction between connector and content cards is essential for the language definition. Content cards are containers for domain-specific data, whereas connectors define the general structure of the language. These card nets are represented as graph structures with n-ary relations, as in the formal concept of a hypergraph, and labeled slots or links associated with these relations.

Operational semantics and intelligent support can be added to a workspace environment using a general architecture called Dalis [6]. Different components for interpretation and for individual and group support can be plugged in as Prolog-based agents into the distributed Dalis system. The system establishes a federation architecture that facilitates the message flow between its internal agents and the user interfaces that are connected to the MatchMaker server. A specific type of agent monitors continuously all the modifications in the card workspaces. It then translates the current card net into the internal formalism of an underlying interpreter based on a descriptive representation of the card symbols. By means of the Dalis system, a number of existing user modelling implementations can be incorporated into the overall architecture. However, as has been described above, the user interface environment is self-contained concerning the construction and modification of card nets with no further feedback from the system. This opens up the opportunity to investigate how much of the functionality of traditional intelligent tutoring systems is really needed in such environment, if at all, and if the need for new functions arises that are clearly not covered by those systems.

3 Sample Applications

Based on the framework architecture, two sample applications among others have been developed. The first application, CardDeriv, has been ported from a former single-user intelligent tutoring system, the second one, TheGap, has been specifically developed to exploit the features of the framework system.

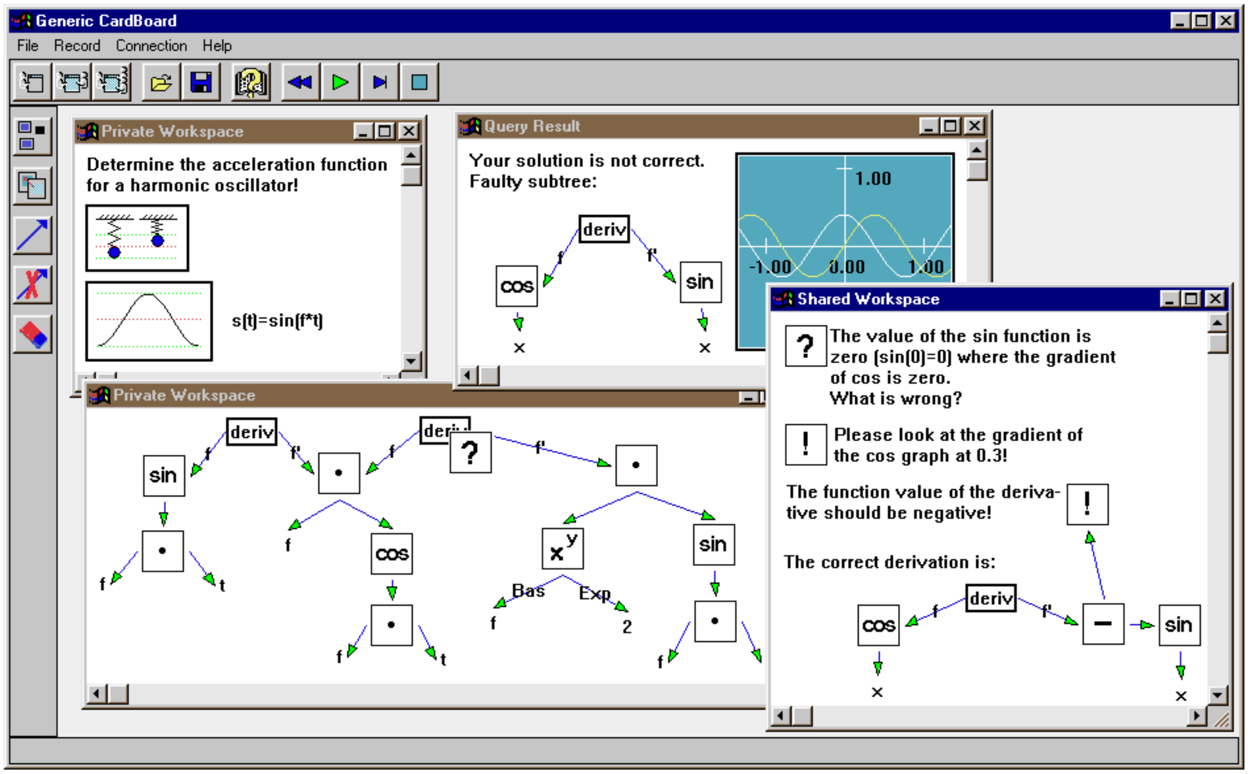

Figure 2. An environment for derivation tasks in mechanics

CardDeriv The application CardDeriv is an intelligent tutoring system in the field of mathematics (symbolic differentiation) and Physics (mechanics). Figure 2 gives an impression of what kind of visual languages are used in this context and how different types of workspaces are used in the CardBoard environment. The current state of the problem solution (in the lower left), a given exercise, and the feedback of the intelligent subsystem (Query Result) are located in private workspaces. The intelligent subsystem gives feedback to a single student working in a private workspace or to a group of students working together in a shared workspace and hence appear as a single learner. The diagnostic procedure follows the reconstructive approach of deductive diagnosis [3]. Given a correct and complete domain theory, in this case of symbolic differentiation, an incorrect student solution is indicated by an unprovable goal in the process of reconstructing a correct solution using a fail-safe meta-interpreter. While backtracking on different sets of error conditions, the algorithm determines structural conditions of erroneous examples. Since the derivation tutor has been implemented in Prolog, it was easy to integrate the tutor into the DALIS architecture. For each student, an agent is automatically invoked which keeps track of the student model. Besides the function of determining deviations from correct symbolic derivatives, the subsystem also gives hints on undefined variables and provides a lookup of keywords [11].

All these functions are typical for single-user intelligent learning environments. For group settings, there are further functions which profit from the existence of the individually assessed user models. The first one allows the teacher or tutor to directly inspect the user model. Moreover, the system abstracts from details and aggregates the individual user models to a combined overview of the learners' performance. As a second function, the data can be used to organise a group setting in regards of the composition of the groups or the selection of exercises a group is supposed to be able to solve. This function is based on the integration of individually assessed learner models has been termed 'parameterisation of group learning' [4]. The support function of the system ends at this point, apart from the fact that the learner can open a shared workspace which they can use to solve the task.

TheGap The prototype application THEGAP is based on microworlds that consist of objects of different type (see figure 3). These microworlds are visible to all users as they are implemented as shared workspaces. Each user is assigned a specific object or a small set of objects in a microworld that he can control by defining a set of rules. Furthermore, object behaviour in a microworld is constrained by an additional set of general rules. For each microworld, a task is given that can be solved by a sequence of object movements. The difficulty lies in the fact that these tasks cannot be solved by a single object alone but rather through the collaboration of several objects. The general learning goals are learning to solve a given problem cooperatively and learning to predict the behaviour of objects whose actions are defined by explicit rules.

The specific microworld consists of bricks that form sort of a gap and of blocks that are to be controlled by rules. Each user specifies rules for his block in a workspace as shown on the right. The sample rules given represent the concept

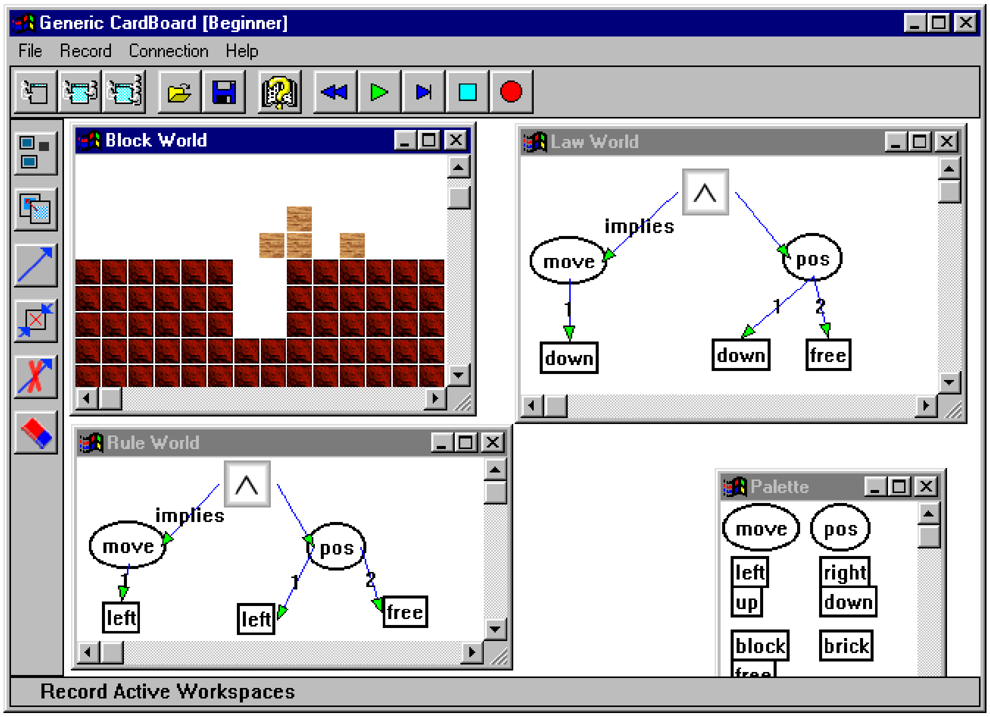

Figure 3. Application THEGAP with three different workspaces and a card palette

that the user's block moves to the left as long as there are bricks below. At first, all blocks rest on one side of the gap, and the task is to bridge the gap and reach the other side. No single block can cross the gap alone. This is assured by rules and constraints given in a further workspace presented at the bottom. It defines the general laws that apply in the microworld. In this case it rules out the possibility of first descending to the bottom of the gap and then climbing up the other side. This task can only be solved by concerted actions of all blocks. Therefore, each user has to model the behaviour of his blocks in such a way that they collaborate with the other blocks.

This application is specified to address all the three aspects mentioned above. It is constructive and open-ended, as there are several possibilities to solve the goal, and even the goal and the constraints can be specified by the users (in the first place, these are predefined). It is collaborative since the problem cannot be solved by a single person but only in a group. Moreover, if the goal cannot be achieved by each user specifying the behaviour of his own object or agent, the users have to observer the behaviour of the other blocks and must communicate on the structure and content of rules that may lead to a solution. They even have to find a consensus on the strategy, as there are several strategic options for goal achievement. Finally, this application relies on user modelling, though this is the least obvious aspect of this application and will be discussed in the following. Generally speaking, there may be situations in the collaborative problem solving process that lead to dead ends and that can only be overcome by an overview of all the users' definitions and the consequences thereof for the achievement of the ultimate goal.

The constructive nature of this application impedes the provision of a complete descriptive subject matter or domain model. Instead, a procedural interpretation is given to the basic relations in such a way that the users' definitions are carried out in the microworld The formal representation of a user's ideas of how the problem at hand can be solved could be taken as an explicit user model which is constructed by the learner himself and that could be inspected by a tutor or by fellow learners.

4 Discussion

MBR and student modelling are currently less, or at least less obviously, in the focus of innovation in AI in education research as they used to be, current 'hot spots' being defined by topics such as collaborative systems, networked educational environments, or personified agents (which are not necessarily multi-agent systems in the sense of Distributed AI, cf. e.g. [5]). As every trend will be replaced by another, it is absolutely fair and reasonable to conserve and further elaborate MBR and student modelling as something different and separate from these current tendencies. In our research, however, we try to combine the striving for open, constructive, and collaborative environments with the provision of intelligent support functions based on analytic models that incorporate domain knowledge and reflect human learning and problem solving strategies. Indeed, we accept this as a specific challenge that might advance our intelligent technologies as well as cognitive models of learning.

In the history of ITS, we have seen a parallel co-development of analytic, computational theories of learning and problem solving under the heading of 'cognitive architectures' on the one side and of intelligent tutors for a variety of domains on the other side. An important share of ITS explicitly claimed to be based on specific cognitive theories and architectures (e.g. [1]). These system aimed at supporting individualised learning, just as the theories were theories of individual problem solving and learning. Cognitive models in the sense of partial computational theories of cognition have not yet been extended to cope with learning and problem solving in groups. Providing model-based support in collaborative environments with open and constructive tasks might be a concrete stimulus for developing these extensions (a first approach of this type is described in [8]).

Yet, extending model-based support to both open task environments and collaborative activities makes 'closed-loop tutoring' [4] intractable. Neither are we practically able to stringently interpret any user action in an open and constructive task environment, nor can we completely monitor and analyse group interactions. This induces a realistic turn in intelligent learning support: Intelligent support will be partial, i.e. not always available, and it must be integrated with a scenario in which human or artificial agents may assume supportive roles. Some roles (particularly those requiring deep 'mutual understanding' and insight in human dispositions) may even be reserved for humans.

Based on this philosophy or view of intelligent support systems, we work towards the following goals:

- developing multi-agent architectures for the distributed monitoring and modelling of distributed learning activities (based on the assumption that the topology of the intelligent subsystem should reflect the topology of the learner's network);
- providing plug-in mechanisms for partially adding modeling and support components to (group-) interactive environments;
- developing tools or toolkits for defining and implementing concrete sharable environments on a high level of abstraction that come with an interface to the modeling subsystem.

The examples mentioned illustrate our achievements along these lines.

Concerning the kind of models needed in these environments, we would distinguish the following types:

- domain models: As opposed to classical ITS, they are not primarily used to generate 'expert solutions', but to provide a semantic interpretation of user actions in open environments (e.g. a given visual language is to be interpreted in logical terms).
- models of goal-directed behavior: They incorporate problem solving methods and strategies, particularly 'strong methods' that reflect specificities of the domain. They would allow not only to semantically interpret a single action but also to assess it 'in context' as goal-directed behaviour. It is, e.g., a big challenge in the Gap environment to interpret why users add specific behavioural rules for their blocks.

As for individual user and belief models, it is unclear to us of how much use they can be before there is a solid basis on the other two levels. In our current work, we can manage domain models flexibly in open and constructive environments. Models of goal-directed behaviour appear more to be in the 'zone of proximal development' (from a research perspective).

REFERENCES

[1] J.R. Anderson, A.T. Boyle, A. Corbett, and M. Lewis, 'Cognitive modeling and intelligent tutoring', *Artificial Intelligence*, **42**, 7–51, (1990).

[2] Katrin Gassner, Frank Tewissen, Martin Mühlenbrock, Andreas Loesch, and H. Ulrich Hoppe, 'Intelligently supported collaborative learning environments based on visual languages: A generic approach', in *Proceedings of the Third International Conference on the Design of Cooperative Systems COOP-98*, INRIA, Cannes, France, (May 1998).

[3] H. Ulrich Hoppe, 'Deductive error diagnosis and inductive error generalization for intelligent tutoring systems', *Journal of AI in Education*, **5**(1), 27–49, (1994).

[4] H. Ulrich Hoppe, 'The use of multiple student modeling to parameterize group learning', in *Proceedings of the 7th World Conference on Artificial Intelligence in Education AIED-95*, ed., Jim Greer, pp. 234–241, Charlottesville, VA, (August 1995). AACE.

[5] J.C. Lester, A.C. Converse, B.A. Stone, S.E. Kahler, and S.T. Barlow, 'Animated pedagogical agents and problem-solving effectiveness: A large-scale empirical evaluation', in *Proceedings of the 8th World Conference on Artificial Intelligence in Education AIED-97*, eds., B. du Boulay and R. Mizoguchi, 23–30, IOS Press, Kobe, Japan, (August 1997).

[6] Martin Mühlenbrock, Frank Tewissen, and H. U. Hoppe, 'A framework system for intelligent support in open distributed learning environments', in *Proceedings of the 8th World Conference on Artificial Intelligence in Education AIED-97*, eds., B. du Boulay and R. Mizoguchi, 191–198, IOS Press, Kobe, Japan, (August 1997).

[7] Ana Paiva, 'Learner modeling for collaborative learning environments', in *Proceedings of the 8th World Conference on Artificial Intelligence in Education AIED-97*, eds., B. du Boulay and R. Mizoguchi, 215–222, IOS Press, Kobe, Japan, (August 1997).

[8] R. Plötzner, H. U. Hoppe, E. Fehse, C. Nolte, and F. Tewissen, 'Model-based design of activity spaces for collaborative problem solving and learning', in *Proceedings of the European Conference on Artificial Intelligence in Education*, eds., P. Brna, A. Paiva, and J. Self, 372–378, Colibri, Lisbon, Portugal, (1996).

[9] John A. Self, 'Formal approaches to student modelling', in *Student Modelling: The Key to Individualized Knowledge-Based Instruction*, eds., J. E. Greer and G. I. McCalla, 295–352, Springer, (1994).

[10] L.A. Suchman, *Plans and Situated Actions: The Problem of Human-Machine Communication*, Cambridge University Press, Cambridge, MA, 1987.

[11] Frank Tewissen, *Begriffsnetze als Basis für ein System zur kooperativen Lösung physikalischer Aufgabenstellungen (Concept Maps as a Basis for a System Supporting the Cooperative Solution of Physics Problems)*, Master's thesis, University of Duisburg, Dept. of Mathematics/Computer Science, 1996. in German.

[12] J. Zhao and H. Ulrich Hoppe, 'Supporting flexible communication in heterogeneous multi-user environments', in *Proceedings of the International Workshop on the Design of Cooperative Systems*, 442–449, Juan-les-Pins, France, (January 1995).