A Formalization of Student Modeling

Frédéric Danna and Pascale Sébillot
Irisa – Campus de Beaulieu – F-35042 Rennes Cedex – France
E-mail: danna@irisa.fr, sebillot@irisa.fr

Abstract

In this paper, we focus on student modeling within an Intelligent Tutoring System (ITS). Elaborating a student modeling system implies to determine the formalism in which the student knowledge will be represented, and the processes that will dynamically acquire and synthesize this knowledge. We describe three domain-independent properties that this formalism and these processes must possess to build sound and accurate student models. First, because the student's knowledge evolves in time, the modeling system must be able to represent knowledge that issues from *defeasible reasoning*. Second, since students may have contradictions in mind, it also must deal with *paraconsistent reasoning*. Third, because the diagnosis of the student's cognitive state is not certain, the results of the cognitive diagnosis must be considered as hypotheses rather than certain facts. These hypotheses may have to be withdrawn if contradictory information is subsequently acquired. Therefore, the system must also be able to follow *hypothetical reasoning*. We show how an implementation based upon probabilistic logic can take into account both student's defeasible and paraconsistent reasonings. We also point out where, when and how hypothetical reasoning mechanisms must intervene. We exemplify our results within the framework of Compounds, an ITS about English compounding processes.

1 Introduction

The wide availability of computers has initiated an attempt to use computers in order to assist teachers in their instructional purpose [Wen87]. Computer-Aided Instruction (CAI) softwares have then been created. Such systems may

possess sets of exercises, examples, counter-examples, explanations, etc, just like instructional books do. But these sets are predefined ones, since they are elaborated during the design phase of the CAI system. Every possible future interaction with the student has to be foreseen and hand-coded by the designers of the system. For example, every possible answers for each question of each exercise have to be anticipated. Moreover, for the system to know which action it must do next, each possible answer has to be associated with either another question, or another exercise, or counter-example, etc. These associations encode decisions that arise from the application of pedagogical knowledge. In order to adapt the interaction to the student (hence the term *adaptive CAI*), some systems make these associations depend on the rate of correct *vs* incorrect answers. But these associations do not depend on the reasons that underly the student's answers. So, the student is only taken into account through parametric summaries of his/her behavior: there is no explicit representation of the student's knowledge [SB82, DS91]. There is also no explicit representation of the target domain knowledge. Thus, these systems are not real experts of the domain. For example, the student may want to know what the correct answer for a given question is. If this answer has not been hand-coded, the system is not able to respond to the student's query. These limitations come from the fact that these systems rely on the encoding of decisions that arise from some knowledge, rather than on the encoding of this knowledge itself. This difference concerns every kind of knowledge that is possessed by the system; namely, the knowledge that has to be communicated, the student's knowledge, the pedagogical knowledge, etc.

More "intelligent" softwares (ICAI, or ITS for Intelligent Tutoring Systems) have then been required. These systems are supposed to use artificial intelligence technics in order to explicitly encode the different kinds of knowledge that a tutoring system should have [Wen87]. These knowledge bases are used by the systems themselves in order to manage the interaction. So doing, they are expected to behave much more like human tutors. Namely, such tutors are able to optimize the learning situation. This optimization is mainly brought through taking the student into account: they observe the student's behavior, and infer the student's knowledge from these observations. They can then use this information in order to dynamically produce individualized exercises, examples, counter-examples, explanations, or to decide when unsolicited advice should be offered [BC88], etc. This ability allows these tutoring systems, among others, to implement complex pedagogical strategies.

For instance, they can manage socratic dialogues with the student (when a tutor diagnoses that the student has some misconceptions, it dynamically generates exercises containing counter-examples that will lead the student to correct himself/herself). In addition, if the tutoring system needs some precise information about the student's level, in order to decide which piece of knowledge should be presented in the next interactions, it can produce exercises that will help in getting this information.

Such ITS are usually constituted of four knowledge bases [Wen87]. The *expert model* contains the knowledge that has to be communicated to the student. The *interface* of the ITS uses knowledge about communication in order to manage the dialogue between the student and the system. The *pedagogical module* of the ITS chooses the pedagogical strategy to apply in order to optimize the learning situation. This choice is based upon general information about the student as well as his/her knowledge, which is stored in the *student model*.

Our current works focus on the automatic elaboration of such a student model. Since the student's cognitive state is not known during the ITS design phase, the elaboration of the student model is obviously a dynamic process [PSH94]. This elaboration is performed by a *diagnosis module* that infers the student's knowledge from his/her behavior. Because this information is mainly intended to be used by the pedagogical module in order to individualize the learning situation, the contents of the student model greatly depend on the needs of the pedagogical module [Sel90]. Up to now, we only consider information about the student's cognitive state, and not other kinds of general information like psychological attributes. This restriction is due to the fact that such attributes are not only difficult to determine, but also to manage and to use. As far as the student's knowledge is concerned, the student model should enable five pedagogical functions [Sel87]. First, it should enable the system to *eradicate* the student's misconceptions. This goal can be reached through describing to the student what his/her error is, or through showing to him a brief summary or a part of the correct knowledge, or through giving to him the correct answer or a counter-example, etc. This implies that the student model depicts the possible misconceptions of the student. Second, the student model may be used in an *elaborative* purpose. This functionality concerns the extension of the student's knowledge. In order for the tutoring system to be able to know what topic to teach next, the student model must represent such information. Third, the student model may be used as data for a *strategic* function, which helps to dynamically change the strategy

underlying the instructional plan (for example, shifting from a detailed examination of a single topic to a shallow examination of a number of topics when the current topic is complete but the ITS has little confidence in its assessment of the student's knowledge). So, the student model must describe to which extent the student masters his/her knowledge. Fourth, the student model can be used in order to perform *diagnoses*. This diagnostic function is related to the withdrawing of ambiguities. Some of these ambiguities concern the student's knowledge state. They arise from the possible lack of information which prevents from accurately stating the student's knowledge. Other ambiguities concern the student model. These ones come from the fact that the diagnosis can be a non-deterministic process. This diagnostic function implies that the student model must be able to represent such ambiguous information about the student. Finally, the student model can also be used in order to *evaluate* what the student knows. This description can be shown to the student, a course administrator, etc. So, the student model must contain sound and accurate information about the student's knowledge. In conclusion, in order to enable these pedagogical purposes, a student model must describe what the student knows, to which level he knows this, what he does not know, and the possible student's misconceptions. Moreover, the student modeling system must be able to tackle the ambiguity in the student knowledge and the student model. The point now is to determine an automatic system that builds models containing such information. This student modeling system should be as domain-independent as possible, in order to be convenient for different domains of application.

Our framework for student modeling can be exemplified within COMPOUNDS, an ITS about the English compounding processes. This domain has been chosen because French speakers show difficulties in producing and understanding English compounds [Bou92, Dal93]. These linguistic deficiencies have particularly important drawbacks in scientific and technical domains, where the rate of compound words often exceeds 50% [Wag91]. This rate is due to the high productivity of the English compounding processes. For example, in the computer science field, this productivity has soon led to the creation of compound words like "data-processing" or "computer-language". English speakers are able to retrieve the meaning of such words, even if they have never encountered them before. So, the English compounding processes that the system should taught to the French speaking students are those which are both used to produce compounds and to understand them, that is to translate "data-processing" by "the act of processing data" and, con-

versely, to know that the definition "the act of processing data" corresponds to the compound "data-processing".

The expert model of the ITS chosen as example is first briefly presented in section 2. The remaining sections focus on the student modeling problem. The issues raised do not depend on the domain. General properties that a student modeling system must have in order to build sound and accurate student models are described, both from an empirical point of view in section 3.1 and from a theoretical standpoint in section 3.2. An implementation that respects most of these properties is then detailed in section 3.3.

2 The expert model

In COMPOUNDS, the expert knowledge corresponds to the English compounding processes. Up to now, only a subset, although very productive, of the whole linguistic knowledge is covered. As far as this expert model is concerned, a *compound word* is a noun (N) or a verb (V) made up of two constituents: N-N[1], N-V[2], V<suffix>-N[3], or N-V<suffix>[4], where <suffix> can be either *er* or *ing*. Moreover, the compounds that are treated here are right-headed ones, i.e. the ones for which the meaning is mainly given by the rightmost constituent. This limitation is not too restrictive since this is usually the case in English. Finally, we also treat non-lexicalized compounds. The corresponding compounding processes enable the expert model to behave like a human expert. First, they permit to understand any compound, i.e. to retrieve the definition of a given compound. Conversely, these processes can be used in order to produce the compound that corresponds to a given definition.

These compounding processes are directed by *concepts* that issue from linguistic theories [Lie83, Dow77, Sel82]. For example[5], a concept, noted `right-headed` hereafter, states that the meaning of a compound is generally directed by its rightmost element, called the *head* of the compound ("boy" in "stable-boy"). Another concept, noted `ISL` hereafter, for Implicit Semantic

[1] "woman-doctor", "trumpet-plant", "stable-boy"

[2] "spoon-feed", "hand-weave"

[3] "drinking-water", "killer-shark"

[4] "data-processing", "truck-driver"

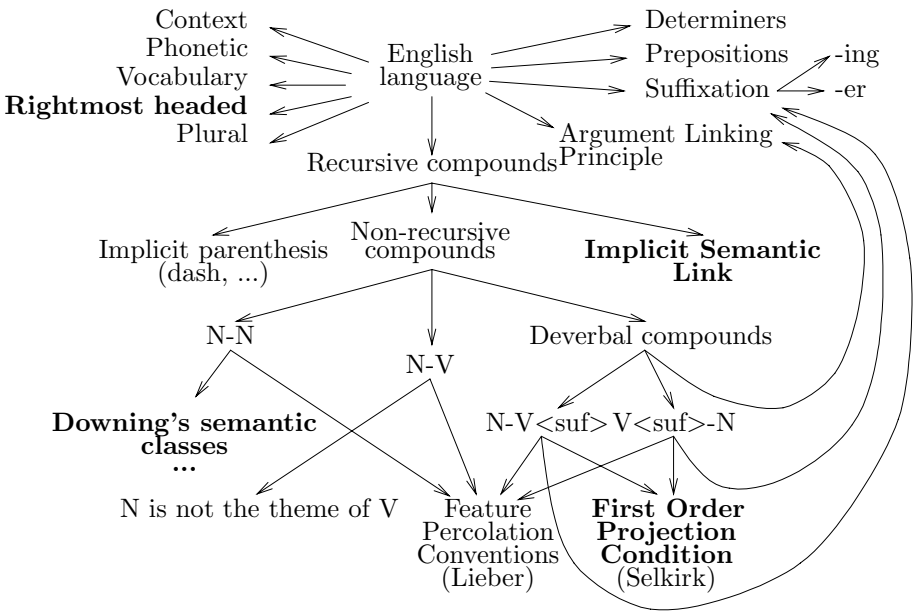
[5] We only present here the concepts that are needed for subsequent explanations. See [BDS93] for a more detailed presentation.

Link, states that an underlying relation exists between the two constituents of a compound. This semantic relation corresponds to the predicate that is used by an English speaker in order to link the two entities of the compound in the definition that he/she gives for it (e.g. "to look like" for "bull-dog": "a dog that looks like a bull"). This predicate has thus to be retrieved in order to give the definition of the compound. In the case of "N-N" compound, this retrieval is based upon P. Downing's works [Dow77], which state that this link depends on the semantic class of the head of the compound. For example, if the head denotes an animal (e.g. "cat-fish"), they chiefly predict a resemblance link between the two entities of the compound. This concept is noted `Downing/animal` hereafter. The combination of the `right-headed`, `ISL`, and `Downing/animal` concepts permits to automatically obtain the semantics of "cat-fish", i.e. "a fish that looks like a cat". The First Order Projection Condition [Sel82], noted `FOPC` hereafter, is a concept which states that the verb or suffixed verb used in a compound possesses arguments that must be satisfied within the compound. For example, the verb "drive" requires two arguments, that play the roles of agent and theme. When this verb is affixed by the agentive suffix *er* in order to form the deverbal constituent "driver", "driver" satisfies the agentive argument of "drive". When the deverbal element is used to form the compound "truck-driver", "truck" must be interpreted as the theme of "drive". Thus, both arguments of "drive" are satisfied within the whole compound. So doing, the `right-headed` and `FOPC` concepts together allow to judge this compound as correct, and to give its definition: "someone who drives trucks".

Some of these concepts apply to every kind of compounds. This is the case for the `right-headed` concept, for example. Others are specific to some types of compounds. This is so for the `FOPC` concept, which is specific to the *verbal* or *deverbal compounds*[6]. So, these concepts can be organized into a graph that makes explicit their high or low degree of specificity (cf. figure 1). This graph also represents one possible partial order of the acquisition of the concepts. For example, the arrow that links *English language* to *Vocabulary* means that one must learn some vocabulary before being an English-speaker.

The expert model has already been implemented [BDS93]. *Logical representations* are used to formalize the meaning of a compound or of a definition, that is, the predicate, the *arguments* of this predicate that have to be satis-

[6]Deverbal compounds are the ones that are formed with a suffixed verb, for example "truck-driver".



$x \rightarrow y$ means that learning concept y is a prerequisite for learning concept x. The concepts in bold face are the ones that are presented in this paper.

Figure 1: Conceptual graph of the linguistic expert knowledge

fied, and their respective *roles*. A logical representation is noted: <pred> (<role$_i$>:<arg$_i$>,<role$_j$>:<arg$_j$>), where <pred> is the predicate, <arg$_k$> an argument[7], and <role$_k$> the role played by the kth argument. For example, the logical representation of "truck-driver" or of "someone who drives trucks" is: "drive"(agent:"driver",theme:"truck"), whereas "giraffe-bird" and its definition correspond to: "look_like"(agent:"bird", theme:"giraffe"). Such logical representations are used to formalize the expert knowledge, through designing compounding patterns which represent the associations made by an expert between compound types and logical representations, and conversely. For example, the expert knowledge concerning the computation of the semantics of N-Ver compounds is formalized by: <arg$_i$>-<pred>er→<pred>(agent:<pred>er,<role$_i$>:<arg$_i$>).

Such a formalization of the expert knowledge can be applied to other domains. These domains are the ones for which the corresponding knowledge can be formalized into a set of production rules that are directed by a set of concepts. This seems to be the case for at least most of the sub-domains of linguistics.

We now turn to the student modeling point. The following issues about student modeling can be applied to any domain for which the expert knowledge can be expressed through concepts, like the one described above.

3 Student modeling

The main purpose of a student modeling system is to dynamically elaborate a knowledge base that represents the student's knowledge. Elaborating such a system implies to determine both a knowledge representation formalism, and the mechanisms that acquire new information about the student's knowledge through the diagnosis of the student's last answer, and that synthesize this new information with the previously acquired ones [Kas87].

The solutions proposed for these problems must have a number of characteristics in order for the student modeling system to build useful student models. In this section, we first enumerate what these properties are, both from an empirical and a theoretical point of view. We then present an im-

[7]External (i.e. agent, instrument), internal (e.g. theme for transitive verbs), or even semantic complements (locative, manner, etc.).

plementation that tackles most of these properties, and we point out where, when, and how the other notions intervene.

3.1 Empirical remarks

Three remarks can be made about student knowledge. These points have been brought to light through a manual analysis of paper tests about compounds, but appear to be quite general (i.e. domain-independent).

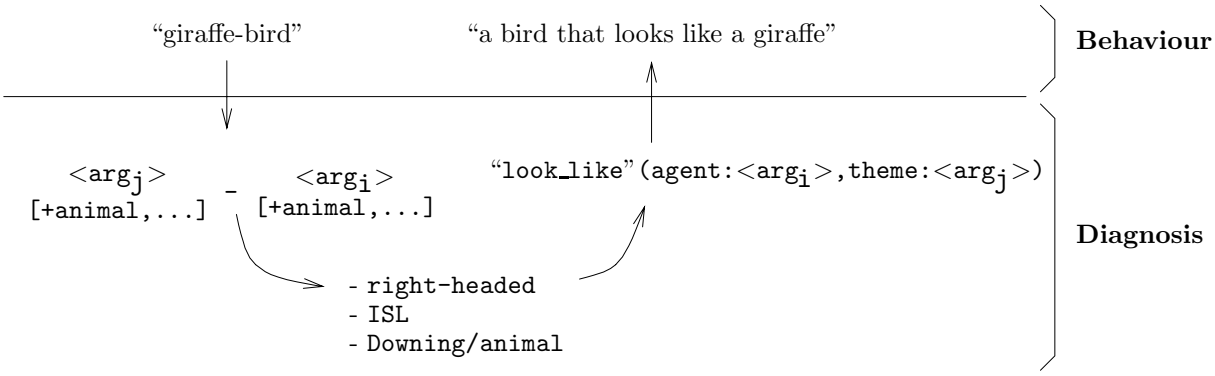
Knowledge evolution First, since the student is in a learning situation, his/her knowledge is expected to evolve in time (this point is also stressed in [PSH94]). So, a student may first incorrectly answer (resp. correctly), and, after that, correctly answer (resp. incorrectly). For example, the student can incorrectly use the `Downing/animal` concept, e.g. answering "a fish that eats cucumbers" for "cucumber-fish"[8], and, after that, correctly use this concept and give the meaning "a bird that looks like a giraffe" to "giraffe-bird".

Contradictions A second remark is that he/she may have contradictions in mind (this point forms the basis of [KIM93]). For example, the student may believe that "killer-shark" and "shark-killer" both denote "a shark that kills people". The student may be aware of this contradiction. That is, he/she may have a set of hypotheses about some pieces of knowledge he/she is not sure of, and wait for more information before taking a decision (a point of view that underlies J. Self's CTP tutor [Sel87]). He/She also may not be aware of this contradiction: the contradiction may be due to mere inconsistent pieces of knowledge.

Uncertainty in diagnosis The final point is that the diagnosis of his/her answers can be erroneously performed (even by a human tutor). This may be due to the fact that not enough information is available at diagnosis time. For example, the student may give the definition "a bird that looks like a giraffe" for the compound "giraffe-bird". This answer seems correct because

[8] Up to now, students are only proposed three kind of exercises: one where they have to tell the definition for a given compound; one where they have to produce the compound that corresponds to a definition; and a last one where they are asked to produce as many compounds as possible, as well as their respective definitions, given a list of nouns, a list of verbs, and a list of suffixes.

it matches the expert's one as defined above. But a deeper analysis shows that this answer is not obviously correct. Indeed, this answer is correct if and only the student relies on the concepts that the expert uses; namely, for the example above: `right-headed`, `ISL`, and `Downing/animal` (cf. section 2). This diagnosis can be pictured by figure 2.



Square brackets unfold the features of the word.

Figure 2: Diagnosis of the underlying linguistic concepts of an answer

Two diagnoses can in fact be elaborated:

1. An optimistic one: the student has used the same concepts than the expert.

2. A pessimistic diagnosis: the student has relied on other concepts. For example, the student may have used an individual version of the correct `Downing/animal` concept. For instance, he/she could have taken into account the fact that both "bird" and "giraffe" denote animals. If so, the incorrect concept used by the student may be: "When the two constituents of a compound belong to the same semantic class, a resemblance relation links them.". This concept is incorrect, although it produces the correct surface solution.

In a pedagogical purpose, the optimistic diagnosis may be prefered. The student is therefore considered as mastering the concepts used by the expert. But if a subsequent diagnosis proves that the student does not in fact master some of these concepts, the first diagnosis has to be changed. For example, the student may further produce "a fish that eats cucumbers" for "cucumber-fish", instead of "a fish that looks like a cucumber".

We state that a student modeling system must be able to take into account these points in order to build sound models of students.

The problem now is to determine such a framework for student modeling in formal terms. They are several advantages in a formalization. First, it makes the framework reusable for other domains. It also facilitates the comparison between already existing representation formalisms, and the determination of a new one if necessary. Moreover, it makes easier the implementation step, through experience acquired from previous implementations.

3.2 Theoretical notions

The preceding empirical remarks can be formalized into three theoretical notions.

3.2.1 Defeasible reasoning

We have seen so far that the student knowledge may evolve during the interaction. So, the student model must be able to represent knowledge that issues from a reasoning that is *defeasible*. From a theoretical point of view, such a reasoning can be divided into two complementary sub-notions: non-monotony and temporal evolution [HBC+91].

Non-monotony The monotonic property of a reasoning system characterizes the fact that the knowledge base of this system cannot reduce: once a piece of knowledge is known, it will be known forever. This is the case for classical (i.e. mathematical) logic. On the contrary, *non-monotony* formalizes the fact that getting new information may change the previous knowledge base[9]. This knowledge reduction is due to the fact that the new pieces of knowledge can block the reasoning steps that were used to infer the concerned knowledge.

For example, the student may first have in his/her knowledge the only fact, about N-N compounds, that a "giraffe-bird" is "a bird that looks like a giraffe". The student may infer from this knowledge base that "a resemblance link exists between the two nouns of a compound if and only if the two nouns belong to the same semantic class", which is incorrect. If the tutoring system

[9]Non-monotony can be formalized by: $\exists k : K \models_L k, \neg(K \cup K' \models_L k)$, where L is the logic, K and K' are knowledge bases, k a piece of knowledge, and $\models_L$ the inference scheme of the logic L.

then gives the student a new information which corresponds to a counter-example of this inferred knowledge (say, "cucumber-fish" is "a fish that looks like a cucumber"), then the student may no longer infer this incorrect piece of knowledge.

Temporal evolution Another possible evolution in the student knowledge corresponds to a complete change of mind: new information can make the student believe the contrary of his/her previous knowledge. This *temporal evolution*[10] may be due, for example, to the fact that the new knowledge raises a contradiction with some previously acquired knowledge.

For example, the student may initially believe that the head of the compound is the leftmost element. This may be due to the fact that it is so in his/her mother tongue. The diagnosis of this misconception may lead the tutoring system to give a course session. After that, the student is expected to know the correct `right-headed` concept, i.e. the contrary of his/her previous belief.

3.2.2 Paraconsistent reasoning

We have also seen that students may have contradictions in mind. Thus, the student modeling system must be able to represent and to reason on inconsistent knowledge. Within a classical logic framework, anything can be logically inferred from a knowledge base that possesses contradictory pieces of information. A *paraconsistent reasoning* system[11] allows to represent and to take reasoning steps upon inconsistent knowledge without deducing everything.

For example, a student may use both the correct `right-headed` concept and another (incorrect) *left-headed* one, depending on the compound: he/she may always use the correct concept, unless for the compounds that resemble the ones of his/her native language, for which the correct concept is the *left-headed* one (like French, for example). It is clear that such a contradiction does not make the student believe anything about any other concepts. Instead, he/she is always able to rationally use his/her beliefs about the other concepts.

[10] Temporal evolution can be formalized by: $\exists k : K \models_L k, K \cup K' \models_L \neg k$.

[11] Paraconsistent reasoning can be formally defined by: $\exists k : K \models_L k, K \models_L \neg k, \neg(\forall k' : K \models_L k')$.

3.2.3 Hypothetical reasoning

As it has been pointed in section 3.1, a lack of information during the diagnosis phase may lead to erroneous diagnoses. So, the diagnoses must better be considered by the system like hypothetical facts rather than certain ones. This hypothetical information has to be withdrawn from the student model if further information shows that it is incorrect. Thus, the student modeling system must be able to follow a *hypothetical reasoning*.

In this section, we have described properties in terms of reasoning characteristics that a student modeling system must possess.

We now turn to the problem of the determination of an implementation that is able to follow such defeasible, paraconsistent and hypothetical reasonings.

3.3 Implementation

Studies about student modeling have emphasized the need to model not only the student's behavior but also the *reasons* that underly this behavior. This distinction corresponds to the *deep vs shallow* knowledge differentiation which has issued from expert system design. The shallow knowledge is a procedural description of the problem solutions. The deep knowledge consists of domain theories and problem solving knowledge [DS91]. For this reason, E. Wenger [Wen87] discriminates between *behavioral diagnosis* and *epistemic diagnosis*. The former deals with the student's behavioral knowledge, whereas the latter is related to the concepts the student relies on. This distinction is also stressed, among others, in the framework that P. Dillenbourg and J. Self provide in order to compare different learner modeling systems [DS91][12].

Our approach to the domain-independent student modeling point is therefore divided into two complementary tasks: *behavioral modeling* and *conceptual modeling*. The former models the student's behavior since the beginning of the interaction, and takes into account the use of licit as well as illicit patterns of behavior. The latter models the student's knowledge in terms of

[12] Within this framework, the behavioral student model is denoted $R_S R_L bk$, where $R_S \ldots$ means "the representation that the system has of ...", $R_L \ldots$ stands for "the representation that the student has of ...", and bk denotes the behavioral knowledge corresponding to the target domain. The conceptual student model is $R_S R_L ck$, where ck represents the conceptual knowledge of the target domain.

correct/incorrect use/non-use of correct concepts. This separation between these two levels has the advantage to make a clear distinction between the observations of the student's *behavior* and the inferences of the *concepts* (that is, the reasons) that underly his/her behavior (cf. figure 2).

In this section, we describe the choices we have made for each of these levels as far as the two modeling problems, namely the determination of the student's knowledge representation formalism and the automatic acquisition/synthesis processes of his/her knowledge by the ITS, are concerned. We show how the chosen formalism can take into account defeasible and paraconsistent reasonings. We also point out when, where, and how the mechanisms that implement hypothetical reasoning have to intervene.

3.3.1 Behavioral modeling

The behavioral model is a synthesis of the student's behavior. Therefore, it takes into account the whole set of the student's answers, regardless they are correct or not. So, by definition, it models the student's misconceptions. This is one of the requirements for student models.

Knowledge representation Probabilistic logic has been chosen as knowledge representation formalism. *Probabilistic logic* is a non-classic numerical logic where a certainty coefficient is associated to each proposition of the language. This coefficient is a real number between 0 and 1 that denotes the probability for the proposition to be true [HBC+91]. Intuitively, this formalism allows to manipulate sentences like "It is likely that p is true." where p is a proposition of the language.

They are several reasons for this choice. The main reason is that probabilistic logic is a non-monotonic logic that allows to implement the student's paraconsistent as well as defeasible reasonings, as we are going to show. Logical formalisms have been chosen because they guarantee a clear semantics: different readers of a given formula should understand it in the same way [Bes89]. This is generally not the case for other formalisms like semantic graphs or objects. Moreover, knowledge bases expressed in such formalisms can be expressed using a logical formalism [TGL+88]. Logical formalisms also enable to describe information in a way close to both natural language and a synthetic language [HBC+91]. Another important point is that logical formalisms possess reasoning mechanisms that allow the system to automatically manipulate its knowledge (it can deduce new knowledge from its

previous knowledge, for example).

Using probabilistic logic, the behavioral model is implemented as a set of *behavioral entities*, each of which denotes an association made by the student between a kind of behavior (a proposition of the formal language) and its certainty coefficient. A *kind of behavior* synthesizes the associations made by the student between a kind of definition and a category of compound, and conversely. The *certainty coefficient* intuitively corresponds to the probability that the student behaves as denoted by the associated type of behavior, rather than as formalized by another behavioral entity that concerns the same kind of question. A formal syntax of the behavioral model is:

$$
\begin{array}{lcl}
\text{<behavModel>} & = & \text{<behavEntity>}^* \\
\text{<behavEntity>} & = & (\text{<behavType>}, \text{<certaintyCoeff>}) \\
\text{<behavType>} & = & \text{<compoundCode>} \rightsquigarrow \text{<definitionLogRep>} \mid \\
 & & \text{<definitionLogRep>} \rightsquigarrow \text{<compoundCode>} \\
\text{<certaintyCoeff>} & \in & [0;1]
\end{array}
$$

Such a behavioral student model allows to tackle both student's paraconsistent and defeasible reasonings. For example, as far as paraconsistent reasoning is concerned, the student may both correctly answer "a bird that looks like a giraffe" for the compound "giraffe-bird", and incorrectly answer "a cat that looks like a fish" for "cat-fish" (i.e. the same kind of compound, but in one case the leftmost constituent is considered to be the head of the compound). The corresponding part of the behavioral model is then:

$$
\left\{ \begin{array}{l}
\left(\begin{array}{c} \texttt{<arg}_\texttt{j}\texttt{>} \\ \texttt{[+ani,...]} \end{array} \texttt{-} \begin{array}{c} \texttt{<arg}_\texttt{i}\texttt{>} \\ \texttt{[+ani,...]} \end{array} \rightsquigarrow \text{“}\texttt{look\_like}\text{”}\texttt{(agent:<arg}_\texttt{i}\texttt{>,theme:<arg}_\texttt{j}\texttt{>),<}p_1\texttt{>} \right), \\
\left(\begin{array}{c} \texttt{<arg}_\texttt{j}\texttt{>} \\ \texttt{[+ani,...]} \end{array} \texttt{-} \begin{array}{c} \texttt{<arg}_\texttt{i}\texttt{>} \\ \texttt{[+ani,...]} \end{array} \rightsquigarrow \text{“}\texttt{look\_like}\text{”}\texttt{(agent:<arg}_\texttt{j}\texttt{>,theme:<arg}_\texttt{i}\texttt{>),<}p_2\texttt{>} \right)
\end{array} \right\}
$$

which effectively models the inconsistence.

Besides, defeasible reasoning can be relatively easily taken into account by implementing the notion of *temporal relativization*. This notion issues from the pedagogical hypothesis that the more recent a student's answer is, the more it denotes his/her current knowledge state. This can be implemented through expressing order constraints over certainty coefficients (on the preceding example, through forcing $p_1 < p_2$).

So far, we have shown how probabilistic logic takes into account all the characteristics that a student's knowledge representation formalism should

have. Hypothetical reasoning does not intervene at this level of modeling since no hypothesis has to be set by the behavioral modeling system, simply because no diagnosis is done.

We are now going to describe how this formalism allows to implement adequate knowledge acquisition and synthesis mechanisms.

Knowledge acquisition and synthesis As it has been said earlier, in order to build a student model, the modeling system first has to acquire new information through analyzing the student's behavior, and then has to synthesize this new information with the one that has been previously acquired.

Information acquisition New information is collected from the syntactico-semantic analysis of each interaction. From this analysis ensue both the compound coding and the logical representation of the definition, i.e. the compounding pattern that the student has used. For example, the analysis of the compound "giraffe-bird" and of the correct definition associated by the student with this compound (cf. figure 2) gives the following pattern:
$<\mathtt{arg_j}>$`[+ani,...]-`$<\mathtt{arg_i}>$`[+ani,...]`$\rightsquigarrow$"`look_like`"`(agent:`$<\mathtt{arg_i}>$`,theme:`$<\mathtt{arg_j}>$`)`.

Information synthesis The update only concerns the part of the behavorial model that corresponds to the current type of question. It can be performed after the analysis of each answer. The update concerns the evolution of the contents of the behavioral model, and the evolution of the certainty coefficients[13]. These modifications are listed below.

1. Each coefficient associated with a behavioral entity that concerns the current type of question is weighted by a given value τ. This weighting results in the diminution of all the certainty coefficients associated with older behavior types. This diminution, allowed by the use of a numerical logic, implements temporal relativization. The works of M. Kuzmycz and G. Webb about the FBM system [KW92] give value $\frac{9}{10}$ to τ. As only a few works are related to the modeling of student cognitive evolution, and because they have validated this particular

[13]The initial contents of the behavioral model depend on the desired pedagogical initialization type, as will be shown in section 3.3.3.

valuation, their results are used to implement the notion of temporal relativization[14] in COMPOUNDS.

2. If the new behavorial entity (the nth one) does not already belong to the model, then it must be added to it. Let p_j^t be the coefficient associated with the jth entity. Because the sum of the probabilities must be the unity, the old probabilities must be redistributed: they are divided by n. But, in the same time, the other probabilities are weighted by τ (figure 3 shows the evolution of the probabilities when new entities are added to the model). The point now is to compute the value α_n that weights the new behavioral entity. This value corresponds to the solution of the system of equations, i.e. $\alpha_n = n - \tau(n-1)$ if α_1 is 1. The probability p_n^t is then $\frac{n-\tau(n-1)}{n}$.

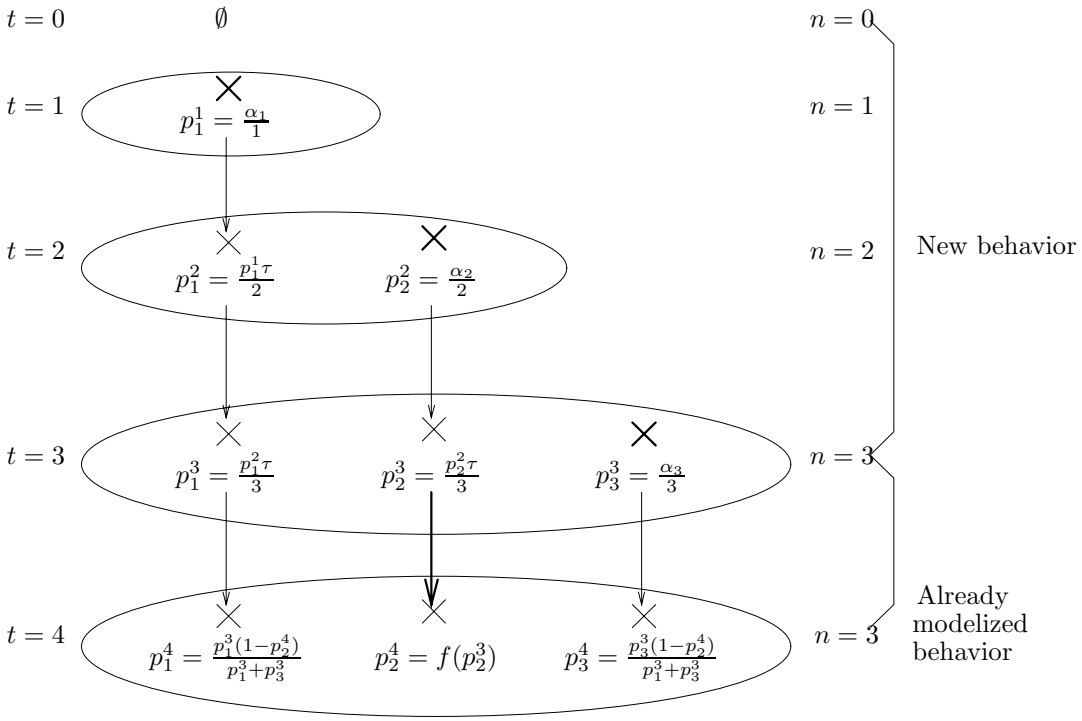


Figure 3: The different evolutions of the subset of the behavioral model related to a given kind of question

[14]It corresponds to FBM's *data aging* mechanisms.

3. If this entity, the ith, already belongs to the model, its probability p_i^t has to be computed again. This probability must be increased, in order to respect the probabilistic theory. The main problem is to determine an (increasing) function f that is pedagogically valid. We use $\sqrt{\ }$ for f. This function issues from the works of L. Chen and B. Kurtz about the XTra-TE ITS [CK89]. This choice relies on the same reasons than the choices underlying the valuation of τ. We thus get $p_i^{t+1} = \sqrt{p_i^t}$. The remaining problem is then to compute the weights of the other probabilities in order to still verify the $\sum_{k=1}^{n} p_k^t = 1$ property. To solve this system of equations, one constraint at least must be added. It seems intuitive to ensure that the distance between old probabilities and new ones is the same for all the behavorial entities related to the current question:

$$\forall (k_1, k_2)[(k_1, k_2) \in [1; n]^2 \wedge k_1 \neq i \wedge k_2 \neq i] \Rightarrow \frac{p_{k_1}^{t+1}}{p_{k_1}^t} = \frac{p_{k_2}^{t+1}}{p_{k_2}^t}$$

The new probability p_j of the jth behavioral entity ($j \neq i$) is then:

$$p_j^{t+1} = \frac{p_j^t(1 - f(p_i^t))}{\sum_{k=1, k\neq i}^{n} p_k^t}$$

Moreover, the global scheme, i.e. without taking into account the features of the predicative arguments, may belong to the model. In this case, the problem is to compute an entity that covers both the current entity and the new one. Indeed, in an intuitive way, when a student answers a certain kind of definition for a compound which is coded `<arg_j>[+animal]-<arg_i>[+animal]`, and subsequently answers the same type of definition for a compound whose coding is now `<arg_j>[+natural object]-<arg_i>[+animal]`, then we infer that the features of the non-head constituent are not taken into account by the student. The following new entity is obtained (by computing the intersection of the sets of features):

$$\left(\begin{array}{l} \texttt{<arg}_\texttt{j}\texttt{>} \\ \texttt{[]} \end{array} \texttt{ - } \begin{array}{l} \texttt{<arg}_\texttt{i}\texttt{>} \\ \texttt{[+ani]} \end{array} \rightsquigarrow \text{“}\texttt{look\_like}\text{”}\texttt{(agent:<arg}_\texttt{i}\texttt{>,theme:<arg}_\texttt{j}\texttt{>)}, p_i^{t+1} = f(p_i^t) \right).$$

This entity covers the two types of behavior.

The new entity may also belong to the model, but in an inverse sense (compound towards definition, or conversely). In the example above,

the new entity corresponds to the production of a compound whose coding is <arg$_j$>[]-<arg$_i$>[+animal,...] for a definition whose logical representation is "look_like"(agent:<arg$_i$>,theme:<arg$_j$>). Experimental tests tend to prove that students do not realize that the two modes are linked, i.e. that the relation between a compound and a definition, and conversely, is a bijection. So this link is not taken into account while searching for an entity that could cover the new behavior.

Figure 3 given above shows an example of certainty increasing ($t = 4, i = 2$).

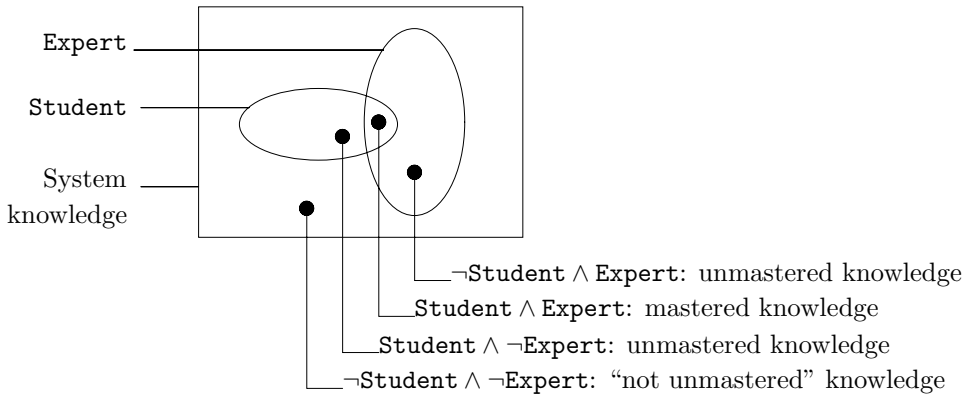
3.3.2 Conceptual modeling

The conceptual student model represents the mastering level that the student possesses on each concept that must be learnt.

Knowledge representation Since conceptual modeling only concerns the expert concepts, the conceptual student model is an *overlay* model [Wen87]. That is, the student knowledge is bound to be included into the expert knowledge. This characteristic is not too restrictive at this level of modeling because the only interesting information about the student's cognitive state concerns the use/non-use of *correct* concepts. So, here, a knowledge representation in terms of *mastering/not mastering* is adequate [Ohl92, Wen87].

Therefore, the mastering level of each concept is measured by four numerical values, which respectively correspond to the number of times the student has used or not a concept when the expert has used it or not (cf. figure 4).

Because of the overlay characteristic, the conceptual knowledge representation can be implemented with a conceptual graph, where each node is decorated with the historic of the student use/non-use of the corresponding concept, depending on the expert use/non-use. It thus deals with *conditional probabilities*. For example, for a given concept (say γ), the conceptual model can contain, after a given interaction:



For a given interaction, `Expert` means that the expert has used these concepts, and `Student` means that the student has used these concepts.

Figure 4: Conceptual diagnosis in differential modeling

Concept γ				Explanations
`Expert`		`¬Expert`		
`Student`	`¬Student`	`Student`	`¬Student`	
0/0	0/0	0/0	0/0	Initialization
0/0	0/0	0/1	**1/1**	Both the expert and the student have not used the concept.
0/1	**1/1**	0/1	1/1	The expert has used the concept, but the student has not.
1/2	1/2	0/1	1/1	Both the expert and the student have used the concept.
1/3	**2/3**	0/1	1/1	The expert has used the concept, but the student has not.
1/3	2/3	0/2	**2/2**	Both the expert and the student have not used the concept.

This part of the conceptual model indicates that there is finally a $\frac{1}{3}$ probability for the student to use the concept γ if the expert uses it (`Student` $\wedge$ `Expert`), and a $\frac{2}{2}$ (=1) probability for the student not to use the concept if the expert also does not use it ($\neg$`Student` $\wedge$ $\neg$`Expert`).

This representation enables to model the paraconsistent reasoning of the student through allowing contradictory entries in the historic of the student's conceptual knowledge. This knowledge representation also gives information

about the student's temporal cognitive evolution. Indeed, if the student first gives n incorrect answers and, after that, m correct answers, he/she is supposed to master the concerned concept. One can note here that this revision of the evaluation of the student knowledge does not depend on a special value for the ratio m/n. This reflects the fact that such a threshold may be different across teachers, whereas this pedagogical behavior is commonly accepted. But keeping the whole history[15] of the information about the student's use of each concept, rather than simply a synthesis[16] brings up a practical problem of memory space encumbrance. Meanwhile, keeping the history has the advantage of permitting to take into account the notion of temporal relativization. This cannot be done if only the synthesis is kept. Moreover, as far as the hypothetical reasoning is concerned, the history makes easier the updating of the conceptual model, when an incorrect hypothesis has to be retracted.

Knowledge acquisition and synthesis To dynamically elaborate the student's conceptual model means to get new information on the conceptual knowledge of the student, through diagnosing his/her answers, and then to synthesize this new information with the current student model.

Information acquisition The diagnosis of the student's answer is performed through the analysis of the differences between the expert and the student's answers (cf. figure 4).

Some of the ITS that deal with second language teaching use both language grammars in order to better understand the linguistic reasons that direct the student's behavior. This implies a better understanding of the student's knowledge. These ITS are based on the *contrastive analysis theory* [Coo94]. This theory states that some of the difficulties encountered by students while learning a nth language come from the previously learnt languages. This theory is in opposition with the L_1=L_2 *theory*[17], which considers that the same learning mechanisms intervene in learning any language (the mother tongue, or the nth). Examples of ITS that deal with second language acquisition are given below. Most of them adopt the contrastive

[15] In practical words, this means the whole table.

[16] That is to say the last line of the table.

[17] L_1 denotes the first language learnt (i.e. the mother tongue), and L_2 any other language learnt.

analysis framework.

L. Ghemri [Ghe92] relies on the *Government and Binding theory.* This theory combines an universal grammar with a set of parameters whose values are language-dependent. These parameters restrict the too numerous productions of the universal grammar to the given language grammar productions. The student model then corresponds to the set of values of these parameters. So, the analysis of the differences between the expert and the student's answers corresponds to the comparison between the values assigned by the student to these parameters and those assigned by the expert. The expert's values correspond to the ones of the language being taught. The student's values are those linked to the language being learnt if the answer is correct. They can also match the values of another language known by the student. Y. Wang and R. Garigliano [WG92] use both Chinese and English grammars to build an ITS that can deal with interferences between the mother tongue and the target language. E. Schuster [Sch86], in her VP^2 ITS, uses a formal grammar of the student's mother tongue. This grammar enables the tutoring system to adapt its behavior, in order to deal with problems that are linked to the mother tongue influence. C. Tschichold *et al.* [TBEC+94] also treat the possible interferences between the mother tongue and the taught language, in their environment designed to help French students in writing English texts. This analysis enables to automatically detect grammatical errors, and to provide adequate corrections.

In Compounds, the contrastive analysis theory cannot be used because the generative and interpretative patterns for French compounds are not currently formalized. Meanwhile, this theory seems to be appropriate. Indeed, some students apparently use the leftmost constituent of the compound as the head when they produce compounds, instead of the rightmost one [Bou92, Dal93]. This left-headed property is, among others, one French characteristics.

Information synthesis To update the conceptual model means to compute the new conditional probability of each concept.

For example, let the interaction be:

1. The student correctly answers to "giraffe-bird": he/she produces "a bird that looks like a giraffe".

2. The student incorrectly answers for "cucumber-fish", producing "a fish that eats cucumbers".

This interaction provides, for the `Downing/animal` concept, the following evaluation:

Student's knowledge evaluation for `Downing/animal` concept				
Diagnosis	`Expert`		`¬Expert`	
	`Student`	`¬Student`	`Student`	`¬Student`
<Initialization>	0/0	0/0	0/0	0/0
1 : `Student`∧`Expert`	**1/1**	0/1	0/0	0/0
2 : ¬`Student`∧`Expert`	1/2	**1/2**	0/0	0/0

The initial values of the conditional probabilities depend on the type of the chosen initial model.

The mechanisms that implement hypothetical reasoning must permit the change of the model when a hypothesis is withdrawn. Indeed, in the preceding example, the first provided diagnosis (`Student`∧`Expert`) is hypothetical. If a subsequent interaction shows that this hypothesis was too optimistic, the conceptual model has to be modified. This modification implies to "backtrack", and to compute the new model, i.e. to replace the first optimistic incorrect diagnosis by the correct diagnosis: `Expert`∧¬`Student`. The conceptual model then becomes:

Student's knowledge evaluation for `Downing/animal` concept				
Diagnosis	`Expert`		`¬Expert`	
	`Student`	`¬Student`	`Student`	`¬Student`
<Initialization>	0/0	0/0	0/0	0/0
1 : ¬`Student`∧`Expert`	0/1	**1/1**	0/0	0/0
2 : ¬`Student`∧`Expert`	0/2	**2/2**	0/0	0/0

Meanwhile, because the pedagogical validity of such mechanisms is not obvious, they are not currently implemented. In fact, the former diagnosis may be correct: the new answer may be due to a student's knowledge evolution rather than to an erroneous diagnosis.

3.3.3 Initial model

The initial student model can be an *empty model*, which denotes that no information is available concerning the student's knowledge, or a *stereotype model* [BM92], which contains *a priori* information about the student's knowledge. This choice depends on the plurality of the users of the system. Indeed,

a human tutor who meets for the first time a student cannot give *a priori* information about his/her knowledge. Meanwhile, if the tutor is the teacher of a group of students that are judged to possess the same knowledge level, he/she can expect a given behavior from the student. Once the stereotypic model defined, it can be used as initial model for some students, in order to increase the student modeling effectiveness.

Initial behavioral model The initial behavioral model can be the empty one, which denotes nothing about *a priori* student's knowledge: the empty set.

A stereotypic behavioral model would be a predetermined set of the behavioral entities, with their respective certainty coefficients, that models the usual student's behavior.

Initial conceptual model An empty conceptual model is a conceptual graph whose values are unknown, i.e. all conditional probabilities are uncomputable (divide by 0).

A stereotypic conceptual model corresponds to a conceptual graph whose values denote the expected student's mastering level.

4 Conclusion

An intelligent tutoring system is supposed to behave like a human tutor. This characteristic mainly corresponds to its ability to adapt the instruction to each student. In order to do this, the system must possess information about the student. Namely, the system must have information about, among others, what the student knows, to which level he/she knows this, what he/she does not know, and his/her misconceptions. This information set is dynamically elaborated by the student modeling component of the tutoring system.

This paper deals with the determination of a system that creates student models containing such information. Two main points issue from our works. First, we propose a formalization of a student modeling system in terms of domain-independent properties that such a system must have in order for it to build a relevant student model: defeasible, paraconsistent (which are related to the student), and hypothetical (related to the diagnosis) reasonings.

These properties must be possessed by both the student knowledge representation formalism and the knowledge acquisition/synthesis processes. Then, we describe one possible implementation of such a student modeling system. This implementation relies on probabilistic logic. It takes into account defeasible as well as paraconsistent reasonings. Hypothetical reasoning has not yet been implemented. This is due to the fact that we do not know how to distinguish a cognitive evolution from an incorrect diagnosis. Nonetheless, we have shown where it should intervene.

The main advantage of our works is that, since the enumerated properties are domain-independent, our framework can be applied to other domains than the one used as example in this article. In fact, any tutoring system whose domain can be formalized in terms of *concepts* can make use of such a student modeling system in order to individualize the instruction.

Some points have still to be studied, both at a practical and pedagogical level, as far as student modeling is concerned.

First, the mechanisms that implement hypothetical reasoning must be elaborated. In parallel, once the implementation will be finished, the exhaustiveness of the student modeling system must be empirically evaluated. That is, the rate of students for which a sound and accurate cognitive model can be obtained by a student modeling system based on our framework has to be determined.

From a more pedagogical standpoint, several problems remain. For instance, the hypothesis that underlies the notion of temporal relativization has to be confirmed. In addition, the pedagogical validity of the valuation of the function f issued from XTra-TE, and of the parameter τ issued from Fbm has to be tested. Indeed, these values have been pedagogically checked within their respective frameworks, but have yet to be validated in our framework. Moreover, stereotypic student models have to be empirically elaborated. In parallel, the evaluation system of the values associated with non-terminal nodes of the conceptual student model has to be determined. Some additional information (cf. figure 5) may also be inferred from the analysis of the differences between expert and student answers, and the answers obtained by executing the student model. Furthermore, psychological attributes that depict the student should be incorporated into the student model, in order for the tutoring system to have at its disposal the remaining pedagogical functions described by J. Self. This implies to determine both

their manual determination, their automatic and dynamic management, and their pedagogical utility.

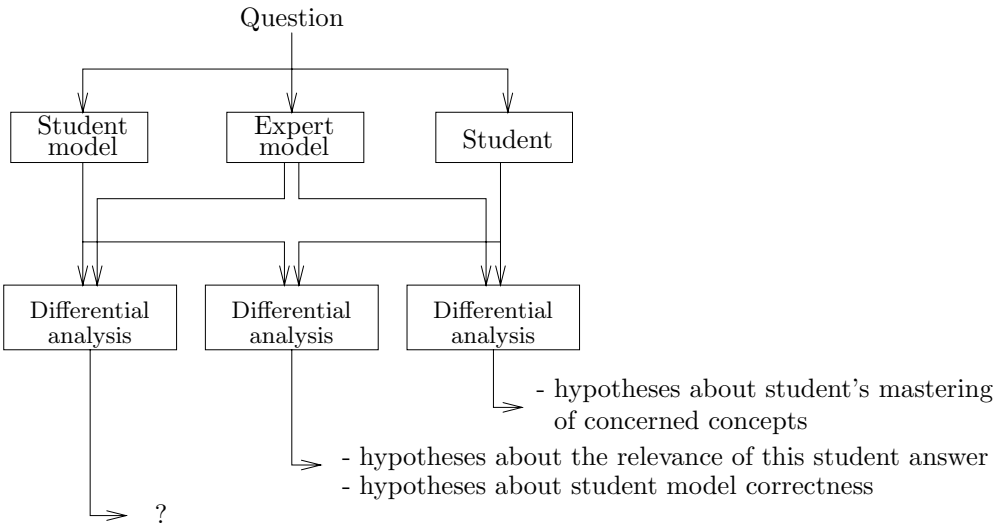


Figure 5: Differential analysis of student answers

The hypotheses that are used during the diagnosis phase must also be validated from a pedagogical point of view. One hypothesis states that, if a student does not give any answer, the concerned concepts are diagnosed to be not mastered. Another hypothesis says that optimistic diagnoses are prefered to pessimistic ones. For example, if the student correctly answers for "giraffe-bird", he/she is said to master the `Downing/animal` concept. Moreover, students are supposed to understand the meaning of the definition they are provided. They are also supposed not to have vocabulary problems. One can note that this last hypothesis is not too restrictive because it corresponds to what a student has to know before learning compounding processes. A final hypothesis states that students are supposed to be fully attentive. Thus, their answers always correspond to their cognitive states. This hypothesis hides the problem of the noise, whose solving implies to be able to differentiate noisy information from sound one.

Only a new empirical study seems convenient to test these pedagogical issues.

References

[BC88] Hugh L. Burns and Charles G. Capps. Foundations of Intelligent Tutoring Systems: An Introduction. In Martha C. Polson and J. Jeffrey Richardson, editors, *Foundations of Intelligent Tutoring Systems*. Lawrence Erlbaum Associates, Inc, 1988.

[BDS93] Paul Boucher, Frédéric Danna, and Pascale Sébillot. COMPOUNDS: an Intelligent Tutoring System for Learning to use Compounds in English. *Computer Assisted Language Learning*, 6(3):249–272, 1993.

[Bes89] Philippe Besnard. Logiques formelles et raisonnement de bon sens. *Annales des télécommunications*, 44(5–6):242–250, 1989.

[BM92] Paul T. Baffes and Raymond J. Mooney. Using Theory Revision to Model Students and Acquire Stereotypical Errors. In *Fourteenth Annual Conference of the Cognitive Science Society*, pages 617–622, 1992.

[Bou92] Paul Boucher. Teaching Compound Nouns in English: an application of the Charlie Brown principle. In *CIEREC/TRAVAUX LXXVI, Université de Saint-Étienne*, pages 33–50, 1992.

[CK89] Li Chen and Barry L. Kurtz. XTRA-TE: Using Natural Language Processing Software to Develop an ITS for Language Learning. In *Artificial Intelligence and Education*, Amsterdam, 1989.

[Coo94] Vivian Cook. *Linguistics and Second Language Acquisition*. MacMillan Press, 1994.

[Dal93] Sébastien Dallain. Testing French students in understanding and producing English compound nouns. Master's Thesis, Université de Rennes 2, 1993.

[Dow77] Pamela Downing. On the Creation and Use of English Compound Nouns. *Language*, 53(4):810–842, 1977.

[DS91] Pierre Dillenbourg and John A. Self. A Framework for Learner Modelling. *Interactive Learning Environments*, 2(2):111–137, 1991.

[Ghe92] Lila Ghemri. Intelligent tutoring systems. In *Intelligent Tutoring Systems*, number 608 in Lecture Notes in Computer Science, chapter A Cognitive Framework for Second Language Error, pages 260–268. Springer-Verlag, 1992.

[HBC+91] Jean-Paul Haton, Nadjet Bouzid, François Charpillet, Marie-Christine Haton, Brigitte Lâasri, Hassan Lâasri, Pierre Marquis, Thierry Mondot, and Amadeo Napoli. *Le raisonnement en intelligence artificielle — Modèles, techniques et architectures pour les systèmes à base de connaissance*. InterEditions, 1991.

[Kas87] Robert Kass. The Role of User Modelling in Intelligent Tutoring System. Technical report, Moore School, University of Pennsylviana, 1987.

[KIM93] Yasuyuki Kono, Mitsuru Ikeda, and Riichiro Mizoguchi. A Modeling Method for Students with Contradictions. *Artificial Intelligence and Education*, pages 473–481, 1993.

[KW92] Mark Kuzmycz and Geoffrey I. Webb. *Intelligent Tutoring Systems*, chapter Evaluation of Feature Based Modelling in Subtraction, pages 269–276. Number 608 in Lecture Notes in Computer Science. Springer-Verlag, 1992.

[Lie83] Rochelle Lieber. Argument Linking and Compounds in English. *Linguistic Inquiry*, 14(2):251–285, 1983.

[Ohl92] Stellan Ohlsson. Constraint-Based Student Modelling. *Journal of Artificial Intelligence in Education*, 3(4):429–447, 1992.

[PSH94] Ana Paiva, John A. Self, and R. Hartley. On the Dynamics of Learner Models. *ECAI*, pages 178–182, 1994.

[SB82] D. Sleeman and J. S. Brown. Intelligent Tutoring Systems. In B. R. Gaines, editor, *Intelligent Tutoring Systems*, Computers and People, chapter Introduction. Academic Press Inc., 1982.

[Sch86] Ethel Schuster. The Role of Native Grammar in Correcting Errors in Second Language Learning. *Computational Intelligence*, 2:93–98, 1986.

[Sel82] Elisabeth O. Selkirk. *The Syntax of Words*. Cambridge Mass., MIT Press, 1982.

[Sel87] John A. Self. Students Models: What Use Are They? *IFIP/TC3*, pages 73–85, May 1987.

[Sel90] John A. Self. Bypassing the Intractable Problem of Student Modelling. In *Intelligent Tutoring Systems: At the crossroads of AI and Education*. C. Frasson and G. Gauthier, 1990.

[TBEC+94] Cornelia Tschichold, Franck Bodmer, Étienne Cornu, François Grosjean, Lysiane Grosjean, Natalie Kübler, and Corinne Tschumi. Detecting and Correcting Errors in Second Language Acquisition. *Computer Assisted Language Learning*, 7(2):151–160, 1994.

[TGL+88] André Thayse, Pascal Gribomont, Georges Louis, Dominique Snyers, Pierre Wodon, Paul Gochet, Éric Grégoire, Edouardo Sanchez, and Philippe Delsarte. *Approche logique de l'Intelligence Artificielle*, volume 1. Dunod informatique, 1988.

[Wag91] Horst Wagner. Dictionnaires, bases de données lexicales et lexicographie des langues de specialité : le traitement des unités complexes. In *Informatique et Langue Naturelle*, Nantes, France, 1991.

[Wen87] Etienne Wenger. *Artificial Intelligence and Tutoring Systems — Computational and Cognitive Approaches to the Communication of Knowledge*. Morgan Kauffman publishers, Inc., 1987.

[WG92] Yang Wang and Roberto Garigliano. *Intelligent Tutoring Systems*, chapter An Intelligent Tutoring System for Handling Errors Caused by Transfer, pages 395–404. Number 608 in Lecture Notes in Computer Science. Springer-Verlag, 1992.