International Journal of Artificial Intelligence in Education (1999), 10, 350-364

The defining characteristics of intelligent tutoring systems research: ITSs care, precisely

John Self, *Computer Based Learning Unit, University of Leeds, Leeds LS2 9JT, England,*
E-mail: jas@cbl.leeds.ac.uk

Based on an invited talk presented at the ITS 98 conference

Abstract: This paper argues that, despite the changes in philosophies and techniques that have occurred since ITS research began, there are continuous threads running through this research which define its essential and distinctive nature. In particular, ITSs are computer-based learning systems which attempt to adapt to the needs of learners and are therefore the only such systems which attempt to 'care' about learners in that sense. Also, ITS research is the only part of the general IT and education field which has as its scientific goal to make computationally precise and explicit forms of educational, psychological and social knowledge which are often left implicit.

INTRODUCTION

The aim of this paper is to review the evolution of the defining characteristics of intelligent tutoring systems research, namely the search for compassion and precision in the design and implementation of computer based learning systems. It is easy to get a superficial view of the way the ITS field has changed. We could, for example, just list the most commonly used keywords in the titles of papers at the ITS 88 and ITS 98 conferences:

ITS 88: expert system, student modelling, problem solving, architecture, planning, ...
ITS 98: training, agents, student modelling, learning environments, collaboration, ...

This immediately tells us that we have moved on from rather general design issues to more applications (especially training rather than school or university learning) with new technologies (especially agents) and new styles of systems (environments and collaborative systems). However, I will risk neglecting new developments, by seeking a longer-term and possibly deeper view, in order to show that, as technologies come and go, there is some underlying continuity and coherence to the field. I will also risk using only examples of my own and my colleagues' work. This is because I want to show that in some respects we have made rather little progress and these examples illustrate this very well.

Figure 1 gives an outline of the paper. I will tackle each of the five ovals in turn.

OVAL 1: FOUNDATIONS (1974-77)

The problem of student modelling has been an issue in ITS research from the beginning (about 1970) until today. The Self (1974) paper assumed as its starting point the tripartite architecture for ITS - the what (domain knowledge), who (student model), and how (tutoring strategy) components which are now part of the encyclopedia definition of what an ITS is. The basic architecture of ITSs was thus already established some 25 years ago.

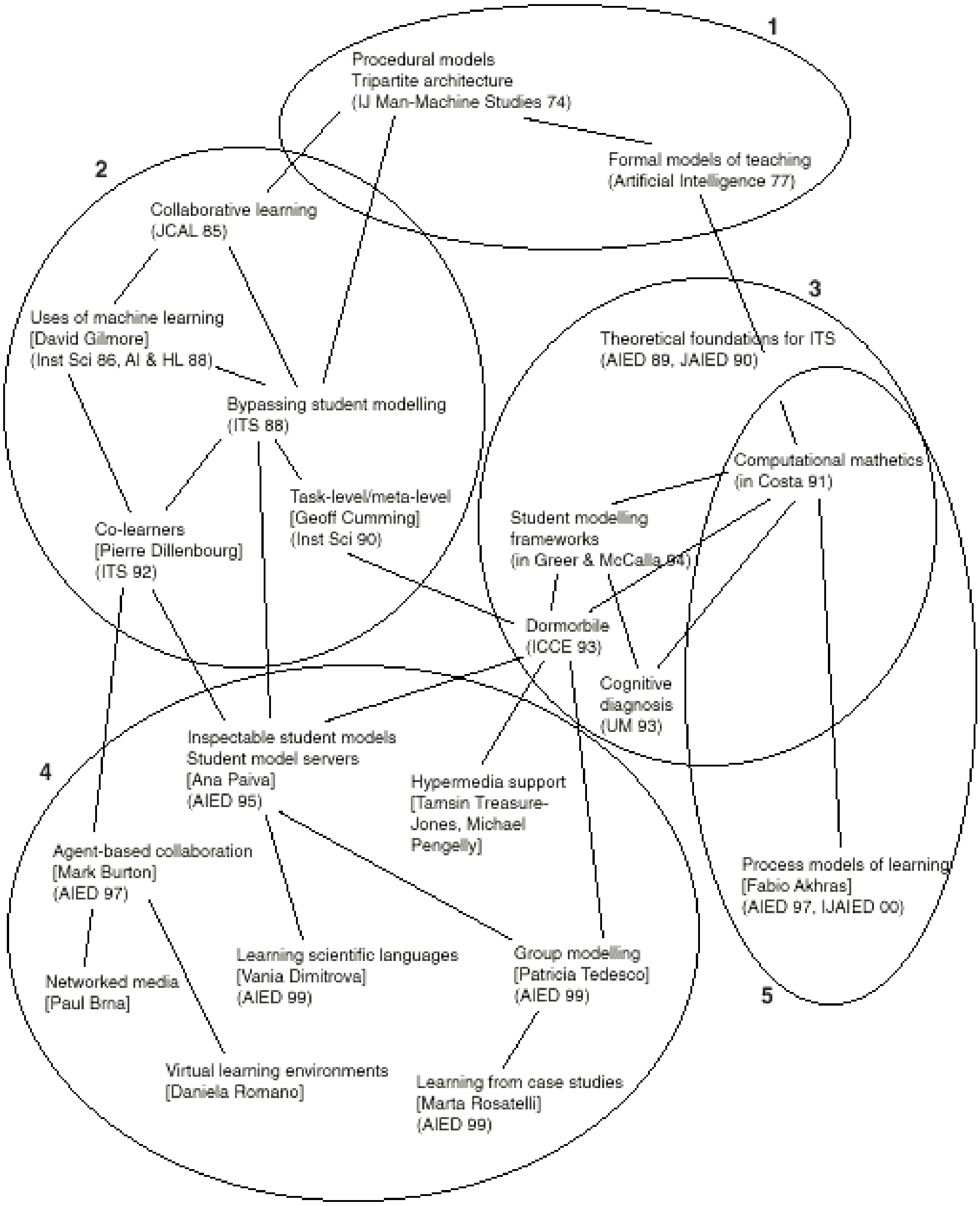

Figure 1. Outline of the paper

The paper developed the distinction between descriptive (or declarative) and predictive (or procedural) student models. This in itself is no great surprise because it was written at the time of the great 'declarative v. procedural controversy' in AI. It was natural to explore its possible implications for student modelling. A student model had previously been regarded as a data structure, representing characteristics of a student and his or her problem-solving performance. If, however, it were to be regarded as a program, describing the steps by which a student solves problems, then it could be used not only descriptively but also predictively, to predict how a student would solve problems in the future (assuming the model were accurate, of course). This would provide potential benefits in terms of monitoring problem-solving performance, providing detailed feedback and developing adaptive tutoring strategies. Of course, the difficulties of such a proposal were also acknowledged, namely, that it would require that we have a sufficiently accurate psychological model of learning to maintain the fidelity of the student model and that it would require addressing the issues being raised in the then fledging research areas of automatic programming and machine learning.

Thus, what I will call the first thread tieing ITS research together was apparent from the beginning, namely:

Thread 1: Who Cares Wins (eventually)

[For the benefit of non-British readers, I probably need to explain that this is a variation on the expression "Who Dares Wins", which I understand is the motto of the British secret services and is much used in TV game shows and the like.]

It seemed rather self-evident that a computer system which cared about an individual student would be better than one which did not. This focus on modelling individual students seemed to be a distinguishing feature of AI-based computer-based learning systems and we can be reasonably proud of the fact that our field was the first to recognise the need for what is now more broadly called user modelling, a term which didn't come into general use until about 1980.

The use of the word 'care' is deliberate. I suggested this in the not-too-serious final session at AIED 97 and now propose it more seriously. Our critics have pushed ITS work off the high moral ground by using positively-loaded terms such as 'constructivist', 'authentic learning', 'learner-centred' and so on to describe alternative approaches. A student model is what enables a system to care about a student. Of course, system designers and teachers care about students too - but somehow our critics have implied that we care less because we want our systems to care more. A system without a student model cannot care about an individual student. If the use of a student model is a defining characteristic of an ITS then ITSs care; non-ITSs do not. Of course, we are using 'care' in a rather special sense. We mean care about what the student knows, misunderstands, wants to do, etc. But there is no harm in basking in the positive glow associated with the term.

The 1977 paper (Self, 1977) further considered the possible benefits of predictive student models. Basically, it proposed to optimise the learning/teaching process through analysis rather than experimentation. The idea is illustrated by Figure 2. A student using a computer-based learning system passes through a series of situations s_1, s_2, .. s_m via a sequence of events. In situation s_m there is a set of possible succeeding events. How may a choice be made between them?

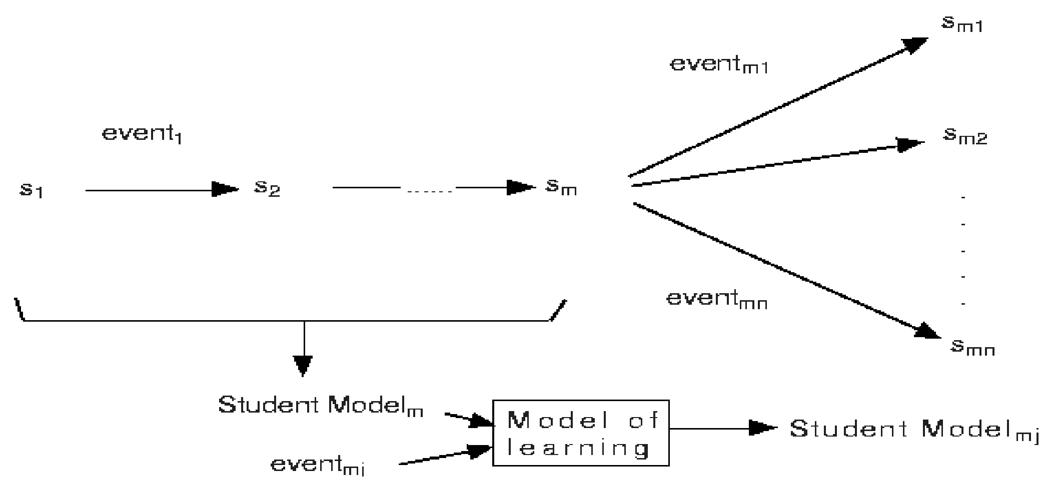

Figure 2. Adaptation through student modelling

We could, in principle, analyse the sequence of situations and events that have already occurred to determine a student model S_m for situation s_m. We could then take each possible event e_{mj} in turn and, using a model of learning, determine the student model S_{mj} which would exist if that event were to happen. We could then consider each of the putative student models S_{mj} and determine which of them might be preferred. Then we could adapt the system so that the event which would lead to that student model actually did occur or was more likely to occur.

This proposal was followed through using two contemporary models of learning, with the following results:

	gain over a random strategy
concept learning	28%
vocabulary learning	172%

A reader paying inadequate attention may misinterpret these figures. These are not figures showing improvements in learner performance as a result of using some computer-based learning system. Indeed, there are no students involved at all. These are entirely theoretical predictions. They say that, if the model of learning is sound, then an adaptive strategy would be better than a random one by the percentages shown. Of course, it is a very big 'if', but the challenge is to make it sufficiently small that the predictions are useful. It might be useful to know, for example, that according to a particular theory of learning an adaptive strategy would provide learning gains which would be so small that an implementation would not be cost-effective.

A further thing to note about these predictions is that they were obtained by computer simulation and not by any kind of formal or mathematical analysis of the models of learning. Mathematics-envy might lead us to hope for axiomatic theories from which predictions may be formally derived. However, theories for any significant kind of learning are likely to be too complex for this to be possible. In any case, there is no fundamental difference between predictions derived by formal analysis and those derived by computer simulation.

This now provides the second thread running through ITS research:

Thread 2: Design is Theory (eventually)

The nature of design is, of course, a controversial question, so I will leave this thread hanging in the air for the moment.

In summary, at the end of the 1970s, the mood of the ITS field was optimistic. A number of demonstration systems had been developed, leading to new contributions to AI itself. The distinctive nature of ITS research had been established. A concerted development programme would now surely show the benefits which would result.

OVAL 2: REACTIONS (1985-92)

The 1985 paper (Self, 1985) presented an alternative to the prevailing expert system and student modelling approach to ITS design. [The 1985 date perhaps needs emphasising: it is well before constructivists and situationists began telling ITS researchers that they were misguided. By then, we already knew we had problems carrying out our programme.] In particular, the paper took on board the growing realisation that student modelling was difficult by considering how we might manage without one. In fact, now I re-read the paper, it is rather more radical: it proposes dispensing with the domain knowledge and tutorial strategies as well! It was (I think now) not a serious proposal towards an ideal computer-based learning system - it was more of a thought experiment to see how we might do things differently.

If one dispenses with the three traditional ITS knowledge bases we have a system which is much like a student - it doesn't know much about the domain or how to learn it. Therefore, there is not much left for the system to do but collaborate with the human learner to try to learn this knowledge. So that, essentially, was the proposal - to build a system to function as a 'computer collaborator' which engaged the student in a genuine shared discovery learning process. The system was not to be 'distracted' by any prior domain knowledge or hidden didactic intent of its own. Since this was to be a shared learning process, the system itself would need to learn from the joint activities, using machine learning techniques.

I regret to say that, try as I might, I cannot find in the paper any educational rationale for this proposal. It is motivated entirely by perceived practical problems: since we were failing to build adequate representations of student models, domain knowledge and tutoring strategies let us try to manage without them.

This proposal, unfounded though it was on any theoretical principles, led to a series of projects, up to Dillenbourg's People Power project, described at ITS 92 (Dillenbourg and Self, 1992). Using Vygotsky's theories of social learning and other influential contemporary work on socially distributed cognition and dialogue (argumentation, negotiation, etc.), he managed to provide a convincing educational basis for computational co-learners. Over the period, the role of machine learning became de-emphasised as it became clear that the co-learner did not need powerful learning techniques, as human learners did not have them either. While all of this is mainstream today, at ITS 92 it was rather novel. At that conference, it was one of only two papers considered to present 'alternatives to one-on-one tutoring'.

However, I have moved out of chronological step. The earlier ITS 88 paper on "bypassing the intractable problem of student modelling" (Self, 1988) has been well-referenced but unfortunately by many who have read only the title (although I am grateful for that). The aim was not to say that student modelling is intractable and should be avoided (the paper was poorly titled!) but to argue that the nature of student modelling needed to be reconceived, away from the form which was being heavily criticised at the time, to make it more tractable. The first thread - the aim of 'caring' systems, supported by student models - remained strong for me.

The pessimistic interpretation of the paper was in keeping with the mood of the ITS field at the end of the 1980s, which one might describe in one word as 'opprobriated' (if it were a word). A great deal of opprobrium was being heaped upon ITS research from all directions, including from within itself, for many leading figures (such as Brown, Wenger, Clancey, Sleeman and Soloway) now considered that AIED research was misguided, relying as it appeared to do on out-moded philosophies of knowledge and learning.

OVAL 3: THEORY (1989-92)

Personally, I retained the earlier optimism - so much so that I day-dreamed further about the possibility of there being firmer theoretical foundations for ITS design (Self, 1990). Design is, of course, a mixture of art and science but for ITS design it seemed to be nearly all art. ITS design did not seem intrinsically more difficult than, say, aircraft design, which is supported by aeronautics - "a blend of beautiful theory and empirical fine tuning" (Shevell, 1983). Aircraft design has progressed through many centuries of visions and a few decades of serious experimentation to largely depend on the theory of aeronautics and specialised test environments. Would ITSs ever be built by a blend of beautiful theory and empirical fine tuning?

Figure 3 (from Self, 1990) summarises the proposal. Conventional ITS design proceeds through the (informal) derivation of imprecise design principles from imprecise theories of learning and interaction. From these principles, we then (informally) derive experimental systems, which are empirically evaluated. Through many iterations, the results of these evaluations lead to revised theories, principles and systems. Instead, we might have precise, axiomatic theories from which we may (formally) derive principles and systems, which, if the theories are sound, must (by definition) be reliable and hence only need to be demonstrated. In practice, of course, no theory would be sound enough to not require empirical fine-tuning and, in reality, we could imagine this process being applicable to only certain components of the whole ITS design, leading to a mixture of theory and empiricism.

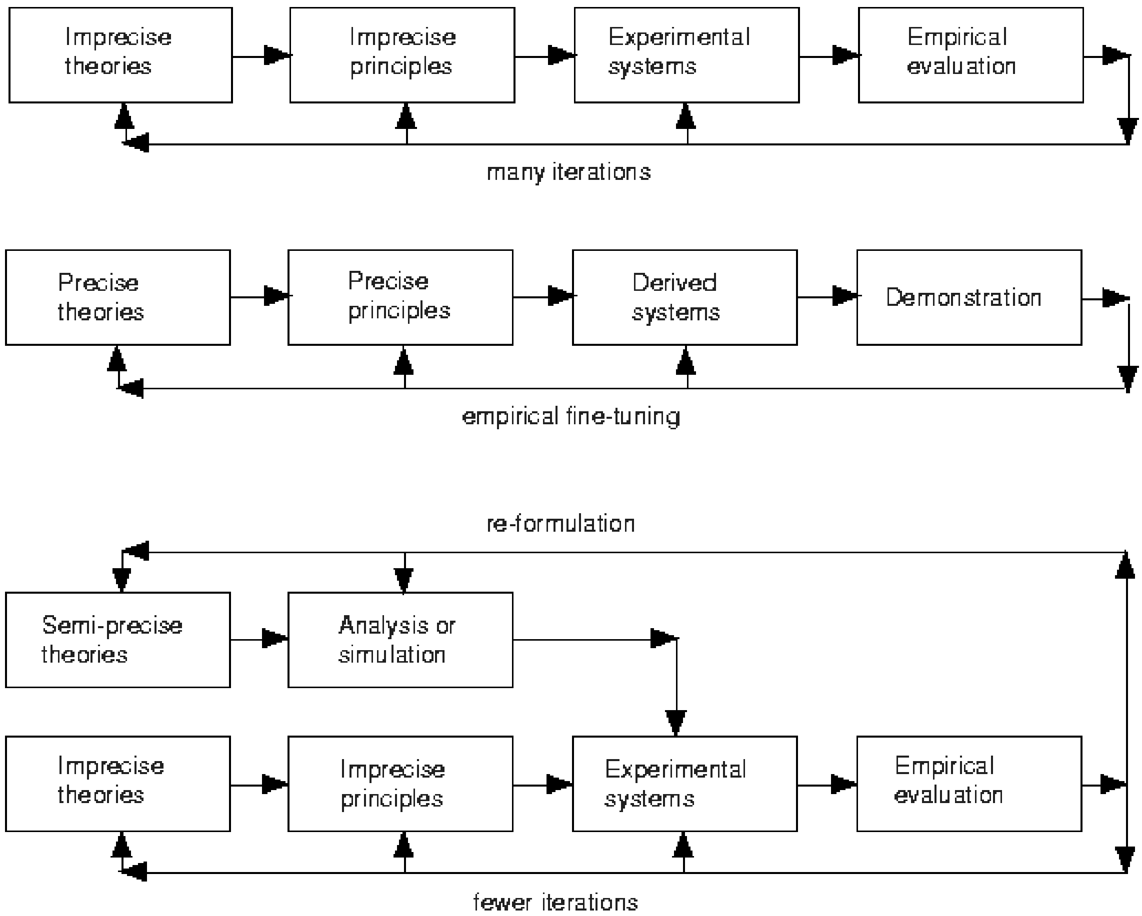

Figure 3. The ITS design process

The early ACT tutors provided the most refined design methodology of the time and an interesting case study. The ACT theory of cognition and learning (1985 vintage) provided an exceptionally detailed theory but one which nonetheless was not detailed enough to provide rigorous derivations of design principles. Informal argumentation led instead to eight, rather imprecise, principles, which themselves are rather indirectly related to the implemented systems (Anderson, Boyle, Farrell and Reiser, 1989). The infeasibility of any rigorous design process was, of course, recognised and indeed it was the whole point of the ACT tutor programme to provide a test-bed for the ACT theory and to use empirical studies to feed back into a revised theory.

The following simple illustration indicates the alternative approach. First, we need to formalise a theory of learning, perhaps as a set of statements describing how the cognitive state of a learner changes as a consequence of instructional actions or other events. In general, then, it might be defined by a set of axioms of the form:

```
State(s,s1,t) & Event(e,x) -> State(s,s2,e(x,t))
```

i.e. if the student s is in cognitive state s1 in situation t and e is an event with parameters x then she will be in state s2 in the situation reached after event e, that is, e(x,t).

For example, an impasse-based theory of learning from examples might have the following axioms (simplifying, of course):

```
1. State(s,empty,t) & Event(present-positive,[p,f,a]) ->
     State(s,Believes(s,f->a),present-positive([p,f,a],t))
```

i.e. if the student knows nothing then if we present a positive example p which has feature f and where action a was used to solve the problem, then the student will come to believe that any problem with feature f can be solved by applying action a.

2. `State(s,empty,t) & Event(present-positive,[p,[f,g],a]) ->`
   `State(s,Believes(s,f&g->a),present-positive([p,[f,g],a],t))`

i.e. similarly, if the positive example has two features f and g then the student will believe that action a is appropriate for problems with both features.

3. `State(s,s1,t) & Event(question,[q,fs]) & Impasse(s1,q,fs) ->`
   `State(s,generalisation(s1),question([q,fs],t))`

i.e. if the student is presented a question q with features fs which leads to an impasse then she generalises her cognitive state (the basic assumption of an impasse-based theory of learning being that the student will attempt a generalisation to overcome the impasse).

4. `State(s,s1,t) & Event(question,[q,fs]) & Solves(s1,q,fs) ->`
   `State(s,s1,question([q,fs],t))`

i.e. if the question is solved then the student's state is unchanged.

We also need definitions of the predicates Impasse and Solves and the function generalisation, which might include, for example, the axioms:

5. `Impasse(Believes(s,f&g->a),q,f)`

i.e. if the student believes only the rule f&g->a and the question has only feature f then she reaches an impasse.

6. `Solves(Believes(s,f->a),q,f)`

i.e. if the student believes f->a and the question has only that feature then she will solve the problem.

7. `generalisation(Believes(s,f&g->a)) =`
   `Believes(s,f->a) v Believes(s,g->a)`

i.e. by generalising, the student will believe that one or other feature alone is sufficient for the action.

A set of such axioms and associated definitions constitutes, in our terms, a theory of learning. We also need to define the set of possible events. Let us imagine that we have just two example problem solutions `p1` and `p2`, the first with feature `f1` and the second with features `f1` and `f2`, both where action a1 was applied. (To be concrete, `p1` could be 745-127 with `f1=borrow-next-left`, `p2` could be 85-27 with `f1=borrow-next-left` and `f2 = borrow-leftmost`, and `a1 = borrow`). If we assume two similar problems could also be presented as questions, we have four possible events:

8. `Event(present-positive,[p1,f1,a1])`
9. `Event(present-positive,[p2,[f1,f2],a1])`
10. `Event(question,[q1,f1])`
11. `Event(question,[q2,[f1,f2]])`

Let us assume the student has no prior knowledge (her cognitive state is empty at time `t0`):

12. `State(s,empty,t0)`

A set of such axioms, defining a learning theory, the events available and the initial state, can now be used to formally derive a plan for achieving some goal. For example, imagine that we wished to determine what the learner may believe after any two events (perhaps we anticipate that a class session has time for only two examples or questions and we wish to determine which ones are most productive). This goal can be expressed as:

```
State(s,x,e2(x2,e1(x1,t0)))
```

i.e. what cognitive state x might the student reach after any two events e1 and e2?

From the twelve premises above, just three solutions may be derived:

```
State(s,Believes(f1->a1),question([q1,f1],
   present-positive([p1,f1,a],t0)))
```

i.e. if we present the positive example p1 followed by question q1 then the student will believe `f1->a1`.

```
State(s,Believes(f1&f2->a1),
  question([q2,[f1,f2]],present-positive([p2,[f1,f2],a1],t0)))
```

i.e. if we present the positive example p2 followed by question q2 then the student will have an over-specialised rule.

```
State(s,Believes(f1->a1) v Believes(f2->a1),
  question([q1,f1],present-positive([p2,[f1,f2],a1],t0)))
```

i.e. if we present the positive example p2 followed by question q1 then the student will be confused as to which feature requires a1 to be used.

This simple formulation re-expresses the argument presented (informally) in VanLehn (1987) that we should present three-column subtraction before two-column subtraction. The proof is general (in the sense that it could be applied to any similar problem-solving situation) and the conclusion could be presented as an 'instructional guideline' rigorously derived from a particular theory of learning, in this case, that examples should make it clear what features of a problem necessitate a particular action. The derivation of such a proof is a general theorem-proving process and could, in principle, be carried out by automatic means. The general aim, then, is to express the theory of learning, events, learner attributes and instructional goals formally so that an instructional theory may be presented as a set of conclusions derived rigorously from assumptions, rather than as a set of vaguely expressed guidelines.

The above illustration is much too simple to convince anyone that the methodology is feasible. Premises to describe the effect of some instructional actions are missing; the description of the learner and goals are too simple; we should represent some kind of probabilistic analysis, as the effects of events are not so definite; and so on. However, simplified though it is, the specific conclusion drawn is one which was originally presented as justification for developing a detailed learning theory to be implemented as a computer program so that its properties may be studied. When the premises become more realistically complex, derivations will require computational assistance and thus there is no fundamental difference between an axiomatic analysis such as the above and a computational simulation. The issue is which mode of description is most clear, concise, precise, convincing, and ultimately successful in leading to the specification of ITS designs.

The suggestion that ITS design should be based on formal theories was not taken seriously, even by me, papers on the topic (e.g. Self, 1992) having a rather frivolous tone. The suggestion that we develop more formal and precise theories of what we are doing, to guide our designs - rather than implement systems based on our intuitions and then see how well they work - did not, and does not, appeal. It is much more fun to just try things out. The term 'computational mathetics' was introduced only with the apology that it is "impressively pretentious". Originally, it was a private joke - a term intended to keep gullible people quiet. If you are asked what your interests are and answer "AI" and "education" then everyone will give you the benefit of their thoughts on those topics. If you say "computational mathetics" they will go away and bother someone else. I recommend it to you. Now, however, I believe that (what is meant by) computational mathetics is an important objective for ITS research - and from a scientific point of view, the most important one.

Occasionally, less gullible people will ask "what's that?" on hearing "computational mathetics" and it is necessary to have an answer, which is summarised by the following dictionary entries:

mathetic, adj. pertaining to learning.
mathetics, n. the study of matters pertaining to learning.
computational mathetics, np. the computational study of matters pertaining to learning, i.e. the use of techniques, concepts and methodologies of computing to study (and support) learning.

The first word is in all good dictionaries. The second should be. The third is a phrase invented in rough analogy to computational linguistics. If we compare:

linguistics mathetics
psycholinguistics psychomathetics
sociolinguistics sociomathetics
computational linguistics computational mathetics
applied linguistics applied mathetics
... ...

then we can see that the analogies to perfectly respectable sub-fields of linguistics all exist in our own field and that many of our disagreements are due to confusions about the sub-fields we are working in.

Over the years, we carried out a few feeble attempts at computational mathetics:

- frameworks for student modelling, using modal logics (Self, 1994)
- open learner models, using belief revision (Paiva, Self amd Hartley, 1995)
- cognitive diagnosis, using GDE, the general diagnosis engine (Self, 1993)
- DORMORBILE, using meta-programming, discussed briefly below (Self, 1995)
- process models of learning, using situation calculus (Akhras and Self, to appear)

This work has been characterised by a naive hope that, just as aeronautics was based on existing notations of physics, mechanics, hydraulics, etc., so computational mathetics might be created from existing notations within artificial intelligence.

OVAL 4: SOME CURRENT PROJECTS

In this section I would like to briefly describe some current projects at Leeds. I am sure my colleagues are focussed only on the timeliness of their research but I like to imagine that there is some thread of continuity running through these projects and earlier work.

Interactive diagnosis based on conceptual graphs

This project (Dimitrova, Self and Brna, 1999) continues the line of work which considers the learner to be a participant in the learner modelling process. The aim is to move from menu-based collaborative diagnosis towards supporting a more negotiational dialogue about the learner model and enabling the learner to directly manipulate the model. This project is being carried out in the domain of non-native English speakers learning English scientific or technical terminology (e.g. the terms of computer science or financial management). Conceptual graphs are being used as a graphical external representation, providing the means through and with which learners communicate.

Mediating strategic conflicts in group problem solving

There has been a lot of work on cognitive conflicts (under the heading of 'belief revision' in AI) but possibly of more relevance to ITS are 'metacognitive conflicts', that is, conflicts over strategic, monitoring and reflective processes. This project (Tedesco, 1999) considers a group of students addressing some planning task, such as deciding upon a sequence of actions to achieve some goal, involving perhaps the creation of a PERT network. The aim is to formalise descriptions of the various metacognitive conflicts that arise and use these as the basis for designing a system to mediate discussions among the group.

Studying, teaching and understanding research methodologies

The aim of the STURM project (Treasure-Jones, Pengelly and Self, 1999) is to develop a computer-based advisor for a student given a rather open-ended essay assignment, involving some research of web-based materials and the structuring of an essay, for example, an arts undergraduate asked to write an essay on "In what sense were Beethoven and Schumann romantic composers". The student's research strategies are modelled using the Dormorbile framework, particularly the higher-levels of the framework, enabling the system to provide relatively domain-independent strategic advice.

Web-based distance learning from case studies by groups

The case study method, involving open-ended problems based on realistic situations which are tackled by groups over extended periods, is a well-established educational activity but one which has been relatively neglected in ITS research. It may, however, be better suited to web-based distance learning than the kinds of narrow problem-solving of conventional ITSs. Students may browse the web for material related to the case study and then come together for on-line discussions on the ideas developed, leading to a coordinated set of off-line and on-line sessions. This project (Rosatelli and Self, 1999) aims to support the group through a general 'seven steps' methodology for case study activity (Easton, 1982). The system's role is to monitor the group's transition through the seven steps and to promote the group discussion.

Collaborative learning as realised in simple simulated agents

The aim of this project (Burton, Brna and Pilkington, 1999) is develop a computer-based laboratory for studying collaboration, in order to determine the determine the functionality of agents acting as collaborators with human learners. A theory of collaboration as the dynamic distribution of dialogue roles is the basis for a computational model which can be explored to help determine how students and simulated collaborative should behave in order to benefit the most from collaborative activity.

Virtual learning environments (for firefighter decision-making)

There are always inflated expectations about the educational applicability of a new technology, when it is necessary to pause to reflect on the precise educational benefits that the technology may provide. In the case of 3D VR, it seems to be most relevant when it is essential to provide a sense of presence for learners developing decision-making skills in stressful, dynamic situations, for example, a group of firefighters. As this is a group training process, involving collaboration and communication, and there is a need for agents to simulate members of the group, there are overlaps with the projects mentioned above.

Networked interactive media in schools

This project (with the University of Duisburg and MediaWorld of Germany and INESC of Portugal) is the most technology-driven of this set, being funded by the European Union to explore the potential of innovative, intelligent interfaces in primary schools (4-8 year olds).

The aim is to introduce networked computers and interactive whiteboards into ordinary classrooms to support a group of children cooperating in a process of 'multimedia authoring' to create, for example, stories (Hoppe, 1999). There will be a need for explicit (animated) agents within the stories and for implicit agents monitoring and supporting the group interactions.

OVAL 5: A PROCESS MODEL OF LEARNING

Finally, I will consider a recent project (Akhras and Self, to appear) in which many of the themes come together to provide a rather novel view of the nature of adaptive learning environments and in which we investigate whether the twin threads of ITS research, compassion and precision, really are irreconcilable with the views of constructivists, who hold ITSs in disdain.

One of the basic assumptions of constructivism is that knowledge cannot be objectively defined and statically represented (as it is in conventional student models). Instead, it is individually constructed from what learners do in their experiential worlds. By means of acting in a world, learners assimilate new concepts to their previously constructed cognitive structures or modify their cognitive structures to accommodate interpretations of the new experiences. It follows that knowing and doing cannot be separated and that the activity and context of an experience become an integral part of the meaning of that experience. Therefore, the focus is on the process by which knowledge is constructed rather than on a target domain knowledge to be acquired.

The more recent discussions of the nature of learning and knowledge concern the properties which a process of learning should have to be conducive to learning. This then leads to suggestions that designers should design systems which enable such desirable properties to hold. The perceived properties of a process of learning depend on what the student already believes and what she does while interacting with the system. Therefore, systems may need to be able to adapt to try to ensure that desirable properties hold. To enable such adaptations, a system needs to model the properties of the interactions between system and student.

Overall, the aim is to model the process of learning as one that happens over time, through interactions between cognitive structures and context, and through activity. We need to describe the environmental contexts in which interactions occur and the events which are possible in a situation - their preconditions and their effects. A series of events creates a sequence of situations which we may consider to be a 'course of interaction'.

Various properties of a course of interaction may now be defined. Of the many properties that may be defined, some may be considered 'better' than others. That is, learning theorists may have argued that, or perhaps provided experimental evidence that, a course of interaction with a particular property is more likely to lead to learning than a course of interaction without that property.

After a particular course of interaction, we have passed through a sequence of situations $\{s_1,s_2,..s_m\}$ by means of a series of events $\{e_1,e_2,..e_{m-1}\}$ (Figure 4). In the situation s_m reached, a set of events $\{e_{m1},e_{m2},..e_{mn}\}$ are possible. If a particular event e_{mj} were to occur, then a new situation s_{mj} would be reached. The (potential) course of interaction $\{s_1,s_2, .. s_m, s_{mj}\}$ would possess a set of properties $P_{mj} = \{p_1,p_2,..p_r\}$. We may say that the situation s_m 'affords' each of these properties p_i through event e_{mj}.

Clearly, some of the potential courses of interaction will possess more desirable properties than others. Therefore, the system could seek to adapt itself to ensure than an event occurred which led to the desired properties. In general, there is a set of potential properties available, each of which is more or less desirable than others, and there needs to be some heuristic strategy for guiding the student towards events which, on balance, are more likely to lead the learner to encounter contexts which allow a continuing learning experience. This general scheme has been elaborated in considerable detail in Akhras (1998), where complicated constructivist properties of cumulativeness, constructiveness, self-regulatedness and reflectiveness are defined.

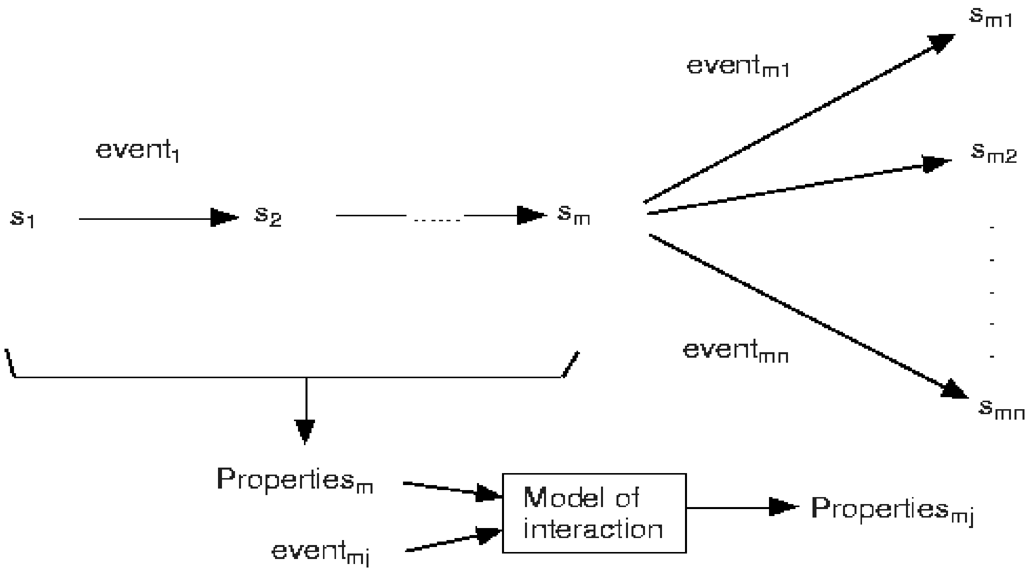

Figure 4. Adaptation through interaction modelling

You may notice more than a passing similarity between Figure 4 and Figure 2. What has changed? We now have no explicit representation of the student's knowledge, beliefs, etc. (that is, no student model as usually conceived). Instead we focus on the events that have occurred. We say that if a 'good' course of interaction occurs then the student is more likely to learn - but we do not attempt to represent precisely what will be learned. This seems closer to the spirit of constructivism in that the 'content' of learning is intimately related to the 'activity' of learning, and what is actually learned is a construction of the learner from the activities she has carried out. In our case, we have no target knowledge we are aiming for - we 'simply' seek to ensure that the learner passes through courses of interaction considered conducive to learning.

What has happened to the conventional tripartite division of ITSs? The architecture that emerges is not one which is fundamentally opposed to the standard ITS architecture or any other. It is simply one which focusses on the different set of issues which arise from the different philosophy. In particular, it is clear that the attempt to develop intelligent systems to support learning is not inherently contradictory to a constructivist view of learning. In order to clarify how this is so we may re-consider the traditional three ITS components.

The model of domain knowledge

An ITS designer tends to assume that knowledge can be described in terms of facts, principles, and so on, which can be represented symbolically and hierarchically and learned in an incremental fashion. Therefore, he invests his effort on developing complex representations of such knowledge. Constructivists emphasise that learners construct their own knowledge through interpreting their experiences in interactional contexts. Therefore, the designer of a constructively-oriented learning environment focusses not on knowledge representations but on the nature of situations, contexts and interactions. This leads to a consideration of the 'content' of contexts and of the dynamics of the learning process. However, a 'situation model' may well contain representations of aspects of the domain of knowledge which a learner may access during interactions. These representations may appear similar to the models of domain knowledge in ITSs but their purpose may be very different. They are not descriptions of target knowledge but descriptions of resources which are available in a learning situation. From this perspective, a model of domain knowledge may be seen as a subset of the broader notion of a situation model.

The model of the learner's knowledge

Typically, the student model of an ITS is determined by analysing the student's interactions with reference to the model of domain knowledge in order to determine gaps or errors which may form the basis for instructional interventions. If, however, as constructivists argue, the student's individual constructive process (leading to personal constructions perhaps unrelated to any target knowledge) is more important than the particular product of any learning process, then our model of the learner should focus more on the interactive process, extended in time, taking into account the learner's actions, the contexts in which they occurred, and the learner's cognitive structures at the time. Developing such an 'interaction process model' enables us to consider the kinds of regularities of interaction sequences which lead to properties which benefit or hinder learning. As we have indicated, the learner's cognitive structures may form part of the descriptions of the time-extended interactive process, for the significance of interactive events may depend on individual cognitions. However, the aspects of cognitive structures to be considered are of a different nature and assume different roles than in standard ITSs, as they are taken in relation to the context and activity that constitute learning interactions. Thus, again, the notion of an interaction process model is, in a way, a superset of (rather than in opposition to) that of an ITS-style student model.

The model of teaching knowledge

ITS designers consider that their systems should, more or less deterministically, determine instructional plans by interpreting their student model with respect to a curriculum structure based on the model of domain knowledge. Constructivists would argue that the learning process is too unpredictable to be amenable to analysis by pre-specified structures and that learning sequences emerge from interactions between the learner and the environment as influenced by the opportunities that become available. Therefore, according to the latter view, the pedagogical role of the system is not to determine instructional events but to provide profitable spaces for interaction to the learner based on some model of the affordances of potential situations. In the previous section, the 'affordance model' focussed on the particular properties of courses of interaction afforded. This was because, according to the constructivist view, the process of learning was more important than the product. However, if one had an objectivist view of knowledge one could develop an affordance model in terms of the 'items of knowledge' which may be learned through particular events (so, for example, an ITS event such as the presentation of remediational feedback might be considered to afford the learning of the item of knowledge remediated). So, again, we see the affordance model as broader than, not opposed to, the model of teaching as curriculum-based planning.

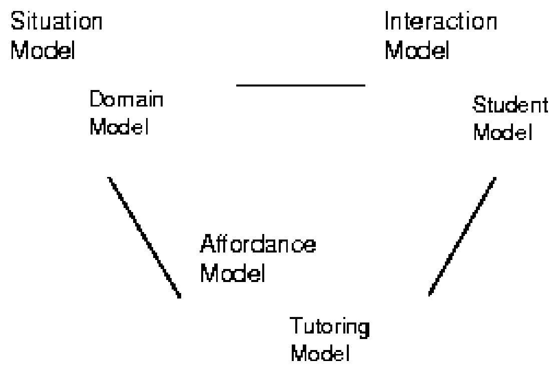

Figure 5. The components of ITSs

So, at the risk of over-simplifying, we may present a tripartite model of the architecture of computer-based learning environments which includes the standard ITS architecture as a subset (Figure 5). Like all simple diagrams, it risks misleading the reader - in this case, by suggesting that the design and implementation of such systems is much, much harder than designing ITSs (which are hard enough) because the ITS architecture is shown as a small subset. In fact, the content of these components may be much simpler to implement than those we strive for in ITSs. For example, the kind of student interaction process model we may need, based on properties of courses of interaction, may be easier to construct than the high-fidelity cognitive models based on detailed considerations of knowledge structures which we need in conventional ITSs.

CONCLUDING COMMENT

I hope to have indicated that, despite the changes in philosophies and technologies, there are continuous threads running through ITS research which define its essential nature. While I have characterised the mood of the ITS field at the end of previous decades as optimistic and then opprobriated, perhaps an appropriate word for the end of the present decade is 'opportunistic'. The field is no longer so pure that it will resist co-opting philosophies and technologies wherever they come from. This naturally leads to an active, multi-faceted research field but it masks the underlying coherence. The real opportunity for the next decade is to tie together the two threads - to design systems that care about students and have a degree of computational precision - and thereby provide a unique scientific and technical contribution.

References

(I apologise for the embarrassingly Self-referential nature of this list of references. Such a degree of egotism is (perhaps) hidden in a talk but appears rather blatant on paper.)

Akhras, F.N. (1998). Reasoning about process in intelligent learning environments, unpublished PhD thesis, University of Leeds.

Akhras, F.N. and Self, J.A. (to appear), System intelligence in constructivist learning, *International Journal of Artificial Intelligence in Education*.

Anderson, J.R., Boyle, C.F., Farrell, R. and Reiser, B.J. (1989). Cognitive principles in the design of computer tutors, in P. Morris (ed.), *Modelling Cognition*, New York: Wiley.

Burton, M., Brna, P. and Pilkington, R. (1999). Clarissa: a laboratory for the modelling of collaboration (under review).

Dillenbourg, P. and Self, J.A. (1992). People Power: a human-computer collaborative learning system, in C. Frasson, G. Gauthier and G.I. McCalla (eds.), *Intelligent Tutoring Systems*, Berlin: Springer-Verlag, 651-660.

Dimitrova, V., Self, J. and Brna, P. (1999). The interactive maintenance of open learner models, *Proceedings of the 9th International Conference on Artificial Intelligence in Education*, 405-412.

Easton, G. (1982). *Learning from Case Studies*, London: Prentice-Hall.

Hoppe, H.U. (1999). Collaborative learning in open distributed environments - pedagogical principles and computational methods, *Proceedings of the 9th International Conference on Artificial Intelligence in Education*, 8.

Paiva, A.M., Self, J.A. and Hartley, J.R. (1995). Externalising learner models, *Proceedings of the 7th World Conference on Artificial Intelligence in Education*, Washington, DC, 509-516.

Rosatelli, M.C. and Self, J.A. (1999). Supporting distance learning from case studies, *Proceedings of the 9th International Conference on Artificial Intelligence in Education*, 457-464.

Self, J.A. (1974). Student models in computer-aided instruction, *International Journal of Man-Machine Studies*, 6, 261-276.

Self, J.A. (1977). Concept teaching, *Artificial Intelligence*, 9, 197-221.

Self, J.A. (1985). A perspective on intelligent computer-aided instruction, *Journal of Computer Assisted Learning*, 1, 159-166.

Self, J.A. (1988). Bypassing the intractable problem of student modelling, *Proceedings of Intelligent Tutoring Systems 88*, Montreal, Canada.

Self, J.A. (1990). Theoretical foundations of intelligent tutoring systems, *Journal of Artificial Intelligence in Education*, 1, 4, 3-14.

Self, J.A. (1992). Computational mathetics: the missing link in intelligent tutoring systems research, in E. Costa (ed.), *New Directions for Intelligent Tutoring Systems*, Berlin: Springer-Verlag, 295-352.

Self, J.A. (1993). Model-based cognitive diagnosis, *User Modeling and User-Adapted Interaction*, 3, 89-106.

Self, J.A. (1994). Formal approaches to student modelling, in J.E. Greer and G.I. McCalla (eds.), *Student Modelling: the Key to Individualized Knowledge-Based Instruction*, Berlin: Springer-Verlag, 295-352.

Self, J.A. (1995). Dormorbile: a vehicle for metacognition, in T.-W. Chan and J.A. Self (eds.), *Emerging Computer Technologies in Education*, Charlottesville: AACE.

Tedesco, P. (1999). Mediating meta-cognitive conflicts in a collaborative problem-solving environment, *Proceedings Young Researchers' Track, AIED 99,* 43-44.

Treasure-Jones, T., Pengelly, M. and Self, J. (1999). Supporting essay preparation: strategic advice for users of hypermedia databases, Technical Report 99/15, Computer Based Learning Unit, University of Leeds.

Shevell, R.S. (1983). *Fundamentals of Flight*, Englewood Cliffs: Prentice-Hall.

VanLehn, K. (1987). Learning one subprocedure per lesson, *Artificial Intelligence*, 31, 1-40.