Formal Approaches to Student Modelling

John Self

1994

AAI/AI-ED Technical Report No.92

(in McCalla, G.I. and Greer, J. (eds.), *Student Modelling: the key to individualized knowledge-based instruction*, 295-352, Berlin:Springer-Verlag))

Computing Department
Lancaster University
Lancaster
LA1 4YR
UK

aai@comp.lancs.ac.uk
http://www.lancs.ac.uk/computing/research/aai-aied/
ftp.comp.lancs.ac.uk (148.88.8.9) /pub/ai/

Formal Approaches to Student Modelling

John A. Self

Computing Department, Lancaster University, Lancaster LA1 4YR, U.K.

Abstract: This paper considers student modelling from the point of view of the formal techniques that are involved. It attempts to provide a theoretical, computational basis for student modelling which is psychologically neutral and independent of applications. It is derived mainly from various areas of theoretical artificial intelligence. Because of the intrinsic difficulty of the student modelling problem, these links to AI are often merely pointed out and not pursued in depth.

Contents

1. Introduction

Like all models, a *student model* is intended to provide information about the object modelled, in this case, the individual student who is using a computer-based learning system. The system uses the student model to help determine actions appropriate for that student. *Student modelling* is the process of creating a student model. Student modelling necessarily occurs mainly at run-time, when the student uses the system, since it is mainly through the evidence provided by the student's inputs to the system that the student model is created. This evidence is usually scanty, making student modelling a difficult process.

The aim of this paper is to review various formal approaches to student modelling. Before embarking on this, some words of justification are required, on why student modelling is important and why formal approaches are necessary. Without a student model a computer-based learning system will perform in exactly the same way with all users, since there is no basis for determining otherwise. But obviously, students are different: they have different prior knowledge, different interests, different learning aptitudes, and so on. An *intelligent learning environment (*or *ILE)* is primarily one which understands the individual student well enough to be able to determine individualised actions. A student model does not have to be completely accurate to be useful. Indeed, it is not the case that a more accurate student model is necessarily better: the computational effort to improve accuracy may not justify the extra pedagogical leverage obtained. Computational utility, not cognitive fidelity, is the measure for student models. Thus, we will need to consider how student models are used in ILEs. For the moment, we will simply assume that the student model is data for the instructional component of an ILE.

A number of student modelling techniques have been developed (Dillenbourg and Self, 1992). Generally, these techniques are embedded within more-or-less complete ILEs making it difficult to analyse them in isolation. In order to determine the properties of such techniques (so that we may compare them, specify when each is appropriate, develop refinements, etc.) some formalisation of them may help. It may not - because, as we will see, many of the techniques skirt difficult issues in theoretical artificial intelligence and it may be better to rely on the pragmatic approach of implementation and empirical evaluation. Also, premature formalisation may focus on what is formalisable rather than what is important. However, most sciences progress through formalisation and we may hope that student modelling will as well, in due course. Such a formalisation should be psychologically and educationally neutral, that is, independent of particular psychological theories and educational discussions of student-model-based ILEs.

The remainder of this paper is organised as follows. First, we introduce a general foundation for formal student modelling and a simple example to illustrate it. We then review methods for creating student models, by initialising them and subsequently revising them in the light of system-student interactions. We first imagine an 'ideal student', that is, one who holds no misconceptions and who reasons and learns rationally. We then consider 'real students' who are naturally less considerate and who therefore present considerably more problems to the formalisation effort. The different kinds of content of student models are described. Finally, we re-consider the potential uses of student models in ILEs, in order to re-emphasise that student models are not independent, autonomous components of ILEs but must be fully integrated with other components.

2. Foundations

Our starting point of view is that an ILE is intended to support productive interactions between the belief systems of the student and the system ('productive' in this case meaning that they lead to an 'improvement' in these belief systems, especially that of the student, of course). We will use $B_a p$ to denote that an agent a believes proposition p (we will omit the

subscript when the agent is irrelevant). The belief-set B_a of the agent a is the set of propositions believed by a: $B_a = \{ p \mid B_a p \}$. We will consider the expression $B_a p$ to be itself a proposition and thus that such expressions can be nested: $B_b B_a p$ denotes that agent b believes agent a believes proposition p.

Our basic framework is shown in Figure 1 - we emphasise that this is only a starting point, with the concepts of belief, proposition, etc. to be elaborated in the following. We have three components:

> B_s denotes the student's belief-set;
>
> B_c denotes the computer system's belief-set;
>
> LM denotes that subset of the system's belief-set which are beliefs that the system has about the student. The set of propositions which the system believes are believed by the student will be denoted by B_{cs}, i.e. $B_{cs} = \{ p \mid B_c B_s p \}$. The set $\{ B_s p \mid B_c B_s p \}$ is a subset of LM.

The computer system has no direct access to B_s : all its reasoning about the student has to be through the analysis of LM.

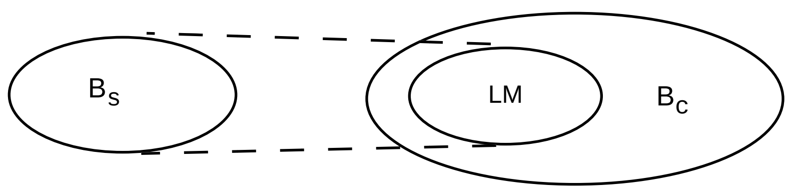

Figure 1. The basic framework

This is intended to be an abstract foundation so general that it can encompass any view of student modelling. It will be made more concrete through examples later. But first we have to say something about the basic concepts of belief and proposition. Philosophers have laboured for millenia over the meaning of such terms, and associated concepts such as knowledge and truth. We will just summarise the main points of relevance to student modelling.

Regarding *belief*, we can make the following points:

1. We can distinguish between the belief itself and the object of the belief, here called the proposition. Thus, we can say "The student's belief that momentum depends only on velocity is reasonable given the examples she[1] has seen", where we are commenting on the belief, not on the proposition itself, that "momentum depends only on velocity".

2. A belief, unlike other attitudes such as regret, wish, etc., can be assessed as true or false, i.e. (to keep it simple!) in accordance with the facts. (We will not consider the everyday use of "belief" in phrases such as "I believe in democracy" - the extent to which such non-propositional beliefs can be reduced to beliefs that certain propositions are true is a matter of controversy.)

3. Beliefs are held (usually) for *reasons,* and often changes in beliefs are a function of an analysis of the reasons that they are held. Thus, our student models will rarely be an unstructured set of beliefs as suggested above but may maintain, for each derived belief, some justification for it.

---

[1] [illegible]

4. Beliefs are related to behaviour: if an agent believes p then it is disposed to act as if p were true. Of course, there is no guaranteed mapping between belief and behaviour (and vice versa): if there were, student modelling would be straightforward. However, some beliefs (e.g. that Neptune is cold) do not appear to have much influence on behaviour (except to enable us to respond "yes" to the question "Is Neptune cold?"). Also, a belief may be 'quiescent', i.e. even though we may consider it to be possessed by an agent (because of previous behaviour perhaps) it may not be manifested in behaviour when one might expect it, perhaps because the agent has for some reason not considered it relevant to the situation at hand.

5. It does not matter whether or not beliefs have any kind of real existence in the mind (whatever that may mean): it is sufficient that agents find it is useful to attribute beliefs to others in order to understand and predict what they do.

Knowledge is usually described in terms of belief - an agent a knows p, $K_a p$, under the following conditions:

$$K_a p \equiv (1)\ B_a p,\ (2)\ p \text{ is true, and } (3) \text{ there is an account for } p.$$

As usual, there is much philosophical debate about this definition, but it will serve our purposes. The third condition is necessary because we would not want to consider that, for example, "I know it is snowing in Moscow" just because I believe it is and it happens to be true, unless I can give a convincing account of why I believe it (e.g. that I have just seen a live television broadcast from the Red Square). This raises the question of when an account is to be considered convincing and thus the belief to be justified. Empiricists claim that all knowledge is acquired through the use of the senses, and hence that no empirical proposition is ever certain, and therefore that we cannot properly claim to know the truth of such propositions. Rationalists, however, consider that it is possible to acquire knowledge by means independent of empirical investigation. In student modelling, part of the computer system's belief set B_c is often considered to represent knowledge, although the fact that it is true is only implicit and the justifications may be absent. The system may consider the student to possess knowledge if parts of B_{cs} correspond to entries in B_c:

$$K_c p \ \&\ B_c B_s p \rightarrow B_c K_s p$$

although this axiom overlooks the role of the account of p, mentioned above.

Computationalists will be happy with the concept of a *proposition*. Clearly, if an Englishman were to assert "I believe that Neptune is cold" and a Frenchman that "Je crois que Neptune est froide" then we would prefer to say that they believe not the sentences "Neptune is cold" and "Neptune est froide" but some language-independent, mind-independent, abstract, symbolic representation of the meaning of the sentences. (We will continue to use sentences to represent propositions where the representation is irrelevant.) Knowledge representation (really, belief representation) in AI is concerned with devising just such representations. There is no assumption that a proposition is an expression in propositional logic or even predicate logic, e.g. `Cold(neptune)`. We will be very liberal in considering what may count as a proposition - we will include procedures ("I believe that the way to do subtraction is ..."), goals ("I believe that I want to ..."), plans ("I believe that I will ..."), maybe even attributes ("I believe that I am reliable"). (Note that these sentences in brackets are not meant to imply that agents only possess beliefs if they assert that they do!) Naturally, the representation and processing of these more complex propositions is difficult.

We can usefully distinguish between *object-level propositions* and what for the moment we will simply call non-object-level propositions. Any given ILE is concerned with some domain, say, elementary physics: true propositions contained in B_c about that domain (e.g. "momentum = mass x velocity"), which it is intended that the student acquire, are object-

level propositions. In general, B_C contains a set of object-level propositions O_C, plus non-object-level propositions of various kinds, for example

: further propositions about the domain (e.g. “momentum is often confused with impetus”) - the union of these propositions and O_C we will call the system's domain knowledge D_C;

: 'meta-level' propositions which are to some extent domain-independent (e.g. “it is best to vary one variable at a time”);

: propositions in the student model LM (e.g. “the student has fallen asleep”, “the student knows what velocity is”, etc.).

Unless the ILE’s objectives are very precisely specified, the borderline between object-level propositions and domain-related non-object-level propositons is not clear-cut. For example, if an ILE decides to help the student learn that “it is best to vary one variable at a time” then that proposition becomes an object-level one. Usually, the student model focusses on domain-related propositions - of course, this is a simplification: the student will believe non-domain-related propositions, too.

Few ILEs explicitly represent their beliefs as beliefs, that is, they do not use representations such as `Believe(computer,author(macbeth,shakespeare))`. There is no need to if the beliefs are not to be processed as beliefs - and if all expressions begin with B_C, as they have so far, then that part may be left implicit. However, we include it in our formalisation as a constant reminder of the empiricist's scepticism about the nature of knowledge and to retain the impression of a 'symmetrical' interaction between two belief-holding agents.

3. An example

Rather than present the various formal approaches in a dry, domain-independent fashion, we will use various example ILEs in order to help describe and illustrate ideas more concretely. This first ILE is as simple as possible: we have no commitment to this particular ILE or the models presented to describe it. We imagine a student using a simulation of two colliding balls. She can vary the masses and velocities of the two balls. Her aim is to predict the resultant velocity (or velocities, for elastic collisions) and thus to develop some understanding of the concept of momentum (and energy).

The student modelling problem is to build a representation of the student which may be useful for instructional interventions. (Whether such interventions are desirable is a separate issue, but few students have no difficulty even with this simple problem.) Building a student model on the basis of a student’s inputs alone is difficult, as an empirical study soon indicates. The ILE may, of course, initiate interactions specifically to clarify the content of the student model, and in general this is to be recommended - however, there are difficulties:

1. An appropriate language for such interactions needs to be devised (natural language being too vague and verbose).

2. Such interactions need to be non-intrusive. To avoid a lengthy interrogation about all possible beliefs, system questions need to be maximally informative: a student model is needed to determine such questions.

3. Students’ assertions about their beliefs are not wholly reliable. They may not fully understand the terms used (“I know what momentum is”) or may be mistaken.

Some illustrative entries in the belief-sets might be:

B_C = { "momentum = mass * velocity",
"momentum is conserved",
"mass cannot be negative",
"students often think the velocity is the average of the previous velocities",
"if you increase the masses, the velocity will decrease", ... }

LM = { "the student believes that mass cannot be negative",
"the student believes that if you halve one mass, the velocity will double",
"the student is a novice physicist",
"the student believes that velocity cannot be negative", ... }

B_{CS} = { "mass cannot be negative",
"if you halve one mass, the velocity will double",
"velocity cannot be negative", ... }

B_S = { "mass cannot be negative",
"velocity can be negative",
"energy has got something to do with where the ball is on the screen", ... }

Where we are not concerned with the content of the propositions, we may write, e.g.

$$B_S = \{p_1, p_2, p_3, \ldots\}.$$

We can, of course, express relations between propositions, thus:

$$p_6 = B_l p_3,$$
$$p_{12} = B_l \sim p_{14}.$$

4. Initialising the student model

When a student first uses the ILE LM is empty. It can be initialised in two ways: by explicit questioning or by default assumptions.

4.1 Explicit questioning

If there is a finite set of independent propositions $\{p_1, p_2, p_3, \ldots\}$ such that $B_S p_i$ may be true, we may ask the student "Do you believe p_1?" etc. Clearly this is tedious in the extreme. If the propositions are not independent, then some optimum sequence of questions may be determined. For example, if we use $K_a C$ to denote that an agent a knows a concept C if it believes all the relevant true propositions concerning that concept, then we may express a prerequisite structure in terms of a set of rules of inference (where we use italics to denote a concept):

$$B_C K_S \textit{momentum} \rightarrow B_C K_S \textit{velocity}$$
$$B_C K_S \textit{velocity} \rightarrow B_C K_S \textit{speed}, \text{ etc.}$$

Given such a set of inference rules, an optimum (or at least sensible) order of questions can be determined. In general, concepts in the middle of the structure should be asked first, since if a student says she knows that concept, the rules say she knows all the prerequisites, whereas if she doesn't then she also does not know all the concepts of which it is a prerequisite, assuming we also have the rules of the form:

$$B_C \sim K_S \textit{velocity} \rightarrow B_C \sim K_S \textit{momentum}$$

In addition to asking the student about object-level propositions, we may also ask about certain non-object-level propositions. For example, we may ask the user to assign herself

to one of a set of classes (usually called *stereotypes*): "Do you believe yourself to be an expert, novice or beginner?" Associated with each such class may be a set of inferences:

B_c"student is a novice" -> B_cK_s*velocity* & $B_c{\sim}K_s$*momentum*, etc.

4.2 Default assumptions

Normally, the stereotypes are arranged into a hierarchical structure (e.g. Figure 2), which permits some ordering of such questions to be determined and the inheritance of inferences. However, because the stereotypes are broad, the inferences they provide for the student model are generally considered to be default assumptions, liable to be over-ridden by later evidence. A set of propositions is associated with each node, representing a stereotype, such that if the student is (believed to be) a member of that stereotype then the system believes that she believes those propositions, together with those propositions attached to any encompassing stereotypes. We might also assume that a student does not believe those propositions attached to sub-stereotypes. In general, a student may be assigned to stereotypes along several dimensions, leading to the possibility of inconsistencies in the default assumptions (considered later).

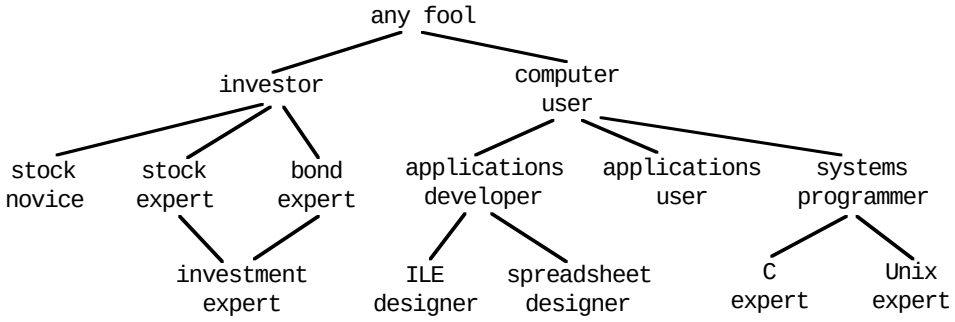

Figure 2. A stereotype hierarchy.

In the absence of information about the student, she might be assigned to the 'any fool' stereotype, such that she is assumed to believe only those things which any fool believes, that is, so-called common knowledge. Some subset of B_c might be distinguished as common knowledge and hence assigned to B_{cs}. We might also attempt to make some a-priori ascriptions of beliefs to the student. For example, the system might assume that she knows nothing of what she is to learn:

$$\forall p \; (\; p \in O_c \; \text{->} \; B_c{\sim}B_s p \;)$$

Wilks and Ballim (1987) suggest a general default belief ascription axiom:

$$B_a p \; \& \; {\sim}\exists q \; (B_a q \; \& \; q \; \text{->} \; B_a{\sim}B_b p) \; \text{->} \; B_a B_b p$$

i.e. if an agent believes a proposition p and believes there is no reason q why a second agent should not believe it then assume that it does. However, they also point out that there are several 'special cases' (e.g. secrets) where almost the opposite applies, i.e. the default is to assume that the second agent does not believe it. An ILE is an intermediate case: some but not all (otherwise the learning interaction would be redundant) of the system's beliefs are to be ascribed to the student (but we cannot know which ones).

Stereotypes are useful for initialising user models generally, especially for systems which do not anticipate much subsequent change in the content of the model (which is the case in most other user modelling contexts). But, for student modelling, where the focus is on the subsequent dynamic tracking of changes in the user (i.e. student), stereotypes are not of much use beyond the initialisation stage because they do not permit the necessary fine-grained analysis.

5. Updating the student model

5.1 Diagnosis

If the ILE encounters evidence that the current student model is inaccurate, for example, by observing that the student acts differently (e.g. when solving problems) to the way the student model would predict, then the system may try to diagnose the student model, that is, find and alter those components necessary to enable the model to correspond to observed behaviour.

5.1.1 Reconstruction

After the student model has been initialised there are two further sources of information on the basis of which the student model may be updated: (1) the student's inputs to the ILE, and (2) the current contents of the student model. In general terms, our problem is to map the inputs into a set of propositions:

$$\text{Interpret}(\{i_1, i_2, i_3, \ldots, i_n\}) \rightarrow \{p_1, p_2, p_3, \ldots, p_m\}$$

such that we can assert $B_C B_S p_j$ for $j = 1$ to m. An answer to a direct question is the simplest such interpretation:

```
Interpret({"yes, I know that force = mass x acceleration"})
                  -> {"force = mass x acceleration"}.
```

Usually, the interpretation is much more complex, for example:

```
Interpret({ student sets m1 to 3 and v1 to 6,
            student sets m1 to 2 and v1 to 9 })
                  -> {"the product m x v is a useful concept"}
```

`Interpret` has to do much more than just identify the propositions explicit in the student's inputs: it must reason about what the inputs mean in terms of the student's beliefs. Kass (1989) gives a set of 'acquisition rules' which makes default interpretations of students' inputs. Two such rules, for example, state (adapting to the student modelling context):

$$\text{Tells}(s,c,p) \rightarrow B_C B_S p \;\&\; \forall q\ (\text{Component}(q,p) \rightarrow B_C B_S q)$$

i.e. if the student states a proposition then the system believes that she believes it and all components of it;

$$\text{Problem}(c,p) \;\&\; \text{Subproblem}(q,p) \;\&\; \sim\text{Do}(s,q) \rightarrow B_C(\sim B_S(\text{Problem}(c,p)) \vee \sim B_S(\text{Subproblem}(q,p)) \vee B_S(\sim\text{Can-do}(q)))$$

i.e. if the computer sets a problem `p` for which a subproblem `q` must be solved but the student does not attempt to solve the subproblem then the computer believes that either the student does not believe that `p` is the problem or she does not believe that `q` is a subproblem or she believes that she cannot solve the subproblem. If (as is likely) the system cannot resolve this indeterminacy it might form the basis for the subsequent dialogue with the student.

Often the set of inputs is the student's solution to a problem posed by the ILE. Let us consider first the case where the solution is just the answer `A` to the problem (with no intermediate steps). The process of inferring the student's derivation of `A` is called *reconstruction*. If `A` is correct, according to D_C, we might assume that

$$\{p_1, p_2, p_3, \ldots, p_m\} \rightarrow A \text{ and } p_i \in D_C \text{ for } i = 1 \text{ to } m$$
$$\rightarrow B_C B_S p_i \text{ for } i = 1 \text{ to } m$$

provided that there is a unique such set $\{p_i\}$. Of course, this is only an assumption because the student may have used beliefs not in D_c. Normally, however, there is no such unique set.

If there are several such sets serving as potential explanations of the student's answer we may attempt to disambiguate them by using intermediate steps and the technique of *model tracing*. If we have some intermediate steps, our problem is to

$$\text{Interpret}(\{step_1, step_2, step_3, \ldots, A\})$$

given that we have several ordered sets $\{p_i\}$ such that

$$\{p_1, p_2, p_3, \ldots, p_m\} \rightarrow A$$

Usually here each p_i is a rule in a production system - the application of the rules transforms the problem P into the answer A. The sequence of applications yields a sequence of sub-problems $P_1, P_2, P_3, \ldots$ For each of the candidate sets $\{p_i\}$ we can compare the system-derived sequence $\{P_i\}$ with the student-input sequence $\{step_i\}$. In general, there will be no exact match - for one thing, the lengths of the two sequences may not be equal. The lengths can be forced to be equal (as in GREATERP (Reiser, Anderson and Farrell, 1985)) by:

1. Ensuring that the grain of each rule is such that it corresponds to a single problem-solving step, e.g. in GREATERP a rule determines the next Lisp symbol to be input, and

2. Requiring the student to specify the corresponding step (which raises pedagogical issues of no concern here).

It is not necessary to wait for the complete input sequence to be available before beginning this analysis. At each intermediate stage, the ILE may determine the potentially relevant rules and hence the potential next steps. These predicted steps can be compared to the student's input step: non-corresponding steps can be eliminated from the search (ideally there should be only one step, and hence only one rule, left). This 'model tracing' process yields no computational benefit unless the student makes mistakes, as considered below. (Here we are not comparing with the attempted analysis of solutions where the sequence information has been lost, such as a completed Lisp program.)

The outcome of this process, if it is successful, is the addition to the student model of a set of propositions of the form $B_cB_sp_i$ where each $p_i \in D_c$. If:

$$\forall i \; (B_cB_sp_i \rightarrow (p_i \in D_c))$$

the student model is called an *overlay* .

In an overlay the propositions p_i act as the unjustified *basic beliefs* of a *foundation theory* of knowledge (Pollack, 1986). Indeed, in an overlay there are *only* such basic beliefs. For example, if

$$D_c = \{ a\rightarrow b, b\rightarrow c, c\rightarrow d, \ldots \}$$

then

$$\text{Interpret}(\{\text{“a->d”}\})$$

might yield $\{a\rightarrow b, b\rightarrow c, c\rightarrow d\}$ but it would not, by the process as described, result in $B_cB_s(a\rightarrow d)$, or indeed $B_cB_s(a\rightarrow c)$ or $B_cB_s(b\rightarrow d)$, being added to the student model. Thus the student model would contain no derived beliefs and hence, of course, no derivations of them. However, if we were to consider D_c to contain not only the explicitly mentioned propositions but also any proposition derivable from them, then we could extend the interpretation to yield also any derived propositions (and their derivations). The implicit axiom

$$B_a p \;\&\; B_a(p \rightarrow q) \;\rightarrow B_a q$$

may be safe for the system, which we may desire to be a rational agent, but perhaps not for the student, as we consider further below.

So far, we have assumed that a proposition p which the system now believes the student believes previously existed only in D_c. But if the student model is not empty, then we may already have $B_c B_s p$. Reasserting the same proposition does no harm, but the question arises as to whether the system should attempt to reconstruct the student's derivation of A from the student model or from its own domain knowledge. Let us define the student's domain knowledge D_s by:

$$D_s = \{ p \mid B_c B_s p \;\&\; p \in D_c \}$$

If the system is able to reconstruct A using D_s then the system may believe that the student already believes the necessary propositions (but of course the student may have been guessing) and hence that there is no need to update the student model.

5.1.2 Cognitive diagnosis

However, if the student's answer is incorrect, i.e. A cannot be derived from D_c, and if the student model is an overlay then one or more of the propositions necessary (according to D_c) to solve the problem posed by the ILE must be missing from D_s. The general problem of reconciling the student's answer A_s with the computer system's (correct) answer A_c, derived from D_c using propositions $p_1, p_2, p_3, ..., p_m$, is an instance of *diagnosis*. Indeed, some authors use the term 'cognitive diagnosis' interchangeably with 'student modelling'.

Reiter (1987) has developed a general theory of diagnosis from first principles, that is, by reasoning not dependent on domain-dependent heuristics representing compiled experience. The system (e.g. an electrical circuit, or a medical patient) to be diagnosed is described by means of a set SD of axioms defining the behaviour of the components of the system if they are not faulty, e.g.

$$\text{and-gate}(x) \;\&\; \sim\text{faulty}(x) \rightarrow \text{out}(x) = \text{in1}(x) \text{ and } \text{in2}(x)$$

i.e. if an and-gate is not faulty its output is the conjunction of its inputs. If all the components are not faulty the axioms determine the expected behaviour of the system, i.e.

$$SD \;\&\; \sim\text{faulty}(c_1) \;\&\; \sim\text{faulty}(c_2) \;\ldots \rightarrow \text{expected behaviour}$$

where $c_1, c_2, ..$ are the components of the system. Diagnosis begins if the observed behaviour OBS differs from the expected behaviour.

Diagnosis involves retracting one or more assumptions that a component is not faulty to restore the consistency between expected and observed behaviour. Clearly we would tend to prefer a diagnosis which conjectures that some minimal set of components is faulty. Thus, a diagnosis D is a minimal set such that

$$D \sim \{c_1, c_2, ..\} \;\&$$

$$SD \cup OBS \cup (\text{faulty}(c) \mid c \in D) \cup (\sim\text{faulty}(c) \mid c \in \{c_1, c_2, ..\} - D)$$

is consistent. Re-expressing this in our previous notation, we have a cognitive diagnosis D being a minimal set such that

$$D \sim \{p_1, p_2, ..\} \;\&$$

$$D_c \cup A_s \cup (\text{faulty}(p) \mid p \in D) \cup (\sim\text{faulty}(p) \mid p \in \{p_1, p_2, ..\} - D)$$

is consistent, where

$$D_c \ \&\ \sim faulty(p_1)\ \&\ \sim faulty(p_2)\ \ldots\ \rightarrow A_c$$

As a result of such a diagnosis, the ILE may assert $B_c \sim B_s p$ for each $p \in D$.

To determine a diagnosis we may systematically postulate that each component (proposition or rule) in turn is faulty, and then that each pair of components is faulty, and so on. Obviously, this method is too inefficient for systems with large numbers of components when a number of them may be faulty. Reiter (1987) gives a more efficient algorithm, later modified by Greiner, Smith and Wilkerson (1990), for computing diagnoses. In the student modelling context, however, we may have little confidence in a diagnosis that postulated that several components were faulty, and hence a simple generate-and-test algorithm may be adequate. Huang, McCalla, Greer and Neufeld (1991) describe an application of Reiter's diagnosis procedure to student modelling.

In fact, a rather similar procedure to Reiter's had been proposed by Young and O'Shea (1981) in their production system analysis of subtraction. The postulate $faulty(c_i)$ effectively disables the definition of the component c_i from SD: similarly, Young and O'Shea suggested that an ILE could diagnose many of a student's subtraction mistakes by removing one or more rules from the system describing the component sub-procedures of a correct subtraction algorithm. Thus, the rules of their production system:

```
CM: processcolumn  -> compare, finddiff, nextcolumn.

B2a: S>M  -> borrow.

etc.
```

can be re-expressed as 'component axioms' in Prolog, e.g.

```
rule(cm):- not faulty(cm), processcolumn, compare, finddiff, nextcolumn.

rule(b2a):- not faulty(b2a), gr(S,M), borrow.

etc.
```

Running the rules (using an interpreter with the required conflict resolution strategies) with no assertions that a rule is faulty gives the expected (correct) output, because of Prolog's closed-world assumption. But if we assert, for example, `faulty(b2a)`, then the output is not as expected because `rule(b2a)` no longer fires.

Deleting a rule (for example, the one which decrements the preceding digit when borrowing) from a production system does sometimes produce output behaviour which corresponds to standard students' mistakes. But sometimes it is known that when a component is faulty then it doesn't merely not work at all but it often works in some other predictable way. So in addition to deleting a rule by assuming it faulty we may add further rules which may correspond to the faulty behaviour (as suggested by Young and O'Shea). For example, we might add the rule:

```
rule(cm):- faulty(cm), processcolumn, finddiff, nextcolumn.
```

which says that a faulty version of the `cm` rule omits to carry out the `compare` operation (leading to the common mistake that 46 - 29 = 23). We could add a number of such rules to correspond to known faulty versions of the 'correct' rule.

The status of such faulty rules (or, more generally, beliefs) in the ILE needs to be carefully described. If `f` is such a rule, we cannot say $B_c f$ (since the computer system does not believe it to be true) nor $B_c B_s f$ (since the system cannot assume that the student believes it). Rather, we can assert $B_c \sim f$, and, more usefully, that the system believes that a typical hypothetical student may believe it, which we will denote by $B_c B_h f$. We will define a *fault* to be a proposition `f` such that $B_c \sim f\ \&\ B_c B_h f$. Faults are sometimes called *bugs* or *mal-rules* (if the proposition is expressed as a production rule). A set of faults $\{f_1, f_2, ..\}$ is sometimes called a *bug catalogue*. It is common to extend the definition of the student's

domain knowledge D_s to include propositions f such that $B_c B_s f$ & $f \in$ bug-catalogue, although strictly 'knowledge' here is a misnomer.

A *fault-diagnosis* F is a minimal set of paired terms {<p, f>} such that

$$p \in \{p_1, p_2, ..\} \&$$
$$f \in \{f_1, f_2, ..\} \&$$
$$D_c \cup A_s \cup (\text{faulty}(p_i) \& f_i \mid p_i \in D) \cup$$
$$(\sim\text{faulty}(p) \mid p \varepsilon \{p_1, p_2, ..\} - D)$$

is consistent, where

$$D_c \& \sim\text{faulty}(p_1) \& \sim\text{faulty}(p_2) \ldots \rightarrow A_c$$

that is, a fault-diagnosis is a minimal set such that if the description of a component p_i is replaced by that of an associated faulty component f_i, then the system may derive the student's answer. It is conjectured that a fault-diagnosis (which describes *how* a component is faulty) provides more pedagogical leverage than a diagnosis (which says *that* a component is faulty) - this conjecture is considered below.

A revised diagnostic procedure for an ILE might be as follows: first, attempt to derive A_s using D_c and D_s - if this is successful, then the system may postulate that the student believes those propositions used in the derivation; if it fails, then use a diagnosis procedure (such as Reiter's algorithm) to isolate missing or faulty components; then, for those components isolated, use a fault-diagnosis procedure to determine associated faults. An exhaustive fault-diagnosis procedure, in which each faulty component is systematically replaced by a member of the set of faults, is clearly not feasible in general. Instead, all ILEs which make use of a set of faults contain explicit pre-determined links between a component and its associated faults. Thus, identifying the faulty component leads directly to a small set of potential faults which can be exhaustively searched. In the case of GREATERP's model tracing algorithm these steps are combined: the propositions (or rules) such that $B_c p$ or $B_c B_h f$ are merged and this merged set is used to make predictions about the student's next step - some correct, corresponding to the $B_c p$ propositions, some faulty, corresponding to the $B_c B_h f$ propositions.

5.1.3 Generative mechanisms

Prespecifying a set of faults is a laborious process (there are hundreds of such faults in GREATERP). Moreover, most of the faults are irrelevant for any particular student. It may be more efficient to generate them as needed, if it is possible. A few such generative techniques have been proposed:

1. Syntactic transformations of a ‘correct’ proposition, e.g. by removing parts of a production rule (so that it sometimes applies when it should not, or sometimes does not do all that it should), or by replacing constants by variables (or vice versa) to make the proposition more general or more specific. The number of such transformations is clearly very large.

2. The use of ‘meta-rules’, i.e. rules which define likely faulty transformations of other rules, for example, in language learning a meta-rule which says that it is common to overlook gender agreement permits the generation of a number of specific faults.

3. The use of ‘repairs’ to overcome impasses during problem-solving. If when using LM to model the student’s attempt to solve a problem, the interpreter cannot proceed (because no rules apply), then a simple, local patch may be attempted, e.g. to skip a step, back up to a previous point, or find an analogous operation. The cognitive issues involved in such repairs are thoroughly discussed in van Lehn (1989). Technically, two

complications may be pointed out. First, the interpreter needs to be able to detect an impasse - a non-trivial problem - and enter a repair-generating phase, and is thus not the normal problem-solving interpreter. Secondly, when an impasse is enountered it is rarely clear where the repair should be attempted. When parsing a typical incorrect foreign language sentence literally hundreds of apparent impasses are met, where a parser will back up to try to find alternatives - any one of these impasses may indicate the real reason for the eventual unsuccessful parse.

4. The inference of a fault to bridge the gap in an analysis of a problem solution. If when using LM the system can reason from the problem to a point q and from the answer backwards to a point q' and the gap between q and q' is small, then the system may hypothesise a rule to transform q into q'. In general, however, there will be a large search tree and hence a large number of such gaps - it will not be obvious which gap should be filled.

It is somewhat strange that researchers have emphasised the use of generative procedures to generate faults without emphasising that these procedures do not just generate faults. Occasionally they generate correct beliefs. Indeed, they all correspond to perfectly reasonable learning procedures, which inevitably (because of the complexity of what is to be learned) lead to faults from time to time.

Thus, in order to formalise procedures for generating beliefs (whether faulty or not) we may look to the large body of machine learning research. For the moment, we will consider the implications for student modelling.

Instead of labelling a belief as categorically correct or faulty (as we have so far), we may begin to recognise that what is considered faulty depends on the context. In a certain context, a belief may be considered 'applicable', that is, correct for the purposes of this context but perhaps faulty in another one. Thus, instead of implicitly viewing the situation as one in which the ILE aims to move a student from a 'faulty' context to a 'correct' one, we may consider that there is a sequence of contexts and that the applicable rules in one context may form the 'faulty' rules of another context. For example, in learning French, we might have the following rules:

1. the possessive pronoun agrees with the gender of the possessor -> “son table” (for “his table”): this rule may have been acquired by transfer from English.

2. the possessive pronoun agrees with the gender of the thing possessed -> “sa table”, “sa adresse” (for “his/her table”, etc.).

3. as above except when the following noun begins with a vowel -> “son adresse” (for “her address”).

4. as 2 except when the following noun begins with a vowel or a mute “h” -> “son horloge” (for “her clock”), and, in principle, so on.

In any context, a faulty rule may be accepted as correct, at least temporarily. While teachers often give students 'rules' of language they are rarely strictly correct. In most domains, in fact, there are degrees of correctness rather than the clear-cut correct-incorrect division of typical ILE domains (subtraction, Lisp programming, etc.), and in some domains (e.g. economics) there may be considerable disagreement about what the 'correct' rules are.

Of course, the previous view may be regarded as just one snapshot from the new view: at any instant the student is in context i and the system is in context (i+1), and indeed many ILEs function by helping a student move through increasingly complex contexts. However, several refinements to our view of ILE student modelling are now possible:

1. There may be more than one sequence of contexts which leads to the 'target context'. Maybe these sequences can be dynamically generated rather than pre-ordained by the system.

2. There may be no 'target context', that is, the system may seek to help a student move from a given context but not necessarily towards a target known to the system The building and use of a student model in such a situation is obviously a more subtle process.

3. Even if there is a target context, we are reminded that most knowledge is not categorically correct or not but appropriate or not for a context - this may be reflected in the style of interaction adopted by the ILE.

4. The same representation may not be appropriate for beliefs in different contexts, for example, in one context beliefs may be semi-qualitative, in other quantitative. Therefore, there is a potential problem in relating beliefs in different contexts.

5.2 Revising beliefs

If we now regard an ILE as seeking to cause a student to revise her beliefs (possibly toward some target beliefs), rather than aiming to replace a faulty belief by a correct one, then we need to consider when and how a student revises her belief-set, because the student model will need to be revised similarly. In general terms, a student may revise her belief-set if she perceives it to lead to inconsistencies, or if it appears to be inefficient or inadequate in some respect. The former situation tends to lead to the discarding of beliefs, the latter to the creation of new beliefs.

5.2.1 Discarding beliefs

Imagine that D_s contains the following propositions:

p_1 "Fire rises."
p_2 "The higher regions of the universe are more fiery than lower ones."
p_3 "Fire produces light."
p_4 "The sun produces more light than the stars."
p_5 "The sun is more fiery than the stars."
p_6 "The sun is higher than the stars."

and the student is made aware of a new piece of evidence:

p_7 "The stars are higher than the sun."

If the new proposition were simply added to D_s then the student modelling process would be complicated by the contradiction between p_6 and p_7, which under normal inference rules would enable any proposition at all to be inferred. One solution is to discard one or more propositions from the set $\{p_1, .., p_7\}$.

There are two different bases for deciding which proposition(s) to discard. First, we must recall that D_s is derived from the system's beliefs about the student's beliefs. The system may be mistaken in its beliefs about the student. Therefore the system may analyse its reasons for asserting $B_c B_s p_i$ and decide, for example, to discard the 'weakest' default assumption.

Secondly, the system may consider D_s to be an accurate description of the student's beliefs and must then concern itself with how the student would actually resolve the conflict. There are several possibilities:

1. The student may continue to believe all seven propositions (as Anaximander (550 BC) apparently did in this situation). She may be able to reason with inconsistent belief-sets using methods different to those of classical logic, as considered below.

2. She may disbelieve the new proposition: there is a natural reluctance to overthrow an existing belief-set on scanty evidence, especially if no simple modification of the existing belief-set overcomes the problem.

3. She may discard one or more old propositions. In order to determine which propositions to discard, she (and the system) may analyse the reasons why they are held. To permit this, we distinguish between *basic beliefs* (or premises or assumptions or hypotheses) and *derived beliefs*. In the above example, p_1, p_3, p_4 and p_7 may be basic beliefs and p_2, p_5 and p_6 derived beliefs. Those beliefs from which a derived belief is derived constitute a *justification* for that belief. For example, $\{p_2, p_5\}$ may be a justification of p_6. In general (but not here), there may be several justifications for the same derived belief. We denote the justifications $j_1, j_2, j_3, \ldots$ for a proposition p as follows:

$$p: \{j_1, j_2, j_3, \ldots\}$$

For example:

$$p_6: \{\{p_2, p_5\}\}$$

$$p_5: \{\{p_3, p_4\}\}$$

$$p_2: \{\{p_1\}\}$$

For completeness, we may consider a basic belief to be its own justification, e.g.:

$$p_1: \{\{p_1\}\}$$

A justification is thus in terms of the propositions from which a belief is immediately derived. In assumption-based approaches to belief revision, the system maintains a list of the basic beliefs upon which the belief ultimately depends - in this case, we would have:

$$p_6: \{\{p_1, p_3, p_4\}\}$$

$p_5: \{\{p_3, p_4\}\}$, etc.

Discarding a derived belief (e.g. p_6) may overcome the immediate problem but if the basic beliefs from which it was derived remain then the discarded belief may be re-derived. To prevent this, we must discard one or more basic beliefs from its justification (in this case, one or more members of the set $\{p_1, p_3, p_4\}$). In general, when there is a set J of justifications, we must find a hitting set, that is, a set of propositions that contains at least one element of each set in J. If we then discard the propositions in the hitting set then no justifications for the original belief remain. In general, there will be a number of such hitting sets and we would aim to select that which is in some sense minimal, i.e. causes the least disruption to the existing belief-set.

Once a basic belief is discarded, we may also discard any derived belief which is justified only by that basic belief. With a foundation theory of knowledge, beliefs no longer justified are abandoned; with a coherence theory, beliefs are retained in the absence of any challenge to them. Work on this and associated problems in AI goes under the name of *belief revision* (although really it is belief-sets that are revised, beliefs being merely discarded). Psychological studies are relevant only to the second of the two bases mentioned above for discarding beliefs, modelling belief revision by the student. In a student modelling context (as in fact in most belief revision work), it may not be possible or necessary for the system alone to select which proposition(s) to discard: rather, the set of potential amendments may be used to focus subsequent clarification dialogues with the student.

As we have seen, the basic requirement for any belief-discarding scheme is that whenever a new belief is added to a belief-set the system records how that belief depends upon other beliefs. Various refinements to the basic scheme have been investigated (and some of them will be discussed in later sections):

1. The use of non-classical logics - it is not essential that the dependencies between beliefs correspond to standard logical implication.

2. The recording of a set of basic beliefs as inconsistent once they have been found to be so - to avoid subsequent consideration or to permit the system to be aware that the set is inconsistent, if it is considered later.

3. The use of non-monotonic reasoning, by recording with each proposition not only the justifications for believing it but also any propositions that have to be disbelieved in order for the proposition to be believed.

4. The use of *labels* to mark a proposition as disbelieved rather than the erasing of it - this improves efficiency if the system subsequently wishes to re-consider its disbeliefs, and also enables the system to work within limited contexts.

5.2.2 Creating beliefs through reasoning

An agent may derive further beliefs by reasoning about the propositions in a belief-set. The reasoning processes are described or defined by a *logic* L, expressed as a set of axiom schemata. (The word 'logic' is not meant to imply that reasoning is necessarily sound or complete.) Student modelling is difficult partly because a number of different kinds of logic are potentially relevant. Here are a few illustrations:

L_1 = { B(p v q) & B(~p v r) -> B(q v r) }

This logic, with a single axiom schema (the rule of inference called resolution, which is known to be sound and complete in predicate logic), might be used by the system to reason about its domain knowledge, for example, to determine whether the student's answer accords with the system's knowledge. A standard theorem-prover or a language such as Prolog could be used for this purpose.

L_2 = { B(p & q) -> Bp ,
B(p -> q) & Bp -> Bq ,
B(p -> q) & B(q -> p) -> B(p = q) , ... }

This might be intended to define a logic of 'natural deduction'. L_2 might be more useful than L_1 when the system is carrying out the process of reconstruction (section 5.1.1), since the intermediate steps passed through when using L_2 might correspond better with the steps which a rational student might pass through. L_1 is a computationally oriented logic; L_2 is intended to have some psychological validity.

L_3 = { B(p->q) & Bq -> Bp ,
B(p->q) & B~p -> B~q , ... } $\cup$ L_2

Whereas L_2 might be intended to describe the reasoning of an ideal rational student, L_3 might be intended to describe that of an actual student, since it contains some 'irrational' schemata in addition to the axiom schemata of L_2. Since real students may have faulty reasoning schemata, the process of reconstruction may perhaps be better carried out with L_3 than with L_2. Also, of course, the student may be lacking some of the 'ideal rational student's' schemata. In other words, the same kinds of issue as discussed in section 5.1 with respect to object-level propositions arise also at the level of reasoning schemata. If it

is necessary for the system to handle students' difficulties at the reasoning level, then explicit axiom schemata need to be provided.

L_4 = { B(p,α) & B(q,α) -> B(p & q,α) ,
B(p,α) & B(q,α ∪ β) -> B(p->q,β) , ... }

This example illustrates that the logic of our axiom schemata does not have to be classical logic with its standard notion of validity. This particular case (*relevance logic* (Anderson and Belnap (1975)) denies the so-called paradoxes of implication: p->(q->p) and (p & ~p)->q. Instead, in relevance logic, one proposition entails another only if there is an element of causality that relevantly connects them. Each wff p is associated with an *origin set* α containing all the basic beliefs used in its derivation. This is written p,α. The first axiom schema in L_4 says that if p and q are wffs with the same origin set, then we can deduce p & q and associate it with the same origin set. Relevance logic has been used in work on belief revision (Martins and Shapiro, 1988) and is presumably partly motivated by a feeling that human reasoning follows such schemata rather than those of classical logic.

L_5 = { B(goal(x)) & B(precondition(x,y))
-> B(must-satisfy(y)) ,
B(goal(x)) & B(precondition(x,y)) & B(can't-satisfy(y))
-> B(can't-do(x)) , ... }

L_5 indicates that our logics may include 'pragmatic reasoning schemata' (Holland, Holyoak, Nisbett and Thagard, 1989). Such schemata are intermediate between domain-specific rules and the abstract rules of standard formal logic (as illustrated above). Pragmatic reasoning schemata are abstract rules in that they apply to a wide range of content domains but they are constrained by particular inferential goals and event relationships. The extent to which students use such pragmatic schemata is a matter of debate.

L_6 = { $K_c p$ & Tells(c,s,p) -> $B_c B_s p$,
$p \in O_c$ -> $B_c \sim B_s p$,
...
$B_c B_s$(p->q) & $B_c B_s \sim p$ -> $B_c B_s \sim q$,
$B_c B_s$(p->q) & $B_c B_s q$ -> $B_c B_s p$, ... }

The logics L_1 to L_5 all considered the case of a single agent deriving new beliefs. In student modelling, however, we are deriving nested beliefs of the form $B_c B_s p$. The appropriate axiom schemata here will be of two forms: one defining how the system reasons about instructional events and its own beliefs and the other defining how the system believes the student reasons about her beliefs. Thus, in L_6 the earlier schemata are of the first form and the later schemata of the second form. For example, the first schema says that if the system knows a proposition and tells it to the student then the system believes the student believes that proposition. The last schema given in L_6 denotes that the system believes the student reasons using an axiom schema from L_3 (of course, we could also use schemata from L_4, L_5 and other logics).

It is clearly difficult to define a satisfactory set of such schemata for student modelling purposes and indeed no existing ILEs make use of explicit schemata of this kind. However, they do use ad-hoc, implicit schemata, for otherwise the student model would never be updated. Advantages that might follow from making them explicit include:

1. As with any formalisation, it might be easier to understand and analyse the processes being formalised.

2. As we have seen, belief revision is facilitated by recording the justifications for beliefs. But merely listing the justifications, as in p_c: {{p_1 ... p_2 ... p_4}} does not enable the

system to reason about the validity of the justifications. If we also recorded the axiom schemata which enabled the belief to be derived the system could, for example, take account of faulty derivations.

3. With explicit reasoning schemata, the system may be able to discuss the reasoning processes themselves, and thus move beyond domain-related issues, as advocated by many educationalists.

4. We may customise the reasoning process, i.e. we may adopt different logics for different students, or for the same student at different times. Thus, we have additional scope for individualising instructional interactions.

5. By identifying distinct reasoning schemata appropriate for different aspects of the student modelling problem, we may be able to separate computational and cognitive issues which are currently inter-mingled. Student modelling may be "unabashedly psychological" (Clancey, 1986) but, as stated earlier, the primary aim is computational utility, not cognitive validity. By isolating where cognitive issues are important, we may develop computational frameworks which are independent of them.

5.2.3 Limited reasoning

Defining a set of schemata does not define how those schemata will be interpreted by the system. For example, for L_1, the standard rule of resolution, a large number of theorem-proving strategies have been devised. The simplest mechanism - that all axiom schemata are repeatedly applied until no further conclusions can be drawn - is inadequate computationally (in general, it would take much too long), psychologically (human agents do not normally draw all possible conclusions from their beliefs) and philosophically (it seems strange to say that an agent believes a proposition if it takes it ten minutes of intense reasoning before it avers that it does). Therefore, the system's interpreter of axiom schemata will carry out some 'limited reasoning' process. However, we ought not to bury those limitations in the interpreter but to make them explicit so that they too may become a possible focus for student modelling.

The starting point for discussions of limited reasoning mechanisms for knowledge and belief is the *possible world semantics* of modal logic (Hintikka, 1962). The intuitive idea is that besides the true state of affairs there are a number of other possible states of affair, or possible worlds. The worlds are connected by an accessibility relation `R` which may be defined to satisfy various constraints, e.g. it may be transitive, i.e. `uRv` and `vRw` implies `uRw`, where `u`, `v` and `w` are worlds. In each world, a proposition is given a truth value. An agent in a world `w` is said to believe a proposition if it is true in all worlds accessible to `w`.

The modal logic based on a transitive accessibility relation (called weak S4) can be given a sound and complete proof theory comprising the following rules of inference and axioms:

R_1	Necessity	`p => Bp`
R_2	Modus ponens	`p & p->q => q`
A_1	Tautologies	`p`, where `p` is valid in propositional logic
A_2	Distribution	`Bp & B(p->q) -> Bq`
A_3	Positive introspection	`Bp -> BBp`

Other logics, perhaps less suitable for representing belief, have different accessibility relations and use one or more of the following axioms:

A_4	Knowledge	`Bp -> p`
A_5	Negative introspection	`~Bp -> B(~Bp)`
A_6	Consistency	`Bp -> ~B~p`

Any modal logic which includes A_1 and A_2 (as does every modal logic using the possible worlds approach) suffers from the problem of *logical omniscience*, that is, the agent believes all tautologies and all implications of its beliefs. This is considered a problem for the reasons given above - that it is computationally intractable and psychologically implausible. There are basically two ways in which the problem may be overcome and hence some element of limited reasoning introduced: we may adopt a 'semantic approach' in which we use a modified notion of truth compared to the classical one used in possible world semantics; or we may follow a 'syntactic approach' in which beliefs are sentences in some syntactically specified set and sentences are distinguished by syntax (thus, for example, we may have `B(p v q)` and yet not necessarily `B(q v p)`).

Levesque (1984) adopted the former approach in developing the distinction between *explicit belief* and *implicit belief*. We use `Ip` to denote that an agent implicitly believes a proposition - i.e., that it is a logical consequence of its explicit beliefs - and `Ep` to denote that it explicitly believes it. Informally, the explicit beliefs are intended to be that subset of the implicit beliefs which the agent considers relevant or which have been 'activated' within the agent. Formally, implicit belief may be modelled by classical possible world semantics but explicit belief requires a modified version. Levesque used the idea of a *situation*, in which a proposition may be true, false, both or neither. A complete, coherent situation - i.e. one in which propositions are true or false - corresponds to the standard possible world, but we may also have a 'partial world' in which a proposition may be neither true nor false and an 'incoherent world' in which it may be both.

By specifying a semantics similar to that for relevance logic, Levesque showed that, while implicit belief retains the properties of belief in possible world semantics, explicit belief does not suffer from the problem of logical omniscience. For example, all the following formulas are satisfiable:

```
1.  Ep & E(p->q) & ~Eq
2.  Ep & ~E(p & (q v ~q))
3.  ~E(p v ~p)
4.  Ep & E~p
5.  E(p & ~p)
```

The properties of explicit belief follow from the incoherence and incompleteness introduced in situations - the former leads to the possibility of believing unsatisfiable propositions (e.g. 5 above), the latter to the possible lack of belief of valid propositions (e.g. 3 above).

Thus, it is possible to define formal logics to express some aspects of the inconsistencies and incompletenesses which students display. However, many subtle issues remain. For example, if reasoning is considered to be carried out with respect to the situations thought possible by the agent, is it reasonable to allow incoherent situations as being possible? Also, the effect of imperfect reasoning in a classical logic is achieved by assuming perfect reasoning in a non-classical logic (relevance logic). Moreover, the formal logic of explicit belief does not cover the expression of nested, multi-agent beliefs which we have seen that we need for student modelling.

The syntactic approach to overcoming the limitations of possible world modal logics emphasises the role of syntactic form in determining the truth of a belief. However, the semantics of belief logics must differ from those of classical logic because, unlike the ordinary logical operators, the modal operators of belief are referentially opaque, i.e. if `p` is equivalent to `q` then we cannot substitute `q` for `p` in any expression within the scope of `B` (for example, if `largest-planet = neptune`, then we cannot infer `B(Cold(largest-planet))` from `B(Cold(neptune))`). Instead, we may define a sentential semantics in which, for each agent `i`, we associate a belief-set B_i and a logic (a set of inference rules) L_i. The theory formed by the closure of B_i under L_i is denoted by T_i. A proposition $p \in T_i$ if and only if `p` is provable in `i`'s theory using `i`'s inference rules. Then $B_i p$ has the value true if and only if `p` is in the theory associated with `i`. (This semantics is referentially

opaque, as desired, because an equivalent (in our theory) expression may not be in the agent's theory.)

Fagin and Halpern (1987) attempted to isolate the advantages given by Levesque's notion of incomplete situations by defining a syntactic *awareness* function. The intuition is that an agent cannot believe a proposition if it is not aware of it, and that we might say that an agent is aware of p if it explicitly believes p or its negation:

$$Ap \equiv E(p \vee \sim p)$$

Instead of using incoherent or partial worlds, Fagin and Halpern use standard possible worlds with the awareness function to filter out those formulae of which the agent is unaware. A world w supports the truth of Ep if all the worlds the agent considers possible in w support the truth of p relative to the set of primitive propositions of which the agent is aware in world w. Implicit belief is as before and explicit belief is similar to Levesque's except that (1) an agent's set of explicit beliefs is closed under implication and (2) an agent cannot hold inconsistent beliefs, e.g. E(p & ~p) is not satisfiable.

Consider the following belief-set:

$$E_C(\text{Foreigner}(s))$$
$$E_C(\text{Foreigner}(x) \rightarrow \sim A_X(\text{sconce}))$$
$$E_C(\text{Prerequisite}(\text{sconce},\text{college-etiquette}))$$
$$E_C(\sim A_S(c) \wedge \text{Prerequisite}(c,g) \rightarrow \sim \text{Infer}(s,g))$$

i.e. the system believes the student to be a foreigner, that all foreigners are unaware of the concept of a sconce (a fine imposed at Oxbridge), which is a prerequisite for understanding Oxbridge college dining etiquette, and that a student cannot infer propositions if she is unaware of a prerequisite concept. In such a case the system cannot show (for example) that the student can infer (college-etiquette v ~college-etiquette) even though the tautology follows from the system's facts (indeed, from any facts) because s, a foreigner, is unaware of a prerequisite concept (this example is considered further below).

A major benefit, for student modelling purposes, offered by this logic of awareness is that, as we see above, it allows nested beliefs, which Levesque's logic of implicit belief does not. From the definitions, we can derive various relationships between implicit belief, explicit belief and awareness, e.g.

$$E_C E_S(p \vee q) \equiv (A_C p \wedge I_C(A_S p \wedge I_S p))$$

Nested multi-agent beliefs enable us to describe the system's reasoning about the student's beliefs, and nested single-agent beliefs provide us with a basic notation for discussing self-reflection and metacognitive processes (as pursued in section 5.3.4).

Fagin and Halpern's logic of *general awareness* goes further in defining an essentially syntactic operator A_i (for 'agent i is aware of') in addition to E_i and I_i which is not limited to primitive propositions, as the previous logic of awareness is. Since an agent explicitly believes a proposition if it implicitly believes it and it is aware of it, we have:

$$E_i p \equiv I_i p \wedge A_i p$$

Thus, explicit belief retains many of the properties of implicit belief, relativised to awareness. For example, the rules and axioms R_1, A_2 and A_3 in the weak S4 logic become:

R_1	Necessity	p => (Ap->Ep)
A_2	Distribution	Ep & E(p->q) & Aq -> Eq
A_3	Positive introspection	Ep & AEp -> EEp

To these general axioms, we might wish to add restrictions to provide desired properties of the logic - for example, we might specify that an agent `i` is unaware of (any proposition that mentioned) agent `j`, or that an agent is aware only of a certain subset of primitive propositions.

This last restriction can be elaborated to provide a logic of *local reasoning* which differs from the logic of general awareness in that it enables an agent to hold inconsistent beliefs. The idea is that an agent's belief-set may be partitioned into a set of non-interacting clusters such that any cluster is internally consistent but may contradict a different one. In this logic we use $E_i p$ to denote that agent `i` explicitly locally believes `p`, i.e. believes `p` in some 'frame of mind', and $I_i p$ to denote that agent `i` implicitly believes `p`, i.e. believes `p` if all its frames of mind are pooled.

This version of implicit belief satisfies axioms A_1 and A_2 of weak S4 but not axiom A_3, and the version of explicit belief is not closed under implication and hence not subject to the problem of logical omniscience. The formula $E_i p \ \& \ E_i(p \rightarrow q) \rightarrow E_i q$ is satisfiable because `i` might believe `p` in one frame of mind and `p->q` in another but never be in a frame of mind where it puts these facts together. Moreover, an agent may hold inconsistent beliefs, i.e. $E_i p \ \& \ E_i \sim p$, because it might believe `p` and `~p` in different frames of mind. However, agents do not believe in incoherent worlds, i.e. $E_i(p \ \& \ \sim p)$ is impossible. As with the logic of general awareness, we may impose conditions to capture various properties. For example, Fagin and Halpern define a narrow-minded agent to be one who when in one frame of mind refuses to admit it may occasionally be another. For such an agent $E_i(\sim(E_i p \ \& \ E_i \sim p))$ is valid even though $E_i p \ \& \ E_i \sim p$ is consistent. In addition, although the logic of local reasoning assumes an agent can do perfect reasoning within each cluster, we can add an awareness function to the structure for local reasoning to provide a model of belief which is not closed under valid implication.

It should be emphasised at this stage that it is not the aim to develop from among this great variety of limited reasoning mechanisms one which is 'correct'. This is an unattainable aim: some philosophical or computational objection can assuredly be raised against any proposed scheme. Rather, the aim is to develop a general framework within which any such scheme can be explicitly defined and theoretically analysed. The problems of student modelling, involving a limited agent (the computer system) reasoning about the beliefs of another limited agent (the student), are complex but it is no long-term solution to bury techniques within opaque algorithms.

5.2.4 Meta-reasoning

In section 5.2.2, we pointed out that an agent may use different logics (sets of axiom schemata) for reasoning about beliefs and that there are potential benefits in making those logics explicit. In section 5.2.3, it was described how different interpretations of the logics can lead to different kinds of limited reasoning. Similarly, we may expect there to be benefits from making such interpretations explicit. In other words, we are suggesting an explicit meta-logic `M` which interprets a logic `L` with respect to a belief-set `B` to derive new propositions. (In this section, we will ignore the considerations of explicitness, implicitness and awareness, discussed in the previous section.)

For example, consider:

```
B =  {  Cold(neptune) ,
        Cold(pluto) ,
        ∀x Cold(x) -> Lifeless(x) ,
        ~Lifeless(mars) ,
        ~Lifeless(earth) }

L =  {  B(p & q) -> Bp ,
        B(p -> q) & Bp -> Bq
```

```
        B(~q) & B(p -> q) -> B(~p) }

M =  {  Difficult('B(~q) & B(p -> q) -> B(~p)') ,
        ∀s ~Difficult(s) -> Easy(s) ,
        ∀s Difficult(s) -> Apply-at-most(s,1) ,
        ∀s Easy(s) -> Apply-at-most(s,3) }
```

Ignoring for the moment the considerable technical problems, the intention is that the meta-logic express that the third schema in the logic is a difficult one, that all schemata that aren't difficult are easy, and that difficult schemata are only applied once at most and easy schemata three times at most. Applying M to L and B, we obtain the derived beliefs:

```
Lifeless(neptune)
Lifeless(pluto)
~Cold(mars)
```

but not ~Cold(earth), assuming that the schemata are applied from the beginning of the belief-set.

We will refer to M as the meta-level and L and B as the base-level. The general idea, then, is that the meta-level specifies properties of the base-level logic which determine how it is interpreted with respect to a belief-set. The aim, for student modelling purposes, is to enable the system to explicitly model and reason about different aspects of the student's competence. Unless these components are declaratively specified, they cannot be dynamically changed by the system (to model changes in the student or to adapt the general framework to an individual student) and they cannot form the focus of instructional interactions.

The above example is in terms of a single agent, and thus could correspond to the system (or the student) reasoning about its (or her) own beliefs, but as we have seen (e.g. in L_6) student modelling is already at a meta-level, since it concerns the system reasoning about the student's beliefs. This 'level-shift' is one of the things that makes student modelling formally complex.

The idea of a metalanguage has been much studied in AI and in mathematics (for example, to overcome logical paradoxes). At first in AI, meta-reasoning was used to shorten proofs obtained using simple, uniform deduction strategies such as those based on resolution (our L_1 above), by for example looking at syntactic structure rather than repeatedly applying inference rules. Meta-reasoning has since been applied to many areas of AI. We will illustrate the method by two examples related to student modelling, addressing the two different kinds of axiom schemata given in L_6, i.e. those related to the system interpreting instructional events in terms of student beliefs and those related to the system reasoning about the student's reasoning.

The first example is adapted from Cialdea et al (1990), who describe a system called SEDAF to help students learn how to graph mathematical functions by solving for characteristics of the function. They describe a system belief-set:

```
Bc = {  Stationary(x,f) & Decreasing-left(x,f) &
            Increasing-right(x,f) -> Minimum(x,f) ,
        Denominator-zero(x,f) -> Pole(x,f) , ... }
```

and a system mal-rule belief-set

```
Bh = {  Decreasing-left(x,f) & Increasing-right(x,f)
             -> Minimum(x,f) ,
        Numerator-one(x,f) -> Singular(x,f) , ... }
```

The system's logic of schemata is not stated explicitly but is in fact equivalent to resolution (i.e. our L_1).

In order to link a meta-level with the logic, a meta-level predicate `Proof(B,L,p,d)` is defined which asserts that `d` is a derivation or proof of proposition `p` using logic `L` on belief-set `B`. It is assumed that `Proof` is defined by a suitable set of meta-axioms, so that if `Proof(B,L,'p','d')` is a theorem of the meta-theory then `d` is a derivation of `p` using `L` and `B`, where `'p'` and `'d'` are the representations of `p` and `d` at the meta-level (Weyhrauch, 1980). Cialdea et al then provide the following five meta-level axioms (adapted slightly):

$$
\begin{aligned}
M = \{\ & Answers(p,dontknow) \ \&\ Proof(B_c,L_1,p,d) \ \&\ q \in d \rightarrow B_c \sim B_s q\ , \\
& Answers(p,dontknow) \ \&\ Proof(B_c,L_1,not(p),d) \ \&\ q \in d \\
& \quad \rightarrow B_c \sim B_s q\ , \\
& Answers(p,yes) \ \&\ Proof(B_c,L_1,not(p),d) \ \&\ q \in d \rightarrow B_c \sim B_s q\ , \\
& Answers(p,yes) \ \&\ Proof(B_c \cup B_h,L_1,p,d) \ \&\ q \in d \\
& \quad \&\ Acceptable\text{-}proof(d) \rightarrow B_c B_s q\ , \\
& Answers(p,yes) \ \&\ Proof(B_c,L_1,p,d) \ \&\ q \in d \rightarrow B_c B_s q\ \}
\end{aligned}
$$

The first axiom specifies that if the student does not know the answer to a problem which the system can solve (using B_c and L_1) then the system believes the student does not know propositions used in the derivation of the answer. (In general, of course, she would not know one or more of those propositions - as discussed with reference to reconstruction in section 5.1.1 - and the system might begin some dialogue to work out which.) The predicate `Acceptable-proof` verifies whether it is plausible to believe that the student has constructed the derivation `d` (either by interrogating the student or by analysing the current student model). Note that the same logic L_1 is used throughout, which is not necessary formally.

These two meta-level predicates (`Proof` and `Acceptable-proof`) just ask questions about derivations in the base-level. In general, `M` can also impose restrictions (such as those discussed in the previous section) on what can be derived from `L` and `B`. The meta-level could, for example, assume that the base-level will draw the inferences it is capable of, unless the meta-level knows of constraints which prevent them being drawn. This approach is independent of the underlying base-level reasoning procedure - but to actually design such a meta-level we have to commit ourselves to a particular underlying procedure.

Van Arragon (1991) describes a system called LNT in which the underlying reasoning is based upon linear resolution, for which we need the meta-level predicate

$$Infer(B_a, b \rightarrow g)$$

as used in the 'foreigner' example of section 5.2.3, where `b->g` is a fact in B_a, where `b` is a conjunction of literals. (For clarity, we omit the quotes which are strictly necessary for the meta-level.) To reason about limitations in the base-level, the meta-level includes propositions of the form

$$\ldots \rightarrow \sim Infer(B_a, L_a, b \rightarrow g)$$

where the ... defines the conditions under which the inference `b->g` cannot be made, and where we include the logic L_a to make it clear that different such logics may be defined.

For example, the following `M` specifies that the student can infer no more than three steps:

$$M = \{ \; Infer(B_S, L_S, b1 \rightarrow g1) \; \& \; Infer(B_S, L_S, b2 \rightarrow g2) \; \& \; Infer(B_S, L_S, b3 \rightarrow g3) \; \& \; g \neq g1 \; \& \; \ldots \; \& \; g2 \neq g3 \; \rightarrow \sim Infer(B_S, L_S, b \rightarrow g) \; \}$$

Given the following belief-set:

$$B_S = \{ \; true \rightarrow p_1, \; p_1 \rightarrow p_2, \; p_2 \rightarrow p_3, \; p_3 \rightarrow p_4 \; \}$$

the system will determine that the student can infer p_1, p_2 and p_3, but not p_4 (since although p_4 follows from B_S it requires four instances of $Infer(B_S, L_S, b \rightarrow g)$). Thus, the system may handle B_Ss which are potentially inconsistent - for example, if we had:

$$B_S = \{ \; true \rightarrow p_1, \; p_1 \rightarrow p_2, \; p_2 \rightarrow p_3, \; p_3 \rightarrow p_4, \; \sim p_4 \; \}$$

then the system could, using M, consider that the student's reasoning is too limited to realize the inconsistency.

This was the basic idea used in the foreigner example to handle a more interesting case of limited reasoning, that of lack of awareness. Other types of limitation can be similarly expressed. For example, a particular reasoning step may be too difficult for novices to carry out:

$$M = \{ \; Difficult(b \rightarrow g) \; \& \; Novice(student) \rightarrow \sim Infer(B_S, L_S, b \rightarrow g) \; \}$$

Both SEDAF and LNT have been implemented in Prolog, in which it is relatively easy to write a meta-interpreter to define meta-level predicates such as `Proof` and `Infer`. Such implementations demonstrate both the methodological advantages of having a formal specification and the practical advantages of concise, rapid prototyping. However, the extra layer of interpretation may lead to inefficiency although techniques are being developed to help overcome this (Donini et al, 1990).

5.2.5 Non-monotonic reasoning

Non-monotonic reasoning and limited reasoning present orthogonal dimensions of complexity of the student modelling problem - they are orthogonal in the sense that it is possible to have either without the other. Usually they are considered together since reasoning that is limited often leads to non-monotonicity. Non-monotonic reasoning refers to reasoning in which a conclusion made at one time may no longer be valid at a later time because of information acquired in the interim.

Non-monotonic reasoning is usually discussed with respect to a single agent, and we need to consider that aspect too. But student modelling itself, involving two agents, is deeply and unavoidably non-monotonic, for three reasons:

1. The system's beliefs about the student's beliefs can never be confirmed as correct and must therefore always be considered subject to revision. Even apparently objective facts such as students' assertions about what they know or descriptions of students' actions need to be regarded as provisional (because they may not fully understand terms they use, or they may make slips in performing tasks, for example).

2. Because of the 'bandwidth problem' systems will rarely have access to sufficient data to permit reliable inferences about students' beliefs, and consequently will have to make default assumptions which may later have to be withdrawn.

3. Students do occasionally learn (and forget) and therefore what the system believes of the student at one time will not necessarily hold at a later time.

Within AI generally, the field of non-monotonic reasoning is vast and active, but there has been little explicit linking to the student modelling problem. In this section, we will just summarise the main approaches and point out the potential relevance to student modelling. Of course, the effect of non-monotonic reasoning may be achieved through computational techniques (such as semantic networks) which are efficient but not completely understood, rather than through the use of formal systems which are generally intractable. However, formal approaches to non-monotonic reasoning may yield benefits in terms of clarity and correctness, and provide useful tools for specifying and describing non-monotonic systems, in particular for that limited class which are actually covered by relatively ad-hoc techniques (Etherington, 1987).

There are two basic approaches to non-monotonic reasoning: model-theoretic and proof-theoretic. Model-theoretic approaches are based on the idea that anything that does not follow is assumed to be false ('model' here is used in the mathematical sense: a model of a theory `T` is any structure `M` such that `T` is true in `M`). Proof-theoretic approaches use non-standard (non-monotonic) logics to derive conclusions through inference rules.

The closed-world assumption (as used in Prolog) is the simplest example of a model-theoretic approach and the various formalisations of *circumscription* the most comprehensive. The basic idea of circumscription is that one considers not all models of `T` but only those which are minimal with respect to a specific property `P`. For example, if we have:

$$B_S = \{\ \texttt{Prime-minister(Thatcher)} \ \&\ \texttt{Prime-minister(Major)} \ \&\ \texttt{Female(Thatcher)} \ \&\ \texttt{Prime-minister(x)) \& \sim Female(x)) -> Public-school(x)}\ \}$$

which says that a student believes the three facts indicated and believes that prime-ministers who are not female went to a public-school, then we cannot logically conclude from B_S that `Public-school(Major)`, since we do not have `~Female(Major)` or, more generally, that the student believes that all prime-ministers except Thatcher are not female. Assuming the equality axioms, applying circumscription to minimize the predicate `Female` we obtain:

$$\texttt{Circum}(B_S,\texttt{Female}) = B_S \ \&\ (\texttt{Female(x) -> (x = Thatcher)})$$

from which the conclusion `Public-school(Major)` now follows.

Circumscription is achieved by means of a second-order axiom schema:

$$\texttt{Circum(T,P)}: T(\Phi)\ \&\ \forall x(\Phi(x)\texttt{->}P(x))\ \texttt{->}\ \forall x(P(x)\ \texttt{->}\ \Phi(x))$$

where $T(\Phi)$ is the result of replacing all occurrences of `P` in `T` by the predicate expression Φ, `P` is an `n`-ary relation and `x` abbreviates $x_1, \ldots x_n$. Informally, this states that if Φ satisfies the conditions satisfied by `P` and every `n`-tuple that satisfies Φ also satisfies `P`, then the only `n`-tuples which satisfy `P` are those which satisfy Φ. For the simple example above, the outcome is the same as for the closed-world assumption. More complex forms of circumscription have been defined, for example, prioritised circumscription allows the predicates which are to be minimised to be placed in an order of relative importance: this, for example, could be used to allow us to conclude `~Public-school(Thatcher)`, which does not follow from $\texttt{Circum}(B_S,\texttt{Female})$ as above. The precise relationships between these forms of circumscription and other forms of the closed-world assumption have been the subject of many studies.

Any student model which purports to represent what a student believes is bound to make implicit use of some circumscription-like scheme since it would be unreasonable to require the student model to represent explicitly all that which the student does not believe. However, formally, the matter is more complex than even circumscription can handle

For, if there is no proposition of the form $B_c B_s p$ then is the missing default assumption $\sim B_c B_s p$ or $B_c \sim B_s p$ or $B_c B_s \sim p$, all of which mean subtly different things? The four possibilities almost correspond to the values 'yes' ($B_c B_s p$), 'no' ($B_c B_s \sim p$), 'unknown' ($B_c \sim B_s p$, although these last two should probably include the $\sim p$ case as well) and 'fail' ($\sim B_c B_s p$) of the four-valued logic for student modelling suggested by Mizoguchi et al (1988).

The use of second-order axioms as in circumscription obviously allows the first-order mechanisms to stay unchanged. Proof-theoretic approaches to non-monotonic reasoning, on the other hand, include within the first-order logic extra rules which allow non-monotonic inferences to be drawn. For such logics, a new semantics must be defined. These non-monotonic logics focus on the notion of normality, i.e. on rules which tend to apply unless there are exceptions.

For example, *default logic* allows a theory to contain ordinary first-order formulas plus defaults, i.e. expressions of the form

```
p : q  -> r
```

which may be read as "if p is derivable and the formula q is consistent (i.e. its negation is not provable) with the theory, then the default rule is applicable and the conclusion r may be drawn." This is non-monotonic because a formula q, previously consistent with a theory, may become inconsistent if new formulas are added to the theory. The two most common forms of default are where q = r or q = r & s, for example:

```
B_c(Physics-graduate(s)) : B_c(Knows-about-momentum(s))
      -> B_c(Knows-about-momentum(s))

Prime-minister(x) : Male(x) & ~Eq(x = Thatcher)  -> Male(x)
```

Such defaults act as rules of 'conjecture', allowing inferences which would not be possible under ordinary first-order logic rules. The conclusion has the status of a belief which may need to be withdrawn (as discussed in section 5.2.1) if the assumption becomes inconsistent. The potential circularity (arising from the fact that what is provable in a default logic both determines and is determined by what is not provable) is avoided by requiring that an *extension* of a particular theory (1) contain all the known facts, (2) be closed under the implication rules, and (3) contain the consequents of all defaults which apply within that extension. In general, there are many possible extensions for a given default theory. Informally, an extension describes an acceptable set of beliefs that an agent may have about an incompletely specified world and thus is similar to the concept of a possible world. Since determining whether a formula is within an extension is undecidable, the implementation of default logic seems problematic, although for most practical cases (e.g. the kinds of default illustrated above) implementations are possible, for example, the LNT system discussed above has an underlying default logic written in Prolog (Poole, 1988).

Default logic involves an agent reflecting upon its own knowledge, in particular, in considering whether a proposition is consistent with what it believes. The notion of consistency is, however, outside the language of default logic. We could instead attempt to capture the notion within the logic, as is done in an *autoepistemic logic*. For example, we could rephrase:

```
p : q  -> r
```

by the sentence:

```
B_a p & ~B_a ~q  -> r
```

i.e. "if p belongs to the agent's belief-set and ~q does not, then r is true." Not surprisingly, given this translation, various formal equivalences between variations of default logic and autoepistemic logic can be established. The main point here, however, is that we have a link through belief logics to other aspects of the student modelling problem, such as limited reasoning, as discussed above, and thus a prospect of being able to tie together all the threads of student modelling, in due course.

Unfortunately, as discussed by Levesque (1990), this link is weakened by the fact that work on autoepistemic logics and default logics has defined different semantics for the notion of belief to that used in belief logics. Levesque attempts to overcome this by defining a second modal operator, in addition to B, namely O, such that $O_a p$ is to be read as "p is *all* that is believed" or perhaps "*only* p is believed", and then by developing a semantics for a language with two such operators. Subsequently, this operator is modified to $O_a[n]p$ for "only p is believed about n", just as circumscription takes place with respect to a predicate and not the whole belief-set.

Clearly, difficult technical issues remain and most work in this area continues to focus on the detailed properties of different formalisations rather than on considering possible applications to areas such as student modelling. However, some general implications for student modelling may be listed:

1. Although some formalisations are theoretically intractable, practical implementations for realistic applications are possible and are beginning to be developed (Donini et al, 1990).

2. Since non-monotonicity often arises through an agent reasoning about its (or other's) beliefs, the meaning of the belief operator depends on the context (i.e. the other beliefs). Thus, B functions as an indexical and expressions in the student model should ideally be indexed. Formally, this is an aspect which has not been considered.

3. Non-monotonicity is intimately related with other aspects of the student modelling problem: for example, determining which belief(s) to withdraw (section 5.2.1) is likely to depend to some extent on which beliefs were derived by ordinary inferences and which by defaults; similarly, limited reasoning may be achieved by not reflecting too deeply about a set of beliefs and hence leading to the derivation of default assumptions.

4. We can use logical notations to describe non-monotonic reasoning without making any psychological claims, or, as Levesque (1990) puts it, we can use logics objectively rather than subjectively.

5. To repeat a general point, formal characterisations of non-monotonic reasoning begin to provide us with a way of precisely describing and analysing aspects of student modelling which at present are proposed, described and implemented in an ad-hoc manner.

5.2.6 Creating beliefs through learning

In so far as they may be distinguished, reasoning leads (through deductive processes) to the creation of relatively temporary beliefs for solving a particular problem, whereas learning leads (through inductive processes) to the creation of relatively permanent beliefs for solving problems in general. If reasoning leads straightforwardly to a problem solution, then learning may well not occur since the agent may re-generate the temporary beliefs by the same deductive processes. If the reasoning process is inadequate in some way then learning processes may be activated to analyse the results of reasoning.

As with reasoning, work on machine learning is relevant to student modelling in two ways: (a) to enable the system to learn about aspects of the student modelling problem, and (b) to model how the student learns. Again as with reasoning, the aim in the latter case is not to seek the unattainable, that is, fully reliable predictions about the student's learning

processes, but to develop a theoretical framework within which such processes may be described and to hope that particular instantiations lead to useful analyses which can form the basis for instructional interactions. In fact, we can repeat with respect to learning many of the points made in the previous sections about reasoning:

: the potential benefits of developing explicit representations of learning processes;

: the recognition that, for both human and system agents, learning processes will be limited;

: the need to distinguish situations were psychological validity is important from those where it is not;

: the fact that the computational intractability of many techniques imposes limitations on what may be possible as far as dynamic student modelling is concerned; and so on.

There is a rich machine learning literature from which to elaborate these points, and there has been considerable work within student modelling which applies machine learning ideas: for example, Langley and Ohlsson (1984) describe a system which aims to induce a student's problem-solving procedure from observations of her solutions by using psychological heuristics to guide hypothesis formation; van Lehn (1987) develops a theory of inductive learning which is intended to model the process whereby students learn from examples, the theory relying heavily on 'felicity conditions', that is, tacit conventions about the teaching-learning process; Costa et al (1988) describe an application of explanation-based learning to the problem of reconstruction, using the technique to disambiguate between possible contexts which a student may have adopted to solve problems.

All these examples, and almost all others from machine learning, are concerned with learning by a single agent (be it system or human student). Rather than expand on these examples, we will discuss one which illustrates the potential of machine learning research to support the view that learning may occur through an interaction between two agents (system and human student, or two human students). Imagine we have two agents, `a` and `b`, who are both trying to learn the same concept but do so by studying different examples and by describing the examples in different ways. How can the two agents make sense of what the other agent learns, in order perhaps to integrate the two different learned concepts (in order to develop a richer joint one) or to be able to discuss the differences between the learned concepts?

Brazdil (1991) gives the following illustration. Each agent has a vocabulary `V` (a list of predicates used to describe the world), a set of observed examples `E`, and a knowledge base `K` describing what the agent knows about the examples in terms of its vocabulary `V`. For example,

Agent `a`	Agent `b`
V_a = { father,mother }	V_b = { parent,male,female }
E_a = { gfather(oscar,steve), gfather(paul,louis), gfather(oscar,andrew) }	E_b = { gfather(william,steve), gfather(oscar,peter) }
K_a = { father(paul,oscar), father(oscar,louis), father(louis,steve), father(louis,andrew) }	K_b = { parent(william,sylvia), parent(sylvia,steve), parent(oscar,helen), parent(helen,peter), male(william), male(oscar), male(steve), female(sylvia), male(peter), female(helen) }

Applying an inductive procedure (e.g. GOLEM (Muggleton and Feng, 1990)), here assumed the same for both agents, to E_i and K_i the agents might induce, in Prolog notation:

```
gfather(X,Y):-                          gfather(X,Y):-
  father(X,Z),father(Z,Y).                parent(X,Z),male(X),
                                          female(Z),parent(Z,Y).
```

Neither agent's rule applies to the other's knowledge base, since the vocabularies are different. Moreover, even if an agent comes to know both rules (say b tells a its rule), then the combined rule

```
gfather(X,Y):-
  father(X,Z),father(Z,Y);
  parent(X,Z),male(X),female(Z),parent(Z,Y).
```

is of no use unless an agent has access to the other agent's description of its world. For example, if a describes using its vocabulary the world from which b derived K_b, we have

```
Ka' = { father(william,sylvia),
        mother(sylvia,steve),
        father(oscar,helen),
        mother(helen,peter) }
```

for which the combined rule fails. In order to fully integrate b's knowledge, a needs to learn b's vocabulary as well, i.e. the concepts parent,male,female. This is possible if b conveys its description to a, so that a can compare the two descriptions K_b and K_a'. Since K_b contains examples of the concepts to be learned, a could apply an inductive procedure to K_b and K_a', to learn:

```
parent(X,Y):- father(X,Y).
male(X):- father(X,_).
female(X):- mother(X,_).
```

Thus, using standard machine learning techniques, we can enable an agent to understand another agent's theory. If we regard one of the agents to be the system, then we could imagine such a procedure being applied to handle situations where the student has a different viewpoint about the domain to that of the system. If we imagine both agents to be human students, then the above kind of analysis might be adapted to enable a system to help the students work collaboratively to come up with an agreed integrated theory of the domain.

5.3 Beyond belief

So far we have adhered to the original definition of the student model in terms of a belief-set, i.e. an unstructured set of beliefs held by the system about the student (including her beliefs), where the object of a belief was taken to be a simple proposition. However, this adherence has been strained in several ways, for example, in the need to associate links between beliefs to facilitate belief revision, and in the need to consider a production rule as a kind of proposition. In this section, we will review several extensions to the simple belief-set. Again, this is related to the broad area of knowledge representation research in AI and we will only discuss aspects particularly relevant to student modelling.

5.3.1 Belief structures

A belief-set may become structured, and hence a 'belief-structure' rather than a belief-set, in basically two ways: by specifying relationships between pairs of elements and by

partitioning the elements into subsets. The former is needed, for example, to indicate that one belief is derived from or is a generalisation of another one; the latter, for example, to deal with local reasoning (section 5.2.3).

Two of the most useful relationships to specify are those of abstraction (`isa`) and aggregation, i.e. part-whole relationships (`partof`). Greer and McCalla (1989) structure their belief space as a lattice based on these two relationships. One dimension specifies abstractions - for example, in their domain of Lisp programming strategies:

```
{ function-definition isa lisp-program,
  recursion isa function-definition,
  cdr-recursion isa recursion,
  car-recursion isa recursion, ... }
```

The other dimension specifies part-whole relationships, for example:

```
{ null-base-case partof cdr-recursion,
  recursive-cdr-reduction-case partof cdr-recursion,
  some-test-default partof recursive-cdr-reduction-case, ... }
```

Those elements which are not `partof` any other element may be shown in a 'principal abstraction hierarchy'. If there are no abstraction relationships between elements not in the principal abstraction hierarchy then one can picture 'slicing' the space into objects at different 'grain sizes'.

Such a structuring is useful for student modelling because it enables diagnosis to be carried out at an appropriate level of detail. For example, given a particular student solution, such as

```
(defun findb (lst)
   (cond ((null lst) nil)
         ((atom 'b) t)
         (t (cons nil (findb (cdr lst))))))
```

then it may be difficult to identify specific mal-rules, if any, but we may be able to at least recognise the solution as an example of `cdr-recursion`. We can imagine searching the abstraction hierarchy from the root (`lisp-program`) seeking the lowest node (`cdr-recursion`) for which the specified parts are present (here we have a `null-base-case` and a `recursive-cdr-reduction-case`). Pedagogically useful information may be determined by analysing why the solution is not recognised as an instance of lower nodes (e.g. `cdr-tail-end-recursion`) in the abstraction hierarchy on the path from the recognised node to the 'preferred solution' node. Thus, it may be possible to limit some of the problems discussed in section 5.2 (such as belief revision) which may arise because of a premature commitment to a default assumption, for example. Instead, the system may maintain an appropriately vague student model, which is refined only when it is possible to do so. In this sense, Greer and McCalla's granularity scheme is similar to the bounded model approach of Elsom-Cook (1988) where a version space maintains upper and lower bounds on the possible states of the student.

A student-model oriented structuring of the domain concepts is not necessarily the same as a curriculum-oriented one. For example, Greer and McCalla's abstraction hierarchy may not carry pedagogical implications, such as that the general concept of `recursion` should be taught before the more specific concept of `cdr-recursion`, or vice versa. While the actual structures may turn out to be quite similar, their purposes are very different. Here we are concerned only with the aim of enabling 'imprecise' student modelling. The progression of causal models from qualitative to quantitative developed by White and Frederiksen (1990) appears similar to Greer and McCalla's abstraction hierarchy, but the former's aim is to eliminate most student modelling problems by building systems which enable students to build their own models. Student modelling then becomes a matter of the

system identifying which of the pre-specified sequence of causal models the student has acquired, and thus is a version of overlay modelling - "the students are assumed to have the current model when they can correctly solve problems that the current model can solve but the previous model could not" (White and Frederiksen, 1990, p150).

5.3.2 Viewpoints

Specifying relationships between elements of a belief-set is useful when the student modelling component needs to adjust its focus on the student: partitioning the elements into subsets is useful when a different pair of spectacles is needed altogether. For example, imagine (from Costa, Duchènoy and Kodratoff (1988)) that:

```
Bcs =   {   Man(x) & Noble(x) & Live(x,17th-century) -> Wig(x),
            Man(Louis-XIV) ,
            Lived(Louis-XIV,17th-century),
            Wig(Louis-XIV), ... }
```

and the system asks the student why Louis-XIV wore a wig, expecting the answer "because he was a noble" since this may be derived from the student model. If instead the student answers "because he liked having fun" then the system may seek a context different to the 'Louis-XIV as a 17th-century noble' context to make sense of this answer. It may, for example, replace the first proposition above by one or more axioms in the system's belief-set defining alternative contexts:

```
Bc = {  {   Man(x) & Criminal(x) -> Disguise(x),
            Disguise(x) -> Wig(x) }, ...
        {   Man(x) & Bald(x) -> Wig(x) }, ...
        {   Man(x) & Farceur(x) -> Have-fun(x),
            Have-fun(x) -> Wig(x) }, ... }
```

Finding a context, e.g. 'Louis-XIV as a farceur', from which `Wig(Louis-XIV)` may be derived enables the student model to be revised and an appropriate response to be generated.

We provisionally define a *viewpoint* V_a to be a triple $<B_a,L_a,M_a>$, where each element is a subset of the agent's complete belief, logic and meta-logic space, respectively. The above example is just concerned with the belief-set but in general we might imagine different viewpoints to involve different reasoning processes. If the agent is the system then we are emphasising the fact that there may be many different ways of looking at one domain, rather than just one definitively correct one: if the agent is the student then we are recognising that students may well have different views about the domain to the system. Both aspects have been emphasised recently in attempts to move intelligent tutoring systems beyond straightforward knowledge commmunication systems. Wenger (1987) identifies 'viewpoints' as a topic ripe for more research and as a means of shifting from "what is wrong to what is right".

However, the notion of a viewpoint is rather diffuse despite, or perhaps because of, being studied under various guises within many branches of AI and computer science (Self, 1991a). In distributed AI, for example, there are discussions of different agents (or nodes in a network) holding different views about some problem and the need to divide a problem between agents and to coordinate the behaviour of them. They emphasise the role of *negotiation* to reconcile potentially conflicting views (section 6.3). Similarly, in ITS research, we find an increasing emphasis on the student's ability to negotiate both about the concepts to be discussed and about the meaning of the concepts themselves (Moyse and Elsom-Cook, 1991).

To illustrate the potential relevance of viewpoints to student modelling we may adapt an example from Wilks and Ballim (1987), who consider a 'viewpoint' to be an agent's set of

beliefs about some topic. The example emphasises that such viewpoints need to be nested. Imagine the system is mediating an interaction between two medical students a and b. We might have:

```
B_c = {  Type(thalassemia,genetic-disorder),
         Medically-informed(x)
             -> B_x(Type(thalassemia,genetic-disorder)),
         Average-person(x)
             -> B_x(Type(thalassemia,disease)),
         Type(x,genetic-disorder) & Suffers(a1,x)
             & Suffers(a2,x) & Child(a1,a2,a3)
             -> Suffers(a3,x), ... }
```

The system's might have the following two student models:

```
LM_a =   {   B_a(Suffers(fred,thalassemia)),
             B_a(Suffers(mary,thalassemia)),
             Medically-informed(a), ... }

LM_b =   {   B_b(Suffers(fred,thalassemia)),
             B_b(Suffers(mary,thalassemia)), ... }
```

i.e. the system believes a to be medically-informed but not b. From such student models, the system might reason that

```
LM_a =   {   ...,
             B_a(Type(thalassemia,genetic-disorder)),
             B_a(Child(fred,mary,a3) -> Suffers(a3,thalassemia)),
             ... }

LM_b =   {   ...,
             B_b(Type(thalassemia,disease)), ... }
```

i.e. that a will reason that a child of Fred and Mary will suffer from thalassemia but that b will not (on the default assumption that b is an average person). The system could then carry out independent dialogues with the two students but neither such dialogue would be of much interest to the other student. Instead, the system could take account of what one student believes the other student believes. For example, the system might consider that a believes b is also medically-informed (making the default assumption that b is the same as a unless a has evidence otherwise) and thus that

```
B_cab =  {   ...,
             Type(thalassemia,genetic-disorder),
             Child(fred,mary,a3) -> Suffers(a3,thalassemia), .. }
```

Then, for example, the system might engage a in discussing with b why b's conclusions differ from those expected by a of b. In general, the point is that in any interaction between two or more agents it may help for an agent to hold beliefs about what may be believed by the other agent(s).

There are many theoretical and practical difficulties to be overcome before the idea of viewpoints can be fully used in student modelling. For example, it is clear that viewpoints are not entirely independent but that some beliefs may be shared between even radically different viewpoints, although perhaps some core set of beliefs may be unique to one viewpoint. Practically feasible ways of handling multiply-overlapping belief sets need to be developed. Also, we need efficient ways of identifying the student's working viewpoint - will it suffice to work systematically through potential viewpoints (as implied in the 'wig example') since there may not be many of them, or will we need to reason about

mismatches with the previously assumed viewpoint? And, do all the potential viewpoints need to be anticipated or may they be generated, as needed, by the system?

5.3.3 Plans

In section 5.1.1 we defined reconstruction as the interpretation of a set of student inputs in terms of the propositions which the student may be held to believe. Often, however, the system is concerned to interpret the inputs in terms of what the student is doing rather than what she believes or knows, because the system may intend to discuss plans and goals directly or because by re-directing the student's plans the system may lead her more effectively to the desired beliefs. The problem of identifying the student's plans is unfortunately complex for the following reasons:

1. Unlike planning itself, plan recognition is inherently a multi-agent process since it involves one agent (the system) reasoning about the plans of another (the student) - except in those situations where an agent is trying to reconstruct its own planning.

2. It invariably involves uncertain reasoning since a set of observed actions rarely uniquely identifies a plan (yet definite conclusions may still be drawn even after uncertain reasoning).

3. Students are particularly prone to leave out actions, insert faulty actions, interleave actions from some other plan, and often to have no plan anyway!

Many approaches to plan recognition transform it into a parsing problem. A grammar specifies how plans are decomposed into actions and sub-actions, and a particular sequence of observations is regarded as a sentence to be parsed with respect to this grammar. Formally, this is no doubt a sufficient characterisation of the problem, but we will instead describe a method developed by Kautz and Allen (1986) which is closer to our view of student modelling.

The method requires the specification of three kinds of information:

1. The observations, e.g.

```
Occurs(e9,make-pasta)
∃e Occurs(e,make-noodles) & T(e)=17
```

i.e. event `e9` is an instance of type `make-pasta`, and an event of type `make-noodles` occurred at time 17. Such a description is based on a general theory of action and time (Allen, 1984) and inherits from it axioms such as

```
∀e,i During(T(S(i,e)),T(e))
```

i.e. the time of the `i`-th subaction of event `e` occurs during the time of event `e`. (We might imagine a student using a simulation to learn how to cook or to perform some similar activity.)

2. An action hierarchy, i.e. an exhaustive description of the ways in which an action can be performed and of the ways in which an action can be used as a step of a more complex action. These are specified as axioms of specialisation and decomposition (which are just the inverses of Greer and McCalla's abstraction and aggregation):

```
∀e Occurs(e,make-pasta) -> Occurs(e,prepare-meal)
∀e Occurs(e,make-fettucini) -> Occurs(e,make-noodles)
∀e Occurs(e,make-spaghetti) -> Occurs(e,make-noodles)
...
∀e Occurs(e,make-pasta) ->
```

```
        ∃t  Occurs(S(1,e),make-noodles) &
            Occurs(S(2,e),boil) & Occurs(S(3,e),make-sauce) &
            Object(S(2,e))=Result(S(1,e)) &
            Hold(Noodle(Result(S(1,e)),t) &
            Overlap(T(S(1,e)),t) & During(T(S(2,e)),t)
...
```

The decomposition axioms specify the subactions, their preconditions and effects, and constraints on temporal relationships. Of course, subactions may also be decomposed. In addition, the system needs a set of disjointedness axioms, e.g.

```
∀e Occurs(e,make-fettucini-alfredo) not-and
          Occurs(e,make-fettucini-marinara)
```

3. A set of 'simplicity constraints' to choose between interpretations, e.g. "minimise the number of top-level actions". These are represented as second-order logical formulae, which are instantiated to first-order formulae for any particular case.

Before recognising a plan, the action hierarchy is supplemented by axioms derived by applying circumscription to make the assumptions that (a) the known ways of performing an action are the only ways and that (b) all the possible reasons for performing an action are known:

```
∀e Occurs(e,prepare-meal) ->
     Occurs(e,make-pasta) exc-or Occurs(e,make-meat)
...
∀e Occurs(e,make-noodles) ->
     ∃a  Occurs(a,make-pasta) & e=S(1,a)
∀e Occurs(e,make-marinara) ->
     ∃a  (Occurs(a,make-fettucini-marinara) & e=S(3,a)) v
         (Occurs(a,make-chicken-marinara) & e=S(3,a))
```

Although there is no general method for carrying out circumscription, these axioms are easily derivable by special-purpose algorithms which retain the benefits of having a formal semantics for the process.

Now, given an observation, e.g.

```
Occurs(e1,make-fettucini) v Occurs(e1,make-spaghetti)
```

i.e. that the student is making fettucini or spaghetti (but we're not such which), we may infer

```
∃e Occurs(e,make-pasta)                                    ..(1)
```

and hence that, for example,

```
∃e Occurs(S(2,e),boil)
```

So, even though particular actions and plans may not be identified, specific predictions may be made. If we now observe:

```
Occurs(e2,make-marinara)
```

then the system can infer, from the specialisation axioms, that

```
∃e Occurs(e,make-pasta) v Occurs(e,make-meat)
```

Given the previous inference (1), the simplicity constraint mentioned above would eliminate the second disjunct of this inference.

Thus, the system may monitor the student's actions and attempt to derive the student's plans. The virtues of this approach are that it provides a formal theory with a precise semantics for the plan recognition process by specifying axioms (supplemented by circumscription) from which conclusions are derived deductively and it thus integrates plan recognition with other aspects of student modelling discussed previously (instead of regarding plan recognition as a rather specialist sub-problem for which different techniques are needed).

5.3.4 Meta-beliefs

Planning is but one of a set of "mysterious mechanisms" (Brown, 1987) denoted by the terms *metacognition* and *metaknowledge*. We might distinguish, for example, between problem-solving itself and reasoning before, during and after problem-solving (planning, monitoring and reflecting, perhaps). The issues are complex and no attempt will be made here to solve any mysteries - simply, some links to student modelling will be discussed.

Whatever they are precisely, metacognitive abilities are considered important in both education and AI. Dewey, Vygotsky and Piaget all emphasised various aspects of metacognition, and more recently Schoenfeld (1987) has stressed their role in mathematics education. In AI, metaknowledge, metareasoning and meta-level architectures have been extensively discussed (e.g. Maes and Nardi (1988), Genesereth and Nilsson (1987)), and, although there has been no explicit link to such AI research, ILE design has increasingly emphasised metacognitive aspects (e.g. Collins and Brown (1988), Shute and Glaser (1990)).

However, metacognition is not an unqualified benefit. Its alleged importance derives from a view that it is necessary for an agent not only to know more than a set of facts (namely, how to apply them to solve problems) but also to be able to reason rationally about the problem-solving process itself. This, it is assumed, will enable the agent to improve problem-solving performance (i.e. to learn), to develop transferrable metacognitive skills, and to engage in discussions (e.g. tutorial interactions) about such processes. These may sound like platitudes but they are questioned by those who doubt that activity derives, or should derive, from a rational reasoning process: instead, it might follow in response to the situation in which the agent finds itself. Others may even question the implicit educational aim of fostering rationality. We cannot resolve such issues, but we can concede that metacognitive mechanisms must be applied with caution: an agent that spent too much time at the meta-level might accomplish less at the object-level.

Some computational mechanisms for addressing metacognitive issues have already been introduced, for example, axioms of introspection in modal logics (section 5.2.3) meta-level reasoning (section 5.2.4), and autoepistemic logics (section 5.2.5). As usual, our aim is not to develop such mechanisms per se but to apply them to student modelling. For example, the axioms of positive and negative introspection and their inverses (and corresponding axioms for knowledge):

$$B_a p \rightarrow B_a B_a p$$
$$\sim B_a p \rightarrow B_a(\sim B_a p)$$
$$B_a B_a p \rightarrow B_a p \text{ , etc.}$$

are clearly inadequate as a basis for deriving reliable conclusions concerning a student's beliefs about her own beliefs but rather than embark on a probably futile attempt to 'correct' them we may seek to indicate how they may be used to build student models adequate to support instructional interactions. For example, imagine a student attempting to solve fraction problems ($f_1 - f_2 = f_3$) who asserts that she believes that it is always the case that $f_1 > f_3$. From an axiom such as

$$Asserts_S(p) \rightarrow B_S B_S p$$

and the third rule above, the system might infer $B_S(f_1>f_3)$. If however her problem-solving performance leads to $B_C B_S q$, where $q \rightarrow (f_1<f_3)$ - for example, the student may appear to believe the mal-rule that one subtracts both numerators and denominators and thus obtains 7/8 - 3/4 = 4/4 - then the system might initiate a dialogue about the apparent contradiction, but in terms of general beliefs about the problem domain not the specific problem.

In general terms, an intelligent agent should be able to reason about its own problem-solving performance, for example, to consider the merits of alternatives, and should be able to use the results of such deliberations in subsequent problem-solving. This requires a metalanguage in which to formalise the process of problem-solving. Genesereth and Nilsson (1987) show how predicate logic, extended with a quoting mechanism to overcome quantification problems, can be used to formalise the process of resolution theorem-proving. The same method can be used (as we assumed in section 5.2.4) to describe other inference procedures, such as limited or even unsound ones. We might in fact define what it means for an agent to believe a proposition in terms of a meta-level predicate `Provable`, defined as appropriate:

$$\forall a \forall p \; B_a p \Leftrightarrow \texttt{Provable}(B_a, p)$$

In AI there are (at least) two kinds of metaknowledge - knowledge about how to use knowledge, and knowledge about the contents of one's own knowledge. The former has been thoroughly studied in the form of expert system meta-rules, which generally help determine rule selection, but we will look instead at the more subtle issue of *reflection*. The latter kind of metaknowledge, which has been less studied, is concerned with issues of *introspection*, which is clearly of relevance to student modelling and will be considered below.

Dewey (1938) defined reflection as the "active, persistent and careful consideration of any belief or supposed form of knowledge in the light of the grounds that support it". Genesereth and Nilsson (1987) are more specific - reflection is "the process of suspending the process of reasoning, reasoning about that process, and using the results to control subsequent reasoning" - and they also provide a computational realisation of the definition. The basic idea is to include within the meta-level a specification of the conditions under which a 'reflection phase' is entered. For example, we might have

$$\begin{aligned} M = \{ \; & \texttt{Infer}(B, \{p_1, p_2, \ldots, p_n\}) \; \& \; n < d \rightarrow \{p_1, p_2, \ldots, p_n\} \; , \\ & \texttt{Infer}(B, \{p_1, p_2, \ldots, p_n\}) \; \& \; n \geq d \rightarrow \texttt{Reflect}(B) \; \} \end{aligned}$$

to indicate that if the number of inferences which could be made from a belief-set is less than some threshold `d` then they are made, otherwise some reflection process is begun. Obviously, `Reflect(B)` is intended to lead to some change in `B` and hence to changes in subsequent processing. Various other conditions which promote reflection could also be defined, for example, a 'compulsively reflective' agent might be one who reflects during every inference step. (We are glossing over many subtle technical and philosophical issues. For example, on what basis are the steps of `Reflect` separated from those of `Infer` and considered to be at a different level? - exactly the same result could be achieved by redefining `Infer` if we wished.)

Such a mechanism could be used to lend precision to attempts at the cognitive modelling of reflective processes. For example, Foss (1987) tried to specify conditions under which students abandoned a solution path when using the AlgebraLand system - conditions such as 'the result of an operation is longer than the previous expression' or 'an unreasonably complicated piece of arithmetic is required'. At the moment, however, as Genesereth and Nilsson (1987) admit "little is known about when a procedure should reflect", nor indeed about what the results of reflection should be. Nonetheless, we may anticipate that research on computational reflection will lead to some much-needed precision in discussions of such metacognitive aspects and eventually enable ILEs to make some

predictions about which kinds of instructional event may promote reflection and about when to intervene to suggest that the student might reflect (i.e. pause from problem-solving and reason about her progress).

It is sometimes argued that what a student knows is less important than what she knows she knows (or does not know) - which is what the work on introspection is attempting to formalise. Knowledge of one's own limitations can be a reason for acting, to acquire knowledge, and for not acting, to avoid contemplated actions outside one's competence. We would like to be able to handle common situations such as a student asserting that she believes she knows nothing about art:

$$\forall p \ \text{About}(p,\text{art}) \rightarrow B_S \sim K_S p$$

and to make reasonable inferences from such expressions, e.g.

$$\forall p \ B_S \sim K_S p \rightarrow B_S(\text{Not-worth-knowing-about}(p))$$

and to use more general axioms, such as those of 'arrogance' and 'coherence' (Davis, 1990):

$$B_a(B_a p \rightarrow p)$$
$$B_a \sim B_a p \rightarrow \sim B_a p$$

The last of these is logically equivalent to

$$\sim B_a(p \ \& \ \sim B_a p)$$

which holds, according to Davis, for any agent who is not "seriously confused" - unfortunately, our students often are. These kinds of axiom, over-simple though they are, are closely related to methods developed for non-monotonic reasoning, as discussed by Konolige (1988). For example, the closed-world assumption may be re-expressed as

$$B_a \sim K_a p \rightarrow B_a \sim p$$

(anything that is not known is believed to be false), and default logic can be represented in autoepistemic terms, as we saw in section 5.2.5.

But we must pause to reflect on what this research may contribute to student modelling. If reflection "is the transferral of argumentation to an internal level" (Vygotsky, 1978) then there is little chance that an ILE would be able to monitor or reconstruct a student's reflective processes in the way that we have imagined for problem-solving processes (which is difficult enough). 'Internal reflection' is not an activity for which moment-to-moment student modelling is possible or appropriate - rather like reading from a hypertext system, where we found that minutes of apparent inactivity separated flurries of activity (Taylor and Self, 1990). ILE interruptions to "tell me what you are thinking" may well be counter-productive, since they will interfere with on-going cognitive activity.

However, in some situations, it may be beneficial to make the reflection 'external'. The ILE's role might then be to determine when it is appropriate to externalise reflection and other meta-level processes and to share in its execution. For example, if the student model indicates that $B_c B_S p$ and $B_c B_S \sim p$, then it may be more rational for the ILE to conclude nothing (let alone embark on a risky reason maintenance exercise) except perhaps that one or both is wrong and to discuss the issue with the student. Similarly, if the student model indicates that in the current problem-solving situation one of a number of rules could have been applied, rather than attempting to second-guess which (if any) has in fact been applied, an ILE might do better to enter a meta-level where it is discussed explicitly. Often, such a discussion may bring into the open issues of which a student is only implicitly aware. For example, in algebra problem-solving, performance might lead to $B_c B_S f$, i.e. to the system believing that the student 'believes' a particular fault. Whether or not the student 'really believes' such a fault is debatable; perhaps it should be debated with the

student. Payne and Squibb (1990) show that students do have sufficient metacognitive awareness that they are able reliably to assign levels of confidence to their answers: in some cases, a wrong answer (which is actually believed to be wrong by the student) is evidence that the student believes that she does not know something ($B_s(\exists p \sim B_s p)$), not that she genuinely believes something which is in fact incorrect. Different pedagogic interactions are surely needed for such different situations.

5.3.5 Attributes

The idea of using student attributes, i.e. properties or qualities which can be predicated of the student, in student modelling is intuitively appealing but has proved to be of limited practicality, possibly because the exorbitant effort required to build an ILE capable of dealing with a single content in a single way has precluded attempts to build systems capable of dynamically varying content and teaching strategy, for which student attributes may be more relevant. Our aim in this section is not to argue the case for or against the use of student attributes but to consider how they might be encompassed within our theoretical framework.

Our general approach is to associate an 'attribute' with a meta-level description of a component of a student model, although our comments are necessarily speculative. As we have seen, some components of a student model are domain-dependent and some are not (and some are intermediate): attributes tend to be associated with domain-independent components.

We are naturally only concerned with attributes (sometimes referred to as *aptitudes*) which have some bearing on learning. Corno and Snow (1986) identify three kinds of aptitude:

1. Cognitive, e.g. (prior) knowledge and intellectual abilities. ILE design has tended to emphasise the role of knowledge, implicitly agreeing with Chi, Glaser and Rees (1982) that students' difficulties "can be attributed mainly to inadequacies of their knowledge base and not to limitations in either the architecture of their cognitive systems or processing capabilities". This kind of attribute is similar to that of a stereotype (section 4.2): to say that a student is a 'Unix-expert', for example, is to say that she knows a certain set of propositions. Attributes referring to intellectual abilities describe the reasoning and learning components of the student model. For example, a student with good 'visual analogic intelligence' would be modelled by including a component good at visual analogy.

2. Conative, i.e. concerned with wants, intentions, etc. and, in the educational context, cognitive and learning styles. Many such styles have been studied, usually in terms of contrasts: holist v. serialist, reflective v. impulsive, convergent v. divergent, etc. These attributes seem to refer to global properties (rather than the performance properties, as with cognitive attributes) of the meta-level of a student model. Thus a 'holist' style refers to the general strategy of the learning component. Similarly, 'reflective' refers to the number and type of meta-level interruptions on the base-level problem-solving.

3. Affective, i.e. concerned with values. This includes the difficult issue of 'motivation', for which many attributes such as anxiety, autonomy, self-concept, etc. remain to be disentangled. It is hard to imagine how such attributes may be modelled other than by simple labels but 'self-concept', for example, is presumably concerned with what a student believes about the model she has of herself.

The permanence or otherwise of such attributes is also a concern. Some attributes (e.g. knowledge of a particular law of physics) are transient and it may be the ILE's aim to change them, some (e.g. blindness) may be permanent but nonetheless of pedagogic concern if not focus, but many (e.g. anxiety) may be situation-dependent. The value of relatively permanent attributes may be determined off-line, through psychological tests which are outside the scope of this review. On-line interrogation concerning attributes is of

little use since the technical terms used in educational research are rarely used by students in the same sense. The on-line assignment of a particular event (or series of events) to student attributes can be problematic: for example, a student may neglect to try to disconfirm her hypotheses - is this evidence for certain cognitive, conative or affective attributes?

The use of a simple label for attributes ("student is a Unix-expert", "student is reflective") may not be theoretically insightful but may be practically adequate, if that is all the instructional component needs to know. Ideally, however, such attributes should be defined in terms of other contents of the student model:

$$\text{"student is a Unix-expert"} \leftrightarrow K_S\{p_1, p_2, \ldots, p_n\}$$

Perhaps, in general, we could say that the system believes the student possesses an attribute A_S iff the student model possesses certain properties $\{p_1, p_2, \ldots, p_n\}$, to be defined (which is difficult, of course):

$$B_C A_S \leftrightarrow \{p_1, p_2, \ldots, p_n\}(LM)$$

For the attribute to be more than a shorthand summary, however, the set $\{p_1, p_2, \ldots, p_n\}$ needs to be partitionable into two subsets: those properties which enable the attribute to be recognised and those which follow from its recognition. If either subset is empty then the attribute serves no student modelling purpose.

Maybe, as has happened with learning processes, the necessary formal precision will evolve from computational descriptions rather than through attempted analyses of informal educational and psychological literature. Previously, we have seen formal definitions of terms such as 'narrow-minded', 'compulsively reflective', and 'arrogant', and other definitions, for example, of 'persistent' and 'cooperative', exist in the AI literature. Such terms are used without necessarily any psychological claims (and perhaps even semi-seriously), but at least they provide a benchmark against which educational psychologists can try to define the terms when they use them.

6. Using the student model

Describing student modelling, as we have, virtually independently of other ILE activities creates the impression that that is how student modelling should actually be carried out. In fact, only if student modelling is tightly coupled with, in particular, the instructional component of the ILE is the task likely to prove tractable. Analysing the current event becomes much more feasible if it is not done in ignorance of the instructional context and of previous analyses of the agents' plans and goals. Unfortunately, this coupling has not been addressed in any formal way. In this section, we will not attempt a thorough review of instructional activities but just describe some that impinge upon student modelling.

6.1 Prediction and planning

An ILE needs to be capable of *dynamic planning*, that is, the on-line creation and revision of instructional plans, since for any significant learning context the pre-specification of a plan to be strictly followed is not possible. In order to plan, an agent needs to be able to evaluate the states which it predicts that it could reach. In our case, since the ILE's goals concern what the student learns, the evaluation is determined by a function of LM. This function could be defined in many ways, with respect to pre-specified objectives or intrinsic properties of the student model.

This evaluation is not of the current student model but of ones which might exist after a sequence of instructional events. These hypothesised student models need to be predicted by the reasoning and learning components of the student model (section 5). Thus

components of the current student model map the student model and the sequence of events onto a new student model:

$$LM \times \{o_1, i_1, o_2, i_2, \ldots, o_n, i_n\} \rightarrow LM'$$

where $\{o_1, i_1, o_2, i_2, \ldots, o_n, i_n\}$ is a sequence of system outputs and student inputs. Because of the indeterminacy and cost of the mapping and the large number of potential events, n is usually kept small (giving the opportunism often considered characteristic of instructional planning). Of course, the actual student inputs are not known at the time when the plan needs to be created: but they are to some extent predictable by the student model, otherwise the mapping would need to take account of all possible student inputs. In general, for any contemplated system output o_j one of a small number of student inputs may be anticipated. This small set of anticipated inputs greatly eases the reconstruction problem (section 5.1.1), since often the desired analysis has been 'preconstructed'.

The plan created depends on more than the student model. For example, it may depend on any curricular organisation of the subject matter and on any resource constraints, which are outside our scope. But the plan itself is to be regarded not as a recipe for system action but as a context to support interpretation by the student modelling process. Clearly, an ILE that merely reacts to (as opposed to interacts with) a student does not need to predict or plan, but any ILE that takes any kind of instructional initiative must base its decision concerning the initiative to take on some kind of instructional plan, possibly implicit.

Since no plan can be created in a vacuum, plans are 'dynamic' only to a degree. Most current ILEs plan only to the extent of dynamically choosing between pre-specified skeletal plans (e.g. Meno-Tutor (Woolf, 1987)) or between pre-specified problem sets (e.g. GREATERP (Reiser, Anderson and Farrell, 1985)). ILEs which come closer to planning as the term is understood in AI - that is, constructing an explicit plan representation, subject to constraints, which is interpreted, monitored and revised - include those of Murray (1989) and Peachey and McCalla (1986), the latter explicitly representing expected changes to the student model.

6.2 Diagnosis and remediation

The phases of planning and diagnosis need to be interleaved in order to minimise the problems of both, but how precisely this may be done is not known. The results of diagnosis are represented by the contents of the student model and thus are in general terms the system's understanding of the student. More narrowly, diagnosis is often construed as the process of finding faults (as in section 5.1.2), which may then be subject to remediation.

Remediation may take many forms (it is for educational psychology to determine which form is appropriate when - our task here is to relate the forms to student modelling):

1. Reteaching. If we have that $B_c \sim B_s p$ then the system may reteach the proposition p (in the same or a different fashion to previously). More generally, if

$$B_c(\exists p \; p \in \{p_1, p_2, \ldots, p_n\} \; \& \sim B_s p)$$

i.e. the system believes the student does not know one of a set of propositions but cannot determine which, the system may reteach the whole set. A priori, we might expect reteaching to be more effective if the student model enables it to focus on specific gaps in knowledge. But, in general, reteaching is an option whenever the student model indicates that the student has some difficulty but the system cannot identify it. For example, Ikeda, Mizoguchi and Kakusho (1988), rather despairingly, consider that when a student has a "nonlogical belief structure" then "the task of constructing a model is not only difficult but futile ... in such cases ... the only possible instruction ... is to give elementary explanation of the teaching material".

2. Emendation. If the system is able to carry out a fault-diagnosis (section 5.1.2) and has identified one or more paired terms <p,f>, where f is a faulty version of a proposition p, then the system may seek to emend the fault. For example, McCoy (1989) suggests that misconceptions (of expert system users rather than students) may be emended by a three-part system output: (i) a denial of f, (ii) an assertion of p, and (iii) a 'justification', often based on a refutation of the user's support for f. She also emphasises the role of the user's viewpoint or domain perspective in determining the user's support for a misconception. Various frames are specified for addressing certain kinds of misconception - for example, for a 'misattribution':

$B_C B_S$(f: x has attribute y with value v) &
B_C(p: x has attribute y with value w) &
$\exists z\ B_C$(z has attribute y with value v) & B_C(Similar(x,z))
-> Deny(f) & Assert(p) &
Comment("have you confused x and z, etc")

It should also be pointed out that emendation may also address difficulties not at the object level. If the student model has a sufficiently explicit representation of the student's reasoning, meta-reasoning and learning abilities then shortcomings here may also be focussed upon. For example, if the student is developing mal-rules through some impasse-repair mechanism then it may well be more productive to address this mechanism rather than some specific mal-rule that results, that is, to point out that 'syntactic patches' to overcome local difficulties is not always a productive strategy..

3. Counter-examples. A counter-example is a problem p such that

$$D_C(p) \rightarrow A_C \ \&\ D_S(p) \rightarrow A_S \ \&\ \text{not}(A_C = A_S),$$

where the domain knowledge is represented procedurally, for example, as a production system. Evertsz (1989) describes techniques for generating counter-examples, given two production systems (representing the system's and the student's knowledge). Of the set of potential counter-examples, the system might prefer one that generates an A_S which violates any beliefs the student may have about answers in general (for example, a fraction problem for which A_S=0). Different instructional interactions may then ensue depending on whether or not the student realises that the example is in fact a counter-example.

4. Garden-path problems. Among many more subtle remediations, we briefly mention just one, the use of 'garden-path problems', that is, problems which (according to the student model) the student can solve but only in such a tortuous fashion that she may realise that her current knowledge, while not actually incorrect, is inadequate. Methods for the automatic generation of such problems have yet to be developed. Formally, it would appear to be related to the conditions which promote reflection and to the results of any reflective process.

When discussing diagnosis, we should distinguish carefully between situations in which the system attempts to lead the student to diagnose her own (mis)understanding - such as the above - and those in which the system attempts to diagnose its own (mis)understanding about the student. The former case may be characterised as the student resolving $B_C B_S(p \vee q)$; the latter as the system resolving $B_C(B_S p \vee B_S q)$. Similar techniques (e.g. the generation of counter-examples) may be used in both cases, although the aims and instructional interactions are different. Work on formal theories of diagnosis (e.g. Reiter (1987)) has considered the general problem of automatically determining which 'measurements' to take when attempting to differentiate between multiple potential diagnoses of a faulty system.

ILE diagnosis is a rather richer concept than the diagnosis of formal AI theories. The former is more concerned with understanding the student than with identifying her difficulties. Moreover, the student is considered to be an active participant in the diagnostic process (and not just a system to be observed), with the consequence that the diagnostic

process itself may change the student being diagnosed. Such aspects have not been incorporated in formal theories of diagnosis.

6.3 Negotiation and collaboration

The role of a student model in a dogmatic intelligent tutoring system seems clearcut: it is mainly to identify misunderstandings which the system may remediate. In other styles of ILE, however, where the student may have greater scope for following her own goals and developing her own understanding, the role of the student model is more subtle (but not non-existent). No longer is the emphasis on developing detailed object-level models, defined with respect to specified domain knowledge, but on representing the student's beliefs and goals on their intrinsic merits, in order that the system may offer (fallible) comment and advice in the role of a cooperative partner rather than a knowledgeable tutor.

Such a role may be achieved by a disingenuous concealment of its domain knowledge by the system - but the role is likely to be more appropriate in situations where computational representations of domain knowledge are unattainable or controversial. If the system's domain knowledge is not complete or necessarily correct, then the system may need reasoning and learning capabilities commensurate with those of the student. With such capabilities, the system may maintain a student model, which together with the system's model, represents some joint understanding of the domain. Naturally, in such a context, the student model is less an internal component of the ILE but becomes an 'external' focus of discussion, as indeed does the system's model of the domain. Thus the student model may be built by a more direct interaction with the student rather than through some internal analysis by the system.

The view that a system-student interaction is just one instance of the class of multi-agent interactions leads us to consider distributed AI, where concepts such as *negotiation* and *cooperation* have been much studied but have yet to take clear formal shape. Durfee and Lesser (1989) consider that there is "confusion and misunderstanding among researchers who are studying different aspects of the same phenomenon". They urge that we distinguish carefully between negotiations which are about the shared construction of meaning and those which are about task-sharing or planning. Both are of course central to the philosophy of ILEs, the former being concerned with the nature of knowledge and the latter with the issue of student control, and both being the subject of preliminary investigations in the ILE context by, respectively, Dillenbourg and Self (1991) and Baker (1991). Baker ventures a definition of negotiation as "a sequence of dialogue exchanges during which the mental states of the interlocutors are changed from the postures of indifference or conflict with respect to one or more propositions to one of cooperation, and where one or more interlocutor possesses the goal that this posture should be achieved". An agent x is said to cooperate with y's goal that p be eventually true, `Coop(x,y,p)`, if

$$B_x(\text{Goal}(y,\text{possibly}(p)) \;\&\; \text{Prefer}(x,p,\sim p) \rightarrow \text{Persistent-goal}(x,p,\text{Goal}(y,\text{possibly}(p))))$$

i.e. if x's recognition of y's goal that p be true and x's preference of p over ~p results in the generation of a persistent goal for p, relative to y's possession of the goal.

6.4 Interaction and communication

In order to handle interactions beyond the straightforward single-question-single-answer format we need some theory of dialogue. This again is a research field in its own right and one which is of more concern to the instructional component of ILEs than to student modelling. However, such theories have implications for student modelling, as we illustrate here with two examples.

Dialogue game theory is a formal device for generating well-formed sequences of locutionary acts. It is semi-empirical in that it is based partly on analyses of discourse and

partly on abstract specifications of valid processes of reasoning, discussion and argumentation. The theory has three components:

1. A set of 'issue spaces' (Reichman, 1985) or 'commitment stores' describing what each participant believes or is committed to at any given stage of the dialogue;

2. A definition of the set of locutionary events (moves in the dialogue game), with a definition of the changes to the issue spaces when such an event occurs;

3. A set of constraints on the sequence of events, from which is intended to emerge the coherent episodic structure of rational dialogue.

Thus, in an ILE context, dialogue game theory posits a central role for the student model (or issue space) and considers that student model updating occurs as an on-going part of the instructional dialogue, not as a result of some separate diagnostic process. The goals of the 'dialogue game' must of course be generated by the student or the instructional component, which is not of concern here.

Dialogue game theory goes same way - maybe far enough to manage ILE interactions - towards showing how Gricean maxims of conversation "fall out from a general characterisation of the aims and means of linguistic exchanges together with obvious assumptions of rationality of the participants" (Carlson, 1983). We may also attempt to make these "obvious assumptions" explicit, to provide a deeper theory of communication and, in an ILE context, to confirm the need for detailed student models. For example, Cohen and Levesque (1990) present a four-stage derivation of the basis of a theory of communication. The four stages define:

1. Primitives: a set of modal operators intended to define the mental states of the participants. These operators are expressed in a modal logic based on a possible world semantics of knowledge and a situation calculus model of action. The four operators defined are:

`Bel(x,p)` - `p` follows from `x`'s beliefs (this is thus an implicit belief);
`Goal(x,p)` - `p` follows from `x`'s goals;
`Bmb(x,y,p)` - `p` follows from `x`'s beliefs about what is mutually believed by `x` and `y`;
`After(a,p)` - `p` is true in all courses of events that obtain from act `a`'s happening.

These operators are defined through a set of propositions and lemmas, for example, that of 'shared recognition':

```
Bmb(y,x,Goal(x,p)) & Bmb(y,x,Bel(x,Always(p->q)))
              -> Bmb(y,x,Goal(x,q))
```

2. A theory of rational action: a set of propositions defining the properties of ideally rational individual agents with persistent goals, for example, to specify that agents do not knowingly and deliberately make their persistent goals impossible for them to achieve. Theorems may then be derived from such propositions, e.g.

```
∀p Persistent-goal(y,p) & Always(Competent(y,p))
              -> Eventually(p v Bel(y,Always(y,~p))
```

i.e. if an agent has a persistent goal that it is able to bring about then eventually `p` becomes true or it believes that nothing can be done to achieve it.

3. A theory of rational interaction: a set of definitions and propositions intended to characterise interactions between agents. For example, an agent `x` may be said to be sincere or expert with respect to `y` and a proposition `p` under the following conditions:

```
Sincere(x,y,p): Goal(x,Bel(y,p)) -> Goal(x,Know(y,p))
```

```
Expert(x,y,p): Bel(y,Bel(x,p)) -> Bel(y,p))
```

Such definitions of cooperative agents provide formal descriptions of the kinds of behaviour summarised by conversational maxims. No doubt, similar definitions for uncooperative agents could be ventured.

4. A theory of communication: descriptions of communicative acts such as questioning, requesting, etc. derived from general principles of belief and goal adoption between agents. These descriptions enable a distinction between, for example, real questions, rhetorical questions and teacher-student questions. In principle, multi-act utterances and multi-utterance acts can be handled in the same scheme.

Thus, the derivation of communicative acts could be based ultimately upon the kinds of representation of agents' beliefs that we have adopted for student models. Of course, the definition of the content of the various levels is complex, but the intention is that each level be independently motivated, that is, for example, that the notion of a cooperative agent be developed independent of that of communication, and that of rational action be independent of that of interaction. The extent to which such a deep analysis is necessary to support adequate ILE-student interactions in practice remains to be seen.

7. Conclusions

We have reviewed a substantial body of techniques and theories from computer science and AI which may be applied to and adapted for student modelling. We have tried to indicate what particular techniques and theories may contribute by developing a view of student modelling within ILEs seen as systems to support and promote interactions between the belief systems of the agents involved.

We have not considered student modelling to be just a special case of cognitive modelling, emphasising instead that computational utility not cognitive validity provides the main motivation for student modelling. Of course, the techniques and theories considered are justified to some extent by cognitive concerns but they can be developed and analysed independently of their cognitive content - just as a computational linguist may analyse grammars without commitment to their content or any view of human language use: an analogy which led to the coining of the term 'computational mathetics' for the kind of study presented here (Self, 1991b).

Implicit in this analysis is a bias towards 'traditional' symbolic AI as the appropriate basis for student modelling (as opposed to, say, connectionist or situationist approaches). This results from an emphasis on the meta-aspects of ILE interactions, deriving from an assumption that students should not just be able to use knowledge but also be able to reflect upon it, to discuss it, to explain it, etc. For an ILE to participate in such an interaction it would seem to need explicit symbolic representations of that knowledge.

As we have seen, student modelling calls upon many active areas of modern AI. In many cases, student modelling imposes currently unsatisfiable demands on formal AI - for example, to describe the non-monotonic reasoning of the system about the non-monotonic reasoning of the student, to take just one case. Nonetheless, the attempt to clarify what student modelling involves may lead to theoretical and practical benefits in due course.

References

Allen, J.F. (1984). Towards a general theory of action and time, *Artificial Intelligence*, 23, 123-154.

Anderson, A. and Belnap, N. (1975). *Entailment: the Logic of Relevance and Necessity*, Princeton: Princeton University Press.

Baker, M.J. (1991). Negotiating goals in intelligent tutoring dialogues, in E. Costa (ed.), *New Directions for Intelligent Tutoring Systems*, Berlin: Springer-Verlag.
Brazdil, P.B. (1991). Integration of knowledge in multi-agent environments, in E. Costa (ed.), *New Directions for Intelligent Tutoring Systems*, Berlin: Springer-Verlag.
Brown, A. (1987). Metacognition, executive control, self-regulation and other more mysterious mechanisms, in F.E. Weinert and R.H. Kluwe (eds.), *Metacognition, Motivation and Understanding*, Hillsdale, N.J.: Lawrence Erlbaum.
Carlson, L. (1983). *Dialogue Games: an Approach to Discourse Analysis*, Heidelberg: Reidel.
Chi, M., Glaser, R. and Rees, E. (1982). Expertise in problem solving, in R. Sternberg (ed.), *Advances in the Psychology of Human Intelligence*, Hillsdale, N.J.: Lawrence Erlbaum.
Cialdea, M., Micarelli, A., Nardi, D., Spohrer, J. and Aiello, L. (1990). Meta-level reasoning for diagnosis in ITS, Technical Report DIS, University of Rome "La Sapienza".
Clancey, W.J. (1986). Qualitative student models, *Annual Reviews of Computer Science*, 1, 381-450.
Cohen, P.R. and Levesque, H.J. (1990). Rational interaction as the basis for communication, in P.R. Cohen, J. Morgan and M.E. Pollack (eds.), *Intentions in Communication*, Cambridge, Mass.: MIT Press.
Collins, A. and Brown, J.S. (1988). The computer as a tool for learning through reflection, in H. Mandl and A. Lesgold (eds.), *Learning Issues for Intelligent Tutoring Systems*, New York: Springer-Verlag.
Corno, L. and Snow, R. (1986). Adapting teaching to individual differences among learners, in M. Wittrock (ed.), *Handbook of Research on Teaching*, New York: Macmillan.
Costa, E., Duchènoy, S. and Kodratoff, Y. (1988). A resolution based method for discovering students' misconceptions, in J.A. Self (ed.), *Artificial Intelligence and Human Learning*, Chapman and Hall.
Davis, E. (1990). *Representations of Commonsense Knowledge*, Palo Alto: Morgan Kaufmann.
Dewey, J. (1938). *Experience and Education*, New York: Collier.
Dillenbourg, P. and Self, J.A. (1991). Designing human-computer collaborative learning, to appear in C. O'Malley (ed.), *Computer-Supported Collaborative Learning*, Berlin: Springer-Verlag.
Dillenbourg, P. and Self, J.A. (1992). A framework for learner modelling, to appear in *Interactive Learning Environments*.
Donini, F.M., Lenzerini, M., Nardi, D., Pirri, F. and Schaerf, M. (1990). Non-monotonic reasoning, *Artificial Intelligence Review*, 4, 163-210.
Durfee, E.H. and Lesser, V.R. (1989). Negotiating task decomposition and allocation using partial global planning, in L. Gasser and M.N. Huhns (eds.), *Distributed Artificial Intelligence II*, San Mateo: Morgan Kaufmann.
Elsom-Cook, M. (1988). Guided discovery tutoring and bounded user modelling, in J.A. Self (ed.), *Artificial Intelligence and Human Learning*, London: Chapman and Hall.
Etherington, D.W. (1987). Formalizing nonmonotonic reasoning systems, *Artificial Intelligence*, 31, 41-85.
Evertsz, R. (1989). Refining the student's procedural knowledge through abstract interpretations, in D. Bierman, J. Breuker and J. Sandberg (eds.), *Artificial Intelligence and Education*, Amsterdam: IOS Press.
Fagin, R. and Halpern, J.Y. (1987). Belief, awareness, and limited reasoning, *Artificial Intelligence*, 34, 39-76.
Foss, C.L. (1987). Learning from errors in Algebraland, Technical Report IRL-87-003, Institute for Research on Learning, Palo Alto.
Genesereth, M.R. and Nilsson, N.J. (1987). *Logical Foundations of Artificial Intelligence*, Los Altos: Morgan Kaufmann.
Greer, J.E. and McCalla, G.I. (1989). A computational framework for granularity and its application to educational diagnosis, *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, Detroit.

Greiner, R., Smith, B.A. and Wilkerson, R.W. (1989). A correction to the algorithm in Reiter's theory of diagnosis, *Artificial Intelligence*, 41, 79-88.
Hintikka, J. (1962). *Knowledge and Belief*, Ithaca: Cornell University Press.
Holland, J.H., Holyoak, K.J., Nisbett, R.E. and Thagard, P.R. (1986). *Induction: Processes of Inference, Learning and Discovery*, Cambridge, Mass.: MIT Press.
Huang, X., McCalla, G.I., Greer, J.E. and Neufeld, E. (1991). Revising deductive knowledge and stereotypical knowledge in a student model, to appear in *User Modeling and User-Adapted Interaction*.
Ikeda, M., Mizoguchi, R. and Kakusho, O. (1988). Design of a general framework for intelligent tutoring systems, *Proceedings of Intelligent Tutoring Systems 88*, Montreal.
Kass, R. (1989). Building a user model implicitly from a cooperative advisory dialogue, *Proceedings of the Second International Workshop on User Modelling*, Hawaii.
Kautz, H.A. and Allen, J.F. (1986). Generalized plan recognition, *Proceedings of AAAI-86*, 32-37.
Konolige, K. (1988). Reasoning by introspection, in P. Maes and D. Nardi (eds.), *Meta-Level Architectures and Reflection*, Amsterdam: North-Holland.
Langley, P. and Ohlsson, S. (1984). Automated cognitive modelling, *Proceedings of the National Conference on Artificial Intelligence*, Austin.
Levesque, H. (1984). A logic of implicit and explicit belief, Fairchild FLAIR Tech. Report No. 32, Palo Alto.
Levesque, H. (1990). All I know: a study in autoepistemic logic, *Artificial Intelligence*, 42, 263-309.
Maes, P. and Nardi, D., eds. (1988). *Meta-Level Architectures and Reflection*, Amsterdam: North-Holland.
Martins, J.P. and Shapiro, S.C. (1988). A model for belief revision, *Artificial Intelligence*, 35, 25-79.
McCoy, K.F. (1989). Generating context-sensitive responses to object-related misconceptions, *Artificial Intelligence*, 41, 157-195.
Mizoguchi, R., Ikeda, M. and Kakusho, O. (1988). An innovative framework for intelligent tutoring systems, in P. Ercoli and R. Lewis (eds.), *AI Tools in Education*, Amsterdam: North-Holland.
Moyse, R. and Elsom-Cook, M., eds. (1991), *Knowledge Negotiation*, London: Paul Chapman (to appear).
Muggleton, S. and Feng, C. (1990). Efficient induction of logic programs, *Proceedings of the First Conference on Algorithmic Learning*, Tokyo.
Murray, W.R. (1989). Control for intelligent tutoring systems: a blackboard-based dynamic instructional planner, *AI Communications*, 2, 41-57.
Payne, S.J. and Squibb, H.R. (1990). Algebra mal-rules and cognitive accounts of error, *Cognitive Science*, 14, 445-481.
Peachey, D.R. and McCalla, G.I. (1986). Using planning techniques in intelligent tutoring systems, *International Journal of Man-Machine Studies*, 24, 77-98.
Pollock, J.L. (1986). *Contemporary Theories of Knowledge*, London: Hutchinson.
Poole, D.L. (1988). A logical framework for default reasoning, *Artificial Intelligence*, 36, 27-47.
Reichman, R. (1985). *Getting Computers to Talk Like You and Me*, Cambridge, Mass.: MIT Press.
Reiser, B., Anderson, J. and Farrell, G. (1985). Dynamic student modelling in an intelligent tutor for Lisp programming, *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, Los Angeles.
Reiter, R. (1987). A theory of diagnosis from first principles, *Artificial Intelligence*, 32, 57-95.
Schoenfeld, A.H., ed. (1987). *Cognitive Science and Mathematics Education*, Hillsdale, N.J.: Lawrence Erlbaum.
Self, J.A. (1991a). Computational viewpoints, to appear in R. Moyse and M. Elsom-Cook (eds.), *Knowledge Negotiation*, London: Paul Chapman.
Self, J.A. (1991b). Computational mathetics: the missing link in intelligent tutoring systems research, to appear in E. Costa (ed.), *New Directions in Intelligent Tutoring Systems*, Berlin: Springer-Verlag.

Shute, V.J. and Glaser, R. (1990). A large-scale evaluation of an intelligent discovery world: Smithtown, *Interactive Learning Environments*, 1, 51-77.
Taylor, C.D. and Self, J.A. (1990). Monitoring hypertext users, *Interacting with Computers*, 2, 297-312.
van Arragon, P. (1991). Modeling default reasoning using defaults, to appear in *User Modeling and User-Adapted Interaction*.
van Lehn, K. (1989). *Mind Bugs: the Origins of Procedural Misconceptions*, Cambridge, Mass.: MIT Press.
Vygotsky, L.S. (1978). *Mind in Society*, Cambridge, Mass.: Harvard University Press.
Weyhrauch, R. (1980). Prolegomena to a theory of mechanized formal reasoning, *Artificial Intelligence*, 13, 133-170.
Wenger, E. (1987). *Artificial Intelligence and Tutoring Systems*, Los Altos: Morgan Kaufmann.
White, B.Y. and Frederiksen, J.R. (1990). Causal model progressions as a foundation for intelligent learning environments, *Artificial Intelligence*, 42, 99-157.
Wilks, Y. and Ballim, A. (1987). Multiple agents and the heuristic ascription of beliefs, *Proceedings of the 10th International Joint Conference on Artificial Intelligence*, 118-124.
Woolf, B.P. (1987). Representing complex knowledge in an intelligent tutor, *Computational Intelligence*, 3, 45-55.
Young, R.M. and O'Shea, T. (1981). Errors in children's subtraction, *Cognitive Science*, 5, 153-177.