8

Structure and Design of an Intelligent Tutorial System Based on Skill Assignments

Dietrich Albert
Karl-Franzens-Universität Graz

Martin Schrepp
Ruprecht-Karls-Universität Heidelberg

Since the beginning of research in artificial intelligence several attempts have been made to construct intelligent tutorial systems (ITS). Such an ITS consists in general of a representation of the knowledge in a special domain, a diagnostic procedure to determine the knowledge of a student working with the system, teaching material, and a procedure for adaptive teaching. This chapter demonstrates how an extension of the theory of knowledge spaces can be used for the design of a domain-independent ITS. The main components of this ITS are a representation of the skills necessary in the knowledge domain and their dependencies as a surmise system, a set of questions related through a skill assignment to the skill states used for knowledge diagnosis, and a rule that relates skill states to teaching operations. The ITS is adaptive with respect to the consideration of the prior knowledge a student possess and with respect to the learning speed of a student. The strict formalized description of the systems components and their interactions during the teaching process guarantees that an ITS with the described properties can be implemented easily.

INTRODUCTION

One of the research topics discussed in artificial intelligence is the approach to use computers in education for diagnosing knowledge (for example Brown & Burton, 1978) or teaching (e.g., Mandl & Hron, 1985; Okamoto & Matsuda, 1993; Sleeman & Brown, 1982).

Intelligent tutorial systems (ITS) differ from other forms of computer-assisted-instruction (e.g., drill-programs, electronic textbooks, or simulation programs) in being "intelligent" with respect to two criteria. First, the system must be able to diagnose the status of the student's knowledge in the underlying knowledge domain. Second, the tutorial strategy must be adaptive to the prior knowledge of the student and must be able to improve the learner's knowledge.

There exists a variety of architectures for ITS in different knowledge domains. Most of the constructed systems work in highly formalized domains, like subdisciplines of mathematics, for example, symbolic logic (Matsuda & Okamoto, 1992), arithmetic (Takeuchi & Otsuki, 1994), fraction calculation (Kondo, Watanabe, Takeuchi, & Otsuki, 1990), integration (Kimball, 1982), physics (Ploetzner & Spada, 1993), language acquisition (Kunichika, Takeuchi, & Otsuki, 1994), or the usage of software systems or programming languages (Heines & O'Shea, 1985; Yasuda & Okamoto, 1991).

Most of these architectures are highly specialized in their specific knowledge domain and it is, therefore, not possible to use them in other domains.

There is a wide agreement in literature that an ITS must contain at least four basic components, that interact during the teaching process. These components are a *knowledge base*, a *student model*, a *teaching component*, and a *diagnostic component*. Existing ITS differ in the way in which these components are conceptualized, especially in their in general domain specific student models.

The knowledge base contains all relevant knowledge of the domain for which the ITS is constructed. This component represents the knowledge of an expert in the domain and is often realized as an expert system.

The student model is a representation of the cognitive abilities or skills of students in the knowledge domain. It consists, in general, of a set of states. Each state represents a special state of knowledge in which a student may possibly be during the learning process, that is during his or her interaction with the ITS. The learning process is represented by a sequence of states of the student model.

The teaching component consists of teaching materials (instructions, exercises, or demonstrations) and a rule that determines for each state of the student model, which part of this material is relevant for a student who is in that particular state.

The state of knowledge of a student is, in general, not directly observable because the student model contains hypothetical assumptions about necessary cognitive abilities or cognitive procedures. The diagnostic component is used to infer this state from the interaction of the student with the ITS.

The goal of the teaching process is to enable the student to learn all relevant information of the knowledge domain (or, at least, a sufficient large subset of this information), so that her or his knowledge after the teaching process terminates is more or less equal to the knowledge contained in the knowledge base of the ITS.

We want to show how the theory of knowledge spaces (Doignon & Falmagne, 1985, 1998) and skill assignments (Doignon, 1994a; Lukas & Albert, 1993; Korossy, 1993) can be used to develop a general architecture of an ITS. This general architecture is domain independent. Therefore it can be used for the concrete construction of ITS's in different knowledge domains.

Two properties of the theory of knowledge spaces and skill assignments are especially important for this approach. First, the theory of knowledge spaces allows the formulation of effective and adaptive procedures for the diagnosis of knowledge (see Falmagne & Doignon, 1988; Doignon, 1994b). Second, skill assignments allow a precise description of the connection between observable behavior in problem solving and underlying cognitive abilities or skills (Doignon, 1994a; Korossy, 1993, 1997).

In the following section, we describe the basic ideas of an ITS based on the theory of knowledge spaces and skill assignments. Then, we show how this basic system can be generalized by incorporating some important new components. In the last section possibilities for further developments are sketched and open questions leading to further research are discussed.

BASIC IDEAS OF AN ITS BASED ON SKILL ASSIGNMENTS

In this section, we develop the basic architecture of an ITS based on skill assignments. This basic architecture is very simple and therefore limited in its applicability because the main goal of this section is to clarify the basic ideas underlying our approach.

For the formal description of our basic architecture we have to introduce some notations. We write in the following Pow(X) for the power set, the set of all subsets of a set X. Seq(X) denotes the set of all tuples of elements of a set X with Seq$(X) := \{(x_1, \ldots, x_n) \mid n \in \boldsymbol{N} \wedge x_1, \ldots, x_n \in X\}$.

We divide the components of our ITS into *statical* and *dynamical* components. A component is called dynamical if it is updated during the teaching process, thus being adaptive to the student's behavior, and statical if this is not the case. We begin with the description of the statical components.

Knowledge Base

The knowledge necessary in the domain is represented through a set S of *skills*. Dependencies between these skills are described by a surmise function $\sigma : S \rightarrow \text{Pow}(\text{Pow}(S))$. A surmise function assigns to each skill the different sets of prerequisites of this skill. We interpret $\sigma(s) = \{S_1, \ldots, S_n\}$ as "Every student has to master all skills from at least one of the subsets $S_1, \ldots, S_n$ of S to be able to reach mastery of s". The surmise function σ is used during the teaching process to determine an optimal learning path. The surmise system (S, σ) can be considered as the *knowledge base* of the system.

Student Model

Given S and σ, we can define the *student model* $\mathcal{S}$ by:

$$\mathcal{S} := \{S' \subseteq S \mid \forall s \in S' \; \exists S'' \in \sigma(s) \; (S'' \subseteq S')\},$$

$\mathcal{S}$ is the set of all subsets of S that contain with every skill s at least one set S'' of prerequisites of s. $\mathcal{S}$ is the set of all subsets of S consistent with the dependencies of skills described by σ. Therefore, the knowledge of every student working with the system can be described by an element of $\mathcal{S}$, if we presuppose that S contains all the skills relevant[1] in the knowledge domain and that σ describes the dependencies between these skills correctly. If the knowledge of a student is described by $S' \in \mathcal{S}$ we call S' the *competence state* of that student.

Diagnostic Component

The skills in S are hypothetical and are not directly observable constructs. Therefore, our system must contain a possibility to infer the skills a student possess, the student's competency, from his or her solving-behavior.

Let Q be a set of questions or problems in the underlying knowledge domain. We assume in the following that the questions in Q are especially constructed for the diagnosis of the skills in S. Thus, we can assume that the mastery of all skills in S is sufficient for the ability to solve all questions in Q. None of the questions in Q require a skill $s \notin S$. The systematical construction of questions for a given set of skills is discussed for example in Lukas and Albert (1993), Held (1993), Albert, Schrepp, and Held (1994), or Korossy (1993, 1997).

Because we want to diagnose the skills a student possesses from her or his solving behavior, we have to relate the skills in S to the ability of students to solve the questions in Q. This is done by a *skill assignment* $\alpha : Q \rightarrow \text{Pow}(\mathcal{S})$. For a

[1] Relevant skills are the skills that form the specific body of knowledge in the underlying knowledge domain. The ability to add two integers is an example of such a relevant skill in the domain of elementary arithmetic. General skills that are necessary for understanding the instructions or problems of the domain, such as the ability to read, are presupposed and not considered as members of S.

question q, we interpret $\alpha(q) = \{S_1, \ldots, S_n\}$ as "A student answering correctly to question q must master all skills from at least one of the sets $S_1, \ldots, S_n$". Because of this interpretation we can assume that the states $S_1, \ldots, S_n$ of the student model contained in $\alpha(q)$ are minimal with respect to $\subseteq$, for all $S_i, S_j \in \alpha(q)$ we have $S_i \nsubseteq S_j$. For a more detailed discussion of the dependency between the skills possessed by a student and her or his solving behavior, see Korossy (1993, 1997).

The skill assignment determines for each state $S' \in \mathcal{S}$ of the student model the answer behavior $\beta(S')$ consistent with this state by

$$\beta(S') := \{q \mid q \in Q \wedge S' \in \alpha(q)\},$$

$\beta(S')$ is the set of all questions in Q that can be answered correctly by a student with competency S'.

Teaching Component

To allow teaching operations, the system must contain materials that can be presented to students in order to improve their knowledge. Let I be a set of instructions, like examples, facts, informational texts, or exercises. We represent I as a union of pairwise disjoint sets $I = I_1 \cup \ldots \cup I_n$, with $I_l \cap I_m \neq \emptyset$ if and only if $l = m$. Each of these sets I_l represents a special type of instruction.

The teaching operations should be adaptive to the prior knowledge of a student. Therefore, we have to relate them to the skills in S. This relation is established by a function $\delta : S \times \text{Pow}(S) \rightarrow \text{Seq}(I)$, which relates skills and their prerequisites to an instructional sequence. We interpret $\delta(s, S') = (i_1, \ldots, i_n)$ for $n \in \boldsymbol{N}$ as "The instructional sequence $i_1, \ldots, i_n$ should be presented to a student who masters all skills in $S' \setminus \{s\}$ and who does not master s". (I, δ) can be considered as the *teaching component* of the system.

Hypothetical Student States

To enable an adaptive teaching strategy, the system must also contain dynamical components, that is components that change during the interactions of a student with the system.

The actual information of the system about the knowledge of a student working with it can be represented as a subset $\mathcal{M}$ of $\mathcal{S}$. This subset $\mathcal{M}$ consists of all states of $\mathcal{S}$ that are consistent with the previous answers of the student on the presented questions. We call $\mathcal{M}$ the *set of hypothetical student states*. If $q_1, \ldots, q_n \in Q$ are the questions already answered correctly by a student and $q'_1, \ldots, q'_m$ the questions answered incorrectly by that student we have:

$$\mathcal{M} = \{S' \in \mathcal{S} \mid \forall i = 1, \ldots, n \ (q_i \in \beta(S')) \wedge \forall i = 1, \ldots, m \ (q'_i \notin \beta(S'))\}.$$

Storage of Information

The interactions between student and system must be stored to use them for teaching decisions. For example, which questions the student has already answered must be stored and also which information was already presented to him or her. This is done in the history H of the teaching process. H is a list of all previously presented questions, obtained answers, and presented instructions. Thus, H is an element of Seq$((Q \times R) \cup I)$, where $R = \{\text{correct}, \text{false}\}$. For example $H = ((q_1, r_1), (q_2, r_2), i_1, i_2, i_3, (q_3, r_3), \ldots)$ means that first question q_1 was presented and answer r_1 was obtained, second question q_2 was presented and answer r_2 was obtained. Afterwards, instructions i_1, i_2, i_3 were presented to the student, followed by a presentation of question q_3, on which answer r_3 was obtained, and so on.

Teaching Process

Up to this point, we have described only the components of the systems architecture. Now, we describe how these components interact during the teaching process. The teaching process can be considered as an interaction between five modes of the system.

The first mode is the START mode. Here the system sets $\mathcal{M} = \mathcal{S}$ because, up to this point, no information about the students knowledge is obtained. Then, the system changes into the second mode, called DIAGNOSIS mode.

The DIAGNOSIS mode consists of three steps. In the first step, the system tries to determine a question q, for which an obtained answer guarantees an optimal reduction of the set of hypothetical student states consistent with the obtained answer of the student. This can be done by the half-split procedure in the diagnostic algorithm described in Falmagne and Doignon (1988).

For the set $\mathcal{M}$ of hypothetical student states and a question $q \in Q$, we define $\mathcal{M}_{q,1} \subseteq \mathcal{M}$ as the set of all elements of $\mathcal{M}$ that are consistent with the fact that the student solves question q and $\mathcal{M}_{q,0} \subseteq \mathcal{M}$ as the set of all elements of $\mathcal{M}$ that are consistent with the fact that the student fails in solving q. Formally, we have $\mathcal{M}_{q,1} = \{M \in \mathcal{M} \mid q \in \beta(M)\}$ and $\mathcal{M}_{q,0} = \{M \in \mathcal{M} \mid q \notin \beta(M)\}$. The sets $\mathcal{M}_{q,1}$ and $\mathcal{M}_{q,0}$ are pairwise disjoint, that is we have $\mathcal{M}_{q,1} \cap \mathcal{M}_{q,0} = \emptyset$ and we have $\mathcal{M}_{q,0} \cup \mathcal{M}_{q,1} = \mathcal{M}$.

We define a question $q' \in Q$ as being *optimal for diagnosis* if the value of $\mid \text{card}(\mathcal{M}_{q',1}) - \text{card}(\mathcal{M}_{q',0}) \mid$ is minimal over all $q \in Q$. For an optimal question q' the cardinality of the sets $\mathcal{M}_{q',1}$ and $\mathcal{M}_{q',0}$ is optimal balanced, that is the sets $\mathcal{M}_{q',1}$ and $\mathcal{M}_{q',0}$ differ in cardinality as little as possible.

The described procedure can be regarded as the *diagnostic component* of the system. Notice that a question already answered (correct or incorrect) by the student can not be optimal as long as the set $\mathcal{M}$ of hypothetical student states contains at least two elements.

In the second step, an optimal question q, determined by the first step, is presented to the student. If the answer of the student is $r \in \{0, 1\}$ (where 1 represents "correct" and 0 "false") then $\mathcal{M}$ is replaced by $\mathcal{M}_{q,r}$.

The third step tests if there exists a skill s not mastered by the student, (formally: $\forall M \in \mathcal{M}(s \notin M)$), for which a set of prerequisites is completly mastered, (formally: there exists a $S' \in \sigma(s)$ with $\forall M \in \mathcal{M}(S' \setminus \{s\} \subseteq M)$). If this is the case, then the system changes to the TEACH mode. If this is not the case, the system continues with the first step of the DIAGNOSIS mode.

The TEACH mode starts if the DIAGNOSIS mode has detected a skill s not mastered by a student who masters for a set S' of prerequisites of s all skills from $S' \setminus \{s\}$. Formally, this means that $\forall M \in \mathcal{M}(s \notin M)$ and $\forall M \in \mathcal{M}(S' \setminus \{s\} \subseteq M)$.

It is possible that the DIAGNOSIS mode has detected more than one skill with this properties. If this is the case, the system has to decide which of these skills should be taught first. We discuss this problem in the following section. For the moment, it is sufficient to assume that the decission is made by chance.

The TEACH mode then presents the instructional sequence $\delta(s, S')$ to the student. After the presentation of this instructional sequence the system changes to the TEST mode.

The TEST mode consists of three steps. In the first step the system checks if the student has reached mastery of the last skill s taught by the TEACH mode. Therefore, a question is presented for which skill s is necessary and only skills from S' are required, where S' denotes the set of prerequisites to s detected in the third step of the DIAGNOSIS mode. Formally, this means that the system presents a question q with $\forall S'' \in \alpha(q)(s \in S'' \wedge S'' \subseteq S')$.

In the second step, the system checks if the presented question is answered correctly by the student, indicating that he or she had learned the required skill. If the student fails in answering correct, that is if $r \neq 1$, the system goes back to the TEACH mode and starts the presentation of $\delta(s, S')$ again. If the student gives the correct answer to the presented question $\mathcal{M}$ is replaced by $\{M \cup \{s\} \mid M \in \mathcal{M} \wedge M \cup \{s\} \in \mathcal{S}\}$ and the third step of the TEST mode checks if $\mathcal{M} = \{S\}$, indicating that the student masters all skills required in the domain. If this is the case the system changes to the END mode, if not the system changes to the third step of the DIAGNOSIS mode.

The END mode informs the student that he or she had reached the learning goal and finishes the process.

The teaching process is depicted in Fig. 1 as a flow chart.

GENERALIZATIONS OF THE BASIC SYSTEM

In the previous section, we introduced an architecture for an ITS based on the theory of knowledge spaces. Because our main goal in this section is to work out the general ideas underlying an application of knowledge space theory in the field

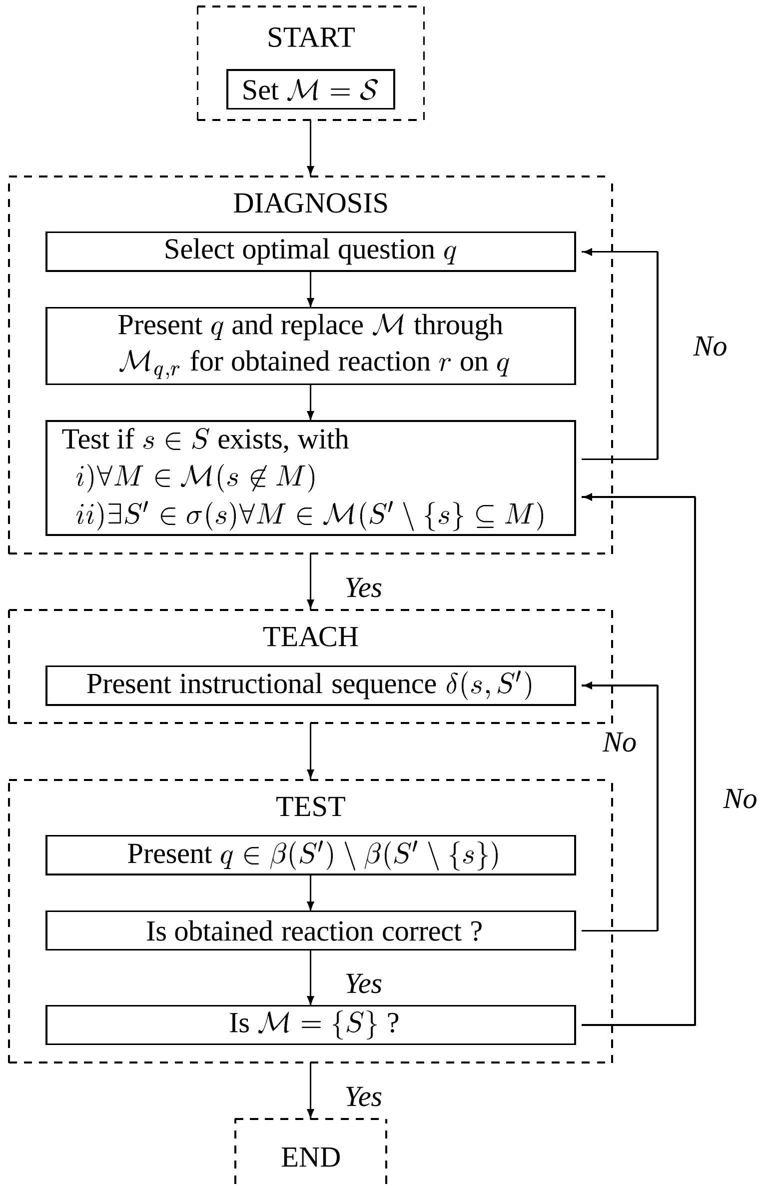

FIG. 1. Teaching process described as an interaction between the five modes of the system.

of computerized teaching, this basic architecture is very simple and is therefore restricted in its applicability. In this section, we try to clarify the limitations of the basic architecture and work out methods to overcome them. So, at the end of this section, we are able to formulate a generalization of our basic architecture.

In our basic architecture, we have considered only two possible answer types of a student to a presented question by assuming that such an answer is either correct or incorrect. This approach has the advantage of simplicity, but does not use the whole information about the student's competency contained in the obtained answer.

To use this information completely we have to introduce the answer in the skill assignment and clarify the role of the answer in the update of the set of hypothetical student states $\mathcal{M}$ within the first and second step of the DIAGNOSIS mode.

Therefore, we introduce the set R of all possible answers (possible inputs to the system) a student can give to the questions in Q. We relate the skills in S to the questions in Q and possible answers in R by a *skill assignment*[2] $\alpha : Q \times R \to \text{Pow}(S)$. The interpretation of $\alpha(q, r) = \{S_1, \ldots, S_n\}$ is "A student who answers r to question q must at least master all skills from one of the sets $S_1, \ldots, S_n$". As in the previous section, we can assume that the sets $S_1, \ldots, S_n$ in $\alpha(q, r)$ are minimal with respect to $\subseteq$, i.e. $\forall S_i, S_j \in \alpha(q, r)\ (S_i \not\subseteq S_j)$.

As in the previous section, the skill assignment α determines for each state $S' \in \mathcal{S}$ the answer behavior $\beta(S')$ consistent with this state by

$$\beta(S') = \{(q, r) \mid q \in Q \wedge S' \in \alpha(q, r)\},$$

$\beta(S')$ is the set of all tuples of questions and obtained answers that can be produced by a student with competency S'.

Now, we have to generalize the half-split procedure in the first step of the DIAGNOSIS mode. For the set $\mathcal{M}$ of hypothetical student states and a question $q \in Q$, we define $\mathcal{M}_{q,r} \subset \mathcal{M}$ as the set of all elements of $\mathcal{M}$, that are consistent with the fact that the student gives answer r on question q, $\mathcal{M}_{q,r} = \{M \in \mathcal{M} \mid (q, r) \in \beta(M)\}$. We define a question $q' \in Q$ as being *optimal for diagnosis* if the value $\mid \max_{r,r' \in R}$ (card$(\mathcal{M}_{q',r})$ – card$(\mathcal{M}_{q',r'})) \mid$ is minimal over all $q \in Q$. This means that for the optimal question q' the cardinality of the sets $\mathcal{M}_{q',r}$ is for all possible answers r optimal balanced, that is the sets $\mathcal{M}_{q,r}$ differ in cardinality as little as possible. The sets $\mathcal{M}_{q',r}$ must not be pairwise disjoint, we can have $\mathcal{M}_{q',r} \cap \mathcal{M}_{q',r'} \neq \emptyset$ for $r \neq r'$, and they can be empty, that is we can have $\mathcal{M}_{q',r} = \emptyset$ for some r.

As before in the second step of the DIAGNOSIS mode, an optimal question is presented to the student and if the answer is $r \in R$, then $\mathcal{M}$ is replaced by $\mathcal{M}_{q',r}$.

[2] In the following we denote concepts that are proper generalizations of concepts introduced earlier by the same symbols as in the previous section. Since, for example, the generalized concept of a skill assignment is for $R = \{\text{correct}, \text{incorrect}\}$ identical with the concept of a skill assignment as introduced in the previous section, it is denoted by the same greek letter α.

To make things easy, in the formulation of the basic architecture we have drawn implicitly the unrealistic assumption that a student answers always accordingly to her or his competency described by a state of the student model. This assumption ensures that our rules governing the update of $\mathcal{M}$ in the DIAGNOSIS mode do not end with an empty $\mathcal{M}$. Remember that the teaching process starts with $\mathcal{M} = \mathcal{S}$ and that after each obtained answer r of a student to a question q, the set $\mathcal{M}$ is replaced by $\mathcal{M}_{q,r} \subseteq \mathcal{M}$.

But if, as a result of a lucky guess or a careless error, the students answer r to question q is not consistent with one of the states in $\mathcal{M}$, that is with the students previous interactions with the system, we can receive an empty $\mathcal{M}_{q,r}$. In this case, we have to find a new rule for the update of $\mathcal{M}$ in step two of the DIAGNOSIS mode.

We distinguish between two cases. If $\mathcal{M}_{q,r} \neq \emptyset$ we replace just as before $\mathcal{M}$ by $\mathcal{M}_{q,r}$. If conversely $\mathcal{M}_{q,r} = \emptyset$ we replace $\mathcal{M}$ by $\mathcal{M} \cup \{S' \in \mathcal{S} \mid \exists M \in \mathcal{M} \ (\mid (M \setminus S') \cup (S' \setminus M) \mid = 1)\}$, enlarging $\mathcal{M}$ by considering all states of the student model that differ only in one skill from the states in $\mathcal{M}$. The enlargement of $\mathcal{M}$ is necessary because we have to consider the possibility that the obtained answer r on q is in accordance with the competency of the student and $\mathcal{M}_{q,r}$ is empty as a result of a lucky guess or careless error to one of the previous questions, that leaded to an elimination of the correct state of the student.

To ensure that a wrong assumption concerning the students' competency, for example, resulting from a careless error, can be corrected, we have to assume that $\mathcal{S}$ is *well–graded* (Falmagne & Doignon, 1988). This means that for two arbitrary states $S', S'' \in \mathcal{S}$, there exists a chain $S' = S_1 \subset S_2 \subset \ldots \subset S_n = S''$ of states in $\mathcal{S}$ with card$(S_i \setminus S_{i-1}) = 1$ for $i = 2, \ldots, n$. Each S_i contains exactly one skill more than his predecessor in the chain.

This assumption is necessary because we enlarge $\mathcal{M}$ in the case of an error in diagnosis only by the states that differ in only one element from the states in $\mathcal{M}$. If in a not well–graded $\mathcal{S}$ one state S' differs in at least two skills from all other states, this state can not be reached by the described rule if it is once eliminated from $\mathcal{M}$, for example as a result of a careless error.

One important restriction of the basic architecture is that, given a skill s not mastered by a student who masters all skills from a set $S' \setminus \{s\}$, where S' is a set of prerequisites of s, there is only one sequence $\delta(s, S') = (i_1, \ldots, i_n)$ of instructions that lead to the mastery of s by the student. Generally one would expect many such instructional sequences having all the same intended effect on the students' knowledge. For example, a missing skill may be taught alternatively by presenting some examples, some instructional texts, or a mixture of both. Which of these alternatives is best for a particular student may depend on the student's personality.

We include the possibility of alternative instructional sequences by assigning to a skill s and a set S' of prerequisites of s a set of instructional sequences through a function δ. So we regard δ as a function $\delta : S \times \text{Pow}(S) \rightarrow \text{Pow}(\text{Seq}(I))$. The interpretation of $\delta(s, S') = \{I_1, \ldots, I_n\}$ is "To a student mastering all skills from S' and not mastering s, one of the instructional sequences $I_1, \ldots, I_n$ should be presented to reach mastery of s".

Because we had assumed in our basic architecture that $\delta(s, S')$ contains only one instructional sequence, we must assume that this instructional sequence is repeated until the student masters skill s. If the TEST mode finds that a student has not learned the last skill taught by the TEACH mode by presenting the sequence $\delta(s, S')$, this sequence is presented again. This may be problematic for two reasons. First, the instructional sequence $\delta(s, S')$ may be inappropriate for the student, so that a repetition of this sequence makes no sense. Second, a repetition of informations already presented may lead to a motivational decrease.

We can overcome this limitation by assuming that solely such instructional sequences can be presented to a student that have not been presented before as long as $\delta(s, S')$ contains at least one informational sequence not already presented. Formally, this means that to a student, who had already received the instructional sequences $I_1, \ldots, I_n$ (these sequences are stored in the history of the teaching process), only instructional sequences from $\delta(s, S') \setminus \{I_1, \ldots, I_n\}$ can be presented if this set is not empty. If $\delta(s, S') \setminus \{I_1, \ldots, I_n\}$ is empty, then an arbitrary element of $\delta(s, S')$ determined by chance should be presented again to the student.

Now the problem arises which of these sequences $I_1, \ldots, I_n \in \delta(s, S')$ should be presented to a student during the TEACH mode of the system. This can be solved by a *choice rule* γ which chooses one of these sequences with respect to the previous interactions between system and student.

Such a choice rule can be implemented in several ways. In the following, we discuss one such possibility. We divide the set $\text{Seq}(I)$ of instructional sequences into a finite number of pairwise disjoint subsets $\mathcal{C}_1, \ldots, \mathcal{C}_n$, that is we have $\text{Seq}(I) = \mathcal{C}_1 \cup \ldots \cup \mathcal{C}_n$ and $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset$ for $i \neq j \in \{1, \ldots, n\}$.

These subsets $\mathcal{C}_i$ represent instructional types, for example instruction by examples, or instruction based on texts. For the classification of sequences into types, the description of I as a union of pairwise disjoint sets of different instructions introduced in the previous section can be used.

We can expect that students differ in the type of instruction they prefer and with which they learn best. So the choice rule γ should ensure that a student receives the type of instruction which is most effective for her or him, that is γ should be adaptive to the success of a special instructional type in the previous interactions between system and student.

We define the *effectivity* $\epsilon(\mathcal{C}_i)$ of an instructional type $\mathcal{C}_i$ as the relative frequency of already presented instructional sequences from $\mathcal{C}_i$ leading to a mastery of the intended skill. The effectivity $\epsilon(\mathcal{C}_i)$ can be measured by the TEST mode. So the effectivity ϵ is a function $\epsilon : \{\mathcal{C}_1, \ldots, \mathcal{C}_n\} \times H \to [0, 1]$.

We use the effectivity for the construction of an adaptive choice rule by assuming that the probability $p(\mathcal{C}_i)$ for the presentation of a sequence $I_i \in \delta(s, S')$

increases with the effectivity $\epsilon(\mathcal{C}_j)$ of the instructional type to which I_i belongs. If $I_1 \in \mathcal{C}_1, I_2 \in \mathcal{C}_2 \in \delta(s, S')$ we should have $p(I_1) \geq p(I_2)$ if and only if $\epsilon(\mathcal{C}_1) \geq \epsilon(\mathcal{C}_2)$. The decision between members of the same instructional type is drawn by chance as before.

Another limitation of the basic architecture is that the goal of the teaching process is mastery of all skills in S by the student, the teaching process terminates only if $\mathcal{M} = \{S\}$. In a practical application it may be for two reasons necessary to define subgoals of the teaching process.

First, for some students the mastery of a subset of S may be sufficient, for example, if they need only basic knowledge in the domain and do not want to become experts there. Second, the teaching process may have taken too much time for one session, so we have to formulate points where an interuption makes sense.

Such subgoals can be introduced by marking a subset $\mathcal{G}$ of $\mathcal{S}$. An element of $\mathcal{G}$ is called a *subgoal* of the teaching process.

Which elements of $\mathcal{S}$ are suitable as subgoals? A subgoal should be a closed piece of knowledge, all skills in it should be connected. If we interpret the surmise function σ on S as a description of the contentual dependency of the skills the optimal candidates for subgoals are the basis elements of $\mathcal{S}$. The reason is that a basis element B includes with each skill s only such skills, which are either elements of a set of prerequisites of s or for which s itself is an element of one of their sets of prerequisites. Therefore we chose $\mathcal{G}$ as a subset of the basis $\mathcal{B}$ of $\mathcal{S}$.

To make use of the subgoals for the control of the teaching process, we only have to change the third step of the test mode of our system. This step tests if $\mathcal{M} = \{S\}$, if the student masters all skills from S, and changes to the END mode if this is the case. This step can be easily generalized by assuming that it tests if all hypothetical student states include a basis element B, that is if the student masters at least all skills from B. Formally, this means that it is tested if the condition $\exists B \in \mathcal{B} \forall M \in \mathcal{M}\ (B \subseteq M)$ is fulfilled. If this is the case the student is asked if he or she wants to interrupt the process. If the student decides to interrupt, the history H of the teaching process is stored and can be used in the next session to continue the process properly. If the learning goal of a particular student is a proper subset of S, for example, if the student only needs basic knowledge in the domain, the process can be terminated if this special goal is fulfilled. In this case, we have also to ensure that skills not contained in the learning goal $G \subset S$ of the student are not taught, even if the diagnostic procedure finds out that the student does not master these skills. This can be done easily by replacing S by G in all components of the system and in the teaching process. This means that the system "forget" all information about skills in $S \setminus G$ and reacts just as if the set of skills necessary in the domain would be G.

The learning goal of a particular student may be given explicitly, for example, if the system is used by students to fulfill a special course requirement, or asked interactively at the beginning of the teaching process from the student.

The set $\mathcal{G}$ of subgoals of the teaching process can also be used to solve a problem already mentioned in the previous section.

The problem arises if the DIAGNOSIS mode has detected at least two skills s_1 and s_2 not mastered by a student and sets of prerequisites S_1 and S_2 for s_1 respectively s_2 for which all skills from $S_1 \setminus \{s_1\}$ and $S_2 \setminus \{s_2\}$ are mastered by that student. The system has then to decide which of these skills should be taught first. In the previous section, we had assumed that this decision is drawn by chance.

Given a set $\mathcal{G}$ of subgoals we can improve the decision strategy by assuming that skills are taught first that ensure that a subgoal is reached. Only if this rule is not sufficient to decide between two skills, for example if none of these skills leads to a subgoal, the decision is drawn by chance by an implemented random decision strategy. This procedure ensures that the teaching strategy tries to teach first closed pieces of knowledge, i.e., subsets of S in which all skills are contentual connected by σ, instead of unconnected pieces of knowledge.

Another possible extension of our basic architecture is the introduction of the learning speed or learning ability of students into the teaching strategy. Students clearly differ in their ability to learn new material. This difference between students should be considered in an ITS to ensure that the strategy the system uses for the presentation of instructions reflects the learning ability of a student working with the system.

For example, for a student who is able to learn fast it may be optimal to present a number of instructional sequences during one step of the teaching process in order to allow the student to aquire a number of skills during that step. For a student who learns slowly, it seems more appropriate to present only the instructions necessary to learn one missing skill during one step of the teaching process.

The teaching process described in the previous section considers only the second case of the example. This can be seen from the interaction of the TEACH and the TEST mode of the system. Assume a student who does not master skill s but who masters all skills from $S' \setminus \{s\}$, where S' is a set of prerequisites of s. We assumed in the previous section that in such a case, the TEACH mode presents the instructional sequence $\delta(s, S')$ and afterwards the system changes to the TEST mode to ensure that skill s was aquired by the student. Therefore, the procedure described in the previous section seems to be optimal only for students who learn slowly.

Because the learning speed of a student is not known to the ITS before the teaching process starts, it must be measured during this process. This can be done adaptively by the TEST mode. The central idea is that the number of cases in which the student had aquired the skill taught in the previous TEACH step is an indicator for the learning speed of that student.

We measure the learning speed g of a particular student by an integer greater than 0, that is we can have $g = 1, 2, 3, \ldots$. At the beginning of the teaching process we set $g = 1$ for every student. If the TEST mode detects in l successive steps, where l is a fixed level, that all skills[3] taught in the previous step are aquired by the student, then g is replaced by $g + 1$. If the TEST mode detects conversly that a student has not aquired the skills taught in the previous TEACH step and g is greater than 1, then g is replaced by $g - 1$.

Now, we have to describe how the learning speed g influences the choice of skills that will be taught during the TEACH mode. Assume that the DIAGNOSIS mode had detected a skill s not mastered by a student who masters all skills from $S' \setminus \{s\}$, where S' is a set of prerequisites of s. If $g = 1$, then the TEACH mode presents just as before an element $\delta(s, S')$ and changes to the TEST mode. If $g = m > 1$ then the TEACH mode checks if there exists skills $s = s_1, s_2, \ldots, s_m$ with $S' \cup \{s_1, \ldots, s_p\} \in \sigma(s_{p+1})$ for $p = 1, \ldots, m - 1$. This means that $S' \cup \{s_1, \ldots, s_p\}$ is a set of prerequisites of the skill s_{p+1}. If such skills exists, then the system presents instructional sequences from $\delta(s_1, S'), \delta(s_2, S' \cup \{s_1\}), \ldots, \delta(s_m, S' \cup \{s_1, \ldots, s_{m-1}\})$, i.e. gives the student the possibility to aquire all the skills $s_1, \ldots, s_m$ in one step. If such skills does not exist, the system choses the maximal number $p < m$ of skills with the described properties and presents instructional sequences corresponding to these skills. We have to ensure in this case that an aquisition of this $p < m$ skills by the student does not lead to an increase of the value g. So only such cases are considered in the update of g, where the number of skills presented during the TEACH step is equal to the actual value of the learning speed g.

DISCUSSION

The ITS described in this article consists of several structures, for example S, Q, I, σ, α, δ, which interact during the teaching process. The design of these structures influences the performance and effectivity of the system. We illustrate this by an example.

The effectivity of the system depends on its ability to diagnose which skills a student does not master. Especially the speed of the diagnosis is very important, since it may be demotivating for a student if a lot of questions will be presented to her or him until a lacking skill is diagnosed and teaching materials are presented. The speed of the diagnosis depends on the functions σ and α. For example a stricter σ results in a faster diagnosis.

Thus, it may be adequate to chose σ as strict as possible, even if this may result in an elimination of some true states from $\mathcal{S}$ which occur with low frequency in the intended population. This approach increases the speed of the diagnosis while it decreases its accuracy. Such manipulations of the components must be handled carefully. Their efficiency can be evaluated only in a concrete application.

[3] We assume here that the TEACH mode can teach more than one skill in each step. How this can be realized is described later.

The described ITS contains no procedure for the repetition of already mastered skills. We have not introduced such a procedure since we assumed that the skills are teached in increasing sequence concerning their difficulty[4]. Therefore we can assume that the skills which are teached first, i.e. the easy ones, are practiced again during the aquisition of the skills which require them as prerequisites. But if we assume that the teaching process may be interupted, for example after a subgoal is reached, for a long period of time it seems necessary to introduce a procedure for the repetition of skills. Such a procedure requires a rule for selecting the skills mastered in the previous session which should be repeated at the beginning of the new session. How such a rule can be formulated and integrated into the teaching process is at the moment an open question.

We have formulated our ITS domain independent. For a concrete application of the ITS in a special knowledge domain several steps are necessary. First, the skills relevant in the domain, i.e. the set S, must be found and the dependencies between these skills, i.e. the surmise function σ, must be formulated. Second, questions must be constructed which can be used for diagnosis of the skills possessed by students. Then their connection to the skills, i.e. the skill assignment α, must be formulated. Third teaching materials, i.e. the set I, must be developed and it must be stated which instructions should be presented to a student in a special state of the student model by constructing the function δ.

These steps should be left to experienced teachers in the domain. But even if we assume that a group of such experienced teachers enforced such a necessary analysis of the domain we have to be aware of the risk that some of the structures σ, α, δ may contain errors which may influence the performance of the system negatively.

For example an indadequate construction of σ may lead to a student model $\mathcal{S}$ containing many states which will result in a very slow knowledge diagnosis.

This risk can be reduced with two different approaches. One approach is to test the assumptions contained in σ, α, δ empirically. How this can be done is for example described in Albert and Held (1994), Albert, Schrepp and Held (1994), Schrepp (1993), Held (1993), or Korossy (1993, 1997). But if the number of skills in S or the number of questions in Q is high this method may be problematic, since to many empirical data will be necessary to draw conclusions on the correctness of σ, α and δ.

A second approach to reduce the influence of an incorrect formulation of the basic structures is to make the functions σ, α, δ adaptive. We will illustrate this idea by an example. Assume that we have $I_j \in \delta(s, S')$, i.e. the experts assumed that I_j should be presented to a student mastering all skills from $S' \setminus \{s\}$ and not mastering s. Assume further that the instructional sequence I_j has not the intended effect, i.e. only a few students show mastery of s after I_j was presented to them. Remember that this can be recognized by the TEST mode. We can made

[4] This is realized within the system since the teaching procedure makes use of the surmise function σ to chose the skills which should be teached next.

δ easily adaptive if we assume that I_j is eliminated if less than a percentage p of students reaches mastery of s after I_j is presented to them. We have to ensure here that $\delta(s, S')$ will not become empty. Another possibility is to chose the instructional sequences accordingly to the probability of their success.

This example shows that it is relatively simple to make the system adaptive concerning δ, since we have a possibility to measure the success of an instructional sequence directly. For σ and α this is more complicated, because the correctness of these functions can not be observed directly. If, for example, a hypothetical skill state $S' \in \mathcal{S}$ is never observed by any student this may be due to the fact that $\mathcal{S}$ contains to much states, i.e. σ is formulated not strictly enough, or due to the fact that Q is not constructed adequately, i.e. there is no combination of solved and unsolved questions from Q which implies state S'. It is at the moment not clear how σ and α can be changed adaptively due to their success to explain the behavior of students.

We have formulated in this paper the structure of an ITS based on the theory of knowledge spaces and skill assignments. The next step in the development of our work should be the implementation of the ITS and its application in a special knowledge domain. Since we have described the components of our system as well as their interaction strictly formalized, the implementation of the system in a language of logical programming such as Prolog or Lisp should cause no problems.

A concrete application of this ITS can be used to evaluate the systems efficiency and can give raise to new insights in the dynamic of the teaching process, which may lead to further improvement of the theoretical structures.

ACKNOWLEDGEMENTS

The research reported in this paper was supported by Grant Lu 385/1 of the Deutsche Forschungsgemeinschaft to J. Lukas and D. Albert at the University of Heidelberg.

REFERENCES

Albert, D., & Held, T. (1994). Establishing knowledge spaces by systematical problem construction. In D. Albert (Ed.), *Knowledge structures* (pp. 81–115). Berlin, Heidelberg: Springer.

Albert, D., Schrepp, M., & Held, T. (1994). Construction of Knowledge Spaces for Problem Solving in Chess. In: G. Fischer & D. Laming (Eds.), *Contributions to Mathematical Psychology, Psychometrics, and Methodology.* New York: Springer.

Brown, J.S., & Burton, R.R. (1978). *Diagnostic models for procedural bugs in basic mathematical skills. Cognitive Science, 2*, 155-192.

Doignon, J.-P. (1994a). Knowledge spaces and skill assignments. In G. Fischer & D. Laming (Eds.), *Contributions to mathematical psychology, psychometrics, and methodology* (pp 111 -121). New York: Springer.

Doignon, J.-P. (1994b). Probabilistic assessment of knowledge. In D. Albert (Ed.), *Knowledge structures* (pp. 1–57). Berlin: Springer.

Doignon, J.-P., & Falmagne, J.-C. (1985). Spaces for the assessment of knowledge. *International Journal of Man-Machine Studies*, *23*, 175–196.

Doignon, J.-P., & Falmagne, J.-C. (1998). *Knowledge spaces.* Berlin: Springer.

Falmagne, J.-C., & Doignon, J.-P. (1988). A markovian procedure for assessing the state of a system. *Journal of Mathematical Psychology*, *32*(3), 232–258.

Held, T. (1993). *Establishment and empirical validation of problem structures based on domain specific skills and textual properties. A contribution to the theory of knowledge spaces.* Unpublished PhD thesis, University of Heidelberg.

Heines, J.M., & O'Shea, T. (1985). The design of a rule based CAI tutorial. *International Journal of Man–Machine Studies*, *23*, 1–25.

Kimball, R. (1982). A self-improving tutor for symbolic integration. In: D. Sleeman & J.S. Brown (Eds.), *Intelligent tutoring systems*. London: Academic Press.

Kondo, H., Watanabe, K., Takeuchi, A., & Otsuki, S. (1990). ITS: Recognizing the context and guiding reduction processes for fraction calculation. *Proceedings of the International Conference on ARCE*, (pp. 31–37).

Korossy, K. (1993). *Modellierung von Wissen als Kompetenz und Performanz.* [Modeling of knowledge as competence and performance.] PhD thesis, University of Heidelberg.

Korossy, K. (1997). Extending the theory of knowledge spaces: A competence-performance approach. *Zeitschrift fr Psychologie*, *205*, 53–82.

Kunichika, H., Takeuchi, A., & Otsuki, S. (1994). Hypermedia English learning environment based on language understanding and error origin identification. *IEICE Transactions on Information and Systems*, *E77-D (1)*, 89–97.

Lukas, J., & Albert, D. (1993). *Knowledge assessment based on skill assignment and psychological task analysis.* In G. Strube & K.F. Wender (Eds.), The cognitive psychology of knowledge: The German *Wissenspsychologie* project. Amsterdam: Elsevier (North–Holland).

Mandl, H., & Hron, A. (1985). *Förderung kognitiver Fähigkeiten und des Wissenserwerbs durch computerunterstütztes Lernen.* [Support of cognitive abilities and knowledge aquisition through computer–aided learning.] Report, Deutsches Institut für Fernstudien an der Universität Tübingen.

Matsuda, N., & Okamoto, T. (1992). Student model and its recognition by hypothesis-based reasoning in ITS. *IEICE Transactions on Information and Systems*, *J75-A*, 266-274.

Okamoto T., & Matsuda N. (1993). *Overview on the studies of intelligent CAIs/ITSs in Japan.* Unpublished report, Univ. of Electro–Communications Tokyo / Kanazawa Institute of Technology.

Ploetzner, R., & Spada, H. (1993). *Multiple mental representations of informations in physics problem solving.* In G. Strube & K.F. Wender (Eds.), The cognitive psychology of knowledge: The German *Wissenspsychologie* project. Amsterdam: Elsevier (North–Holland).

Schrepp, M. (1993). *Über die Beziehung zwischen kognitiven Prozessen und Wissensräumen beim Problemlösen.* [On the relation between cognitive processes and

knowledge spaces in problem solving.] Doctoral dissertation, University of Heidelberg.
Sleeman, D., & Brown, J.S. (1982). *Introduction: Intelligent tutoring systems.* In: D. Sleeman & J.S. Brown (Eds.), Intelligent tutoring systems. London : Academic press.
Takeuchi, A., & Otsuki, S. (1994). An interactive learning environment for an intelligent tutoring system. *IEICE Transactions on Information and Systems*, *E77-D (1)*, 129–137.
Yasuda, K., & Okamoto, T. (1991). *The study of ITS to diagnose the algorithm of the novice's program.* ICOMMET'91. (Int. Conf. on Multi–Media in Education and Training)