

**Research
Paper**

An Alternative Architecture for Intelligent Tutoring Systems

PETROS A. M. GELEPITHIS AND ROBIN GOODFELLOW
Kingston Polytechnic, Kingston Upon Thames, Surrey KT1 2EE, U.K.

ABSTRACT

In this paper we present an alternative architecture for intelligent tutoring systems that is based on: (1) the centrality of the Understanding/Explanation processes for tutoring and (2) the *learner-tutor duality*, that is, the fact that both learners and teachers have, in varying degrees, all the features that traditionally are assigned to only one or the other of the two protagonists of the tutoring (e.g. educational, training) process.

The proposed architecture consists of the learner and tutor models and its interface. The functional core of the system is a dual Understanding/Explanation module. This is characterized by the Understanding/Explanation matrix which is used to determine the evolving cognitive state of both the learner and tutor models during a session.

KEY WORDS Intelligent tutoring architectures Understanding Explanation

INTRODUCTION

Following Carbonell's (1970) 'intelligent' approach to CAI, many tutoring systems have been implemented and considerable theoretical work has been carried out under the banner of ITS (e.g. Murray, 1989; Ohlsson, 1987; Self, 1987). Current generic architectures for ITS (e.g. Elsom-Cook, 1990; Sleeman, 1987; Pitacco and Cerri, 1989) are characterized, to a greater or lesser extent, by the 'trinity' ITS architecture,¹ i.e. three separate modules for the representation of knowledge: a domain expertise module, a student module, and a pedagogical expertise module.

Although many modifications of each and every module within the trinity architecture have been proposed and, in some cases, at least partially, implemented alternatives to the trinity architecture itself had not so far been suggested.² This paper outlines an alternative ITS architecture that is based on:

- (1) The learner-tutor duality; and
- (2) The centrality of the Understanding/Explanation processes for tutoring.

¹ A notable exception is Clancey (1987), who chooses to identify an ITS with an ES plus a Teaching Knowledge module.

² For some criticisms of the trinity model see Self (1988); Elsom-Cook's (1990) guided discovery framework is more of an attempt to focus upon the nature of the educational process and much less, if at all, to provide an alternative to the trinity architecture. This, of course, is not to deny the potential influence of his framework on an ITS architecture.

Each of these observations leads to a characteristics feature differentiating our proposed architecture from the traditional ITS one. The following paragraphs introduce these characteristic features of our design.

Teaching and learning (including apprenticeship) are not 'one-way' processes. It is *not* the case that the teacher teaches only and the learner learns only. Many times it is the learner who teaches (even if indirectly) the teacher, and similarly, many times the trainer enhances his skills via his trainee apprenticeship. In other words, both learners and tutors have, in varying degrees, all the features that traditionally are assigned to only one or the other of the two protagonists of the educational (e.g. training) process. This fundamental feature of the educational process we have called the *learner-tutor duality*. In system design terms, this duality implies two models: one for the learner (e.g. student, trainee) and one for the tutor (e.g. teacher, trainer). Each of these two models (let us call them the learner and tutor models, respectively) aims to provide an approximate representation of the learner's (correspondingly tutor's) cognitive state specific to the learning situation at hand.

So, pedagogic expertise is traditionally believed to be the province of the teacher only. We believe that the learner also has some pedagogic expertise, even if this is not codified or explicit, which does influence his questions, responses, etc. Thus, the traditional pedagogic expertise module (or subsystem) should be substituted by two separate pedagogic expertise subsystems: one each for the learner and tutor models, respectively. Furthermore, although the knowledge of a teacher and learner on a particular topic (or skill to be acquired) are significantly different, they *both* contribute substantial gnostic experience in order to reach the sought tutoring aims, objectives, and goals. Therefore, the traditional domain expertise module (or subsystem) should be substituted by two separate sets of knowledge bases: one each for the learner and tutor models, respectively.

The question now is: What is the central, fundamental driving force shaping the structuring of these models? We submit that the processes of understanding and explanation are uniquely qualified to play this role. Clearly, the question has both a theoretical and an applied aspect. First, a case must be made, given the definitions of the learner and tutor models above, that understanding and explanation are indeed the central processes shaping the structure of at least conscious cognition. Second, the theoretical case having been made or assumed, a demonstration of how these processes could be used to shape the structure of the learner and tutor models. The aim of this paper is to provide such a demonstration. Specifically, we link the processes of understanding and explanation into an Understanding/Explanation (U/E) subsystem constituting the backbone of our architecture. This constitutes the heart of both the learner and tutor models. Its structure provides the hooks for the functional integration of all other modules within the learner and tutor models as well as the interaction link, through the system's interface between the learner and tutor models. Figure 1 illustrates the *learner-tutor duality* and the centrality of the U/E subsystem. An outline description of the overall architecture is given in the next section whereas the third section describes the U/E system itself.³

FUNCTIONAL OUTLINE

As stated in the introduction, the system has as its backbone the U/E subsystems which structure the behaviour of both learner and tutor models. These two models are each composed of a number of processing modules and knowledge bases controlled by the U/E subsystem. The two sets of processing modules comprise, respectively, the Learner's Pedagogic Expertise Subsystem (PES) and the Tutor's PES.

The input to the U/E subsystem may be misunderstood (with respect to the intended meaning

³ For a more detailed description and the industrial relevance of this work see Goodfellow and Gelepithis (1991).

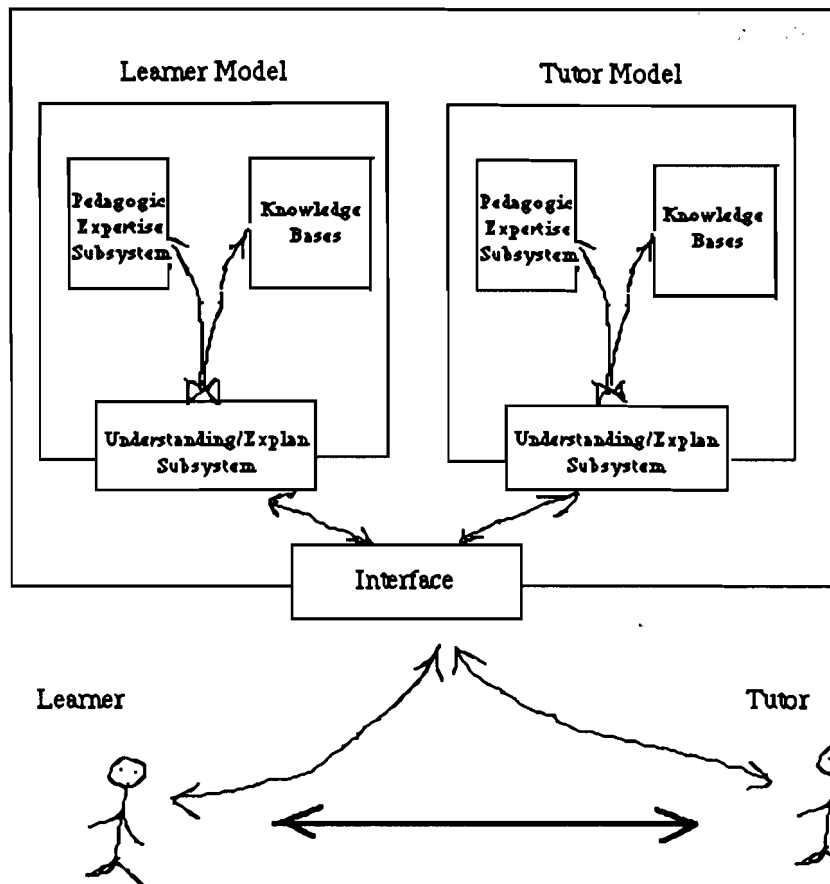


Figure 1. Top level of an alternative ITS architecture.

of the input) or understood (i.e. being reducible to the Learner's current set of premises). In either case, the cognitive state represented by, say, the Learner Model will be modified and an appropriate behaviour (output to the systems interface) will occur which may influence any or all of the other three systems (i.e. Tutor Model, Learner, Tutor). In particular, advancement of the Learner Model's Understanding is achieved by structuring and updating its KBs. They aim to represent both domain and general problem-solving knowledge⁴ and are structured in terms of the topics required to be understood, the goals to be achieved, etc. The structuring for each topic may be in terms of topic-primitives (these are primitives only for the specific topic referred to and the particular learner concerned) or/and the topic's goals, procedural knowledge, etc. They should be highly structured and modular. There should be, for instance, modules for rules and facts, heuristics, misconceptions, skills and competences, etc. which will be in need of continual update. Monitoring can be done on the basis of justifiable articulations (i.e. explanations) of the Domain Knowledge as seen by the learner.

The U/E subsystem of the learner model operates on the Learner's PES in order to determine the state of the learner's current understanding of a particular event (this is expressed in terms of what the learner model can output in explanation of the event). The U/E subsystem of the tutor

⁴ For an implementation see Goodfellow (1990), and McIntyre (1991).

model operates on the Tutor's PES to determine what the teacher model can output in explanation of the event. Both outputs are sent to the interface, where they are tested for equivalence. The result of the test itself constitutes an event. The process is therefore iterative, and may move through multiple levels of explanation and understanding. The objective of the process is to achieve correspondence between the output explanations of the two models. In order to do this, new data may be sought from the knowledge bases or from the user. In some cases the user may be required to arbitrate between conflicting explanations. The manner in which changes or additions are made to the databases is governed by pre-defined modes of interaction.⁵ The processing modules of these subsystems deal both with general tasks such as making plans, doing a diagnosis, etc. and also with specific tasks such as selecting tutoring methods for a particular topic, or matching a problem parameter against a historical record, etc. Each is constructed for the operation of whatever reasoning strategies (such as heuristic classification, reasoning from structure and function, etc.) are appropriate to the particular task it specialises in. The modules are invoked by the U/E subsystems to provide both specific data (i.e. entities, quantities, etc.) and types of data structure (e.g. a model of a task decomposition for diagnosis) to be used in the initialisation of the explanation/understanding matrices (see next section).

THE UNDERSTANDING/EXPLANATION SUBSYSTEMS: THE FUNCTIONAL CORE

The functioning of the explanation part of the U/E subsystems is based on three types of explanation:⁶

- (1) Trace-based (or type 1 according to Chandrasekaran *et al.*): a trace of run-time, data-dependent problem-solving behaviour;
- (2) Deep model-based (or type 2): a representation of how a piece of knowledge relates to the domain; and
- (3) Strategy-based (or type 3): a representation of how the control strategy is used.

Similarly, the functioning of the understanding part of the U/E subsystems is based on three types of understanding:⁷ concept-based, rule-based, and analogy-based. These three types of understanding have been put into correspondence with the three types of explanation described above.⁸ The correlation is represented in a matrix form, and is used to determine the respective viewpoints of learner and tutor during an interaction.

In such a matrix (Figure 2) each cell refers to a different type of domain and general problem-solving knowledge. The domain knowledge concerns objects (such as machines, components, etc.), entities (such as lists, schedules, etc.), and the different kinds of relations between them (e.g. part-of-relations, causal relations, etc.). The general problem-solving knowledge concerns the way concepts, patterns, rules, and strategies are used in finding solutions to problems. Each knowledge type is distinguished from the others by its relevance to a particular combination of explanation type and understanding

⁵ Modes of interaction are top-level protocols governing the objectives of the interaction. Example modes are: exploration (in which the objective is for the user to explore the static databases) and problem solving (in which the objective is for the user to find a solution to a particular problem). The selection determines which of the two models initiates processing and at which part of the matrix initialisation begins.

⁶ For a succinct summary see Chandrasekaran *et al.* (1987).

⁷ The rationale behind kinds, types, and levels of understanding can be found in Gelepithis (1985, 1986, 1991b).

⁸ These three types were chosen as the most appropriate for the purposes of some of the work carried out under grant D1004 (DELTA PROJECT) of the Commission of the European Communities, DGXIII, Telecommunications, Information Industries and Innovation.

level. The nine types specified are briefly described below, with examples from the domain of production scheduling.⁹

Types of explanation	Types of understanding		
	CONCEPT-BASED	RULE-BASED	ANALOGY-BASED
TRACE-BASED	Terms	Procedures	Matches
DEEP KNOWLEDGE-BASED	Structures	Meta-rules	Correspondencies
STRATEGY-BASED	States	Tasks	Contexts

Figure 2. Explanation/Understanding matrix as envisaged for the ARTISAN¹⁰ project.

Knowledge types

Terms: names of atomic objects and entities in the domain and taxonomic relations between them, e.g. an ORDER consists of a succession of JOBS each involving a number of PARTS processed on a single MACHINE.

Structures: relations of structure and function between objects and entities in the domain, e.g. ALGORITHM X causes the REORDERING of a WORKLIST so that ORDERS having the largest number of JOBS are at the beginning.

States: forms of organisation of objects, entities, relations, and structures which have significance for human domain objectives, e.g. the BOTTLENECK PROCESS in SCHEDULE X is MACHINE Y.

Procedures: specific relations of condition and consequence or action in the domain, e.g. IF a BATCH has a large number of OPERATIONS on the BOTTLENECK MACHINE THEN put it at the START of the SCHEDULE.

Meta-rules: general relations of condition and procedure in the domain, e.g. use RULES referring

⁹ This was the domain for which a research tool, along the lines of our alternative architecture, was developed (Goodfellow, 1990; McIntyre, 1991).

¹⁰ ARTISAN is the 'mascot' name adopted by the authors' consortium for grant D1004 ('Intelligent Framework for the Industrial Training Environment') of the Commission of the European Communities, DGXIII, Telecommunications, Information Industries and Innovation.

to BOTTLENECK PROCESSES FIRST, consider FACTS about OPERATORS before making INFERENCES about TIMES.

Tasks: forms of organisation of procedures and meta-rules appropriate to human domain objectives, e.g. to PLAN a SCHEDULE, GENERATE a PLAN then TEST it then ADAPT it.

Matches: values of features of objects and entities in equivalent domains, e.g. ORDER X in SCHE-DULE Y has N different PROCESSES, and SETUP A in PROCESS B has N different STEPS.

Correspondencies: structural or functional equivalencies between objects and entities in different domains, e.g. the LOAD on the CELL has the same effect on EFFICIENCY as the ELECTRICAL LOAD on a GENERATOR.

Contexts: equivalences between domains in general, e.g. making a PRODUCTION SCHEDULE is equivalent to making a MACHINING SETUP, in that both require an ORDERING effort to minimise WORK-IN-PROCESS.

Both U/E subsystems contain a representation of the explanation/understanding matrix. In response to an interface event, each cell of each matrix is initialised with knowledge slots for the event, derived from the static databases by the operation of one or more of the processing modules. Fillers for these slots are then sought from the dynamic database. The starting point (i.e. the first cell to be filled) is determined by the overall mode of interaction selected by the user. As soon as a cell is instantiated it is proposed to the interface and compared to the corresponding cell from the other model's matrix. If the fillers are equivalent then the process continues. If they are not equivalent then the slots themselves are compared. If these are not equivalent then the process reverts to a previous cell and new input is requested. Instantiation continues until the two matrices are in the maximum possible state of equivalence. The contents of each cell of the (partially or completely) filled matrices are then displayed at the interface, and the system waits for the user to initiate a new event.

Along the understanding axis, the three columns represent facts, problem-solving methods, and tacit knowledge. This is the direction in which the learner will build up understanding. Down the explanation axis, the rows represent system behaviour, first principles, and human control. This is the direction in which the teacher will build up explanation. Before any cell in the matrix can be instantiated, the knowledge for that cell of both learner and teacher must correspond. A proposal by either party to instantiate a cell with a piece of knowledge is therefore equivalent to an assumption that the piece of knowledge is held in common. A failure of the cell to be so instantiated requires a retraction of the assumption. The matrix fully instantiated is equivalent to complete correspondence in explanatory knowledge of some topic or event between learner and teacher.

SUGGESTIONS FOR FURTHER WORK

In this paper we have introduced the functional core of an alternative ITS architecture, namely, an Understanding/Explanation system. Our approach was empirical. We now need to consolidate and extend our work along several directions. We briefly introduce two of these here. We believe that the first objective should be to provide firm theoretical support for the claim that understanding and explanation processes constitute the central, shaping processes of human cognition. Some preliminary work has already been done (Gelepithis, 1991a) but the great bulk of it lies ahead. Anderson (1987, p. 475) notes that:

The key factor in delivering instruction is that at all points the tutor has a rich interpretation

of the student's inferred mental state. Exactly how one brings the tutor's cognitive interpretation to bear in instruction depends on one's learning theory.

Our contention is that learning is subservient to understanding. This is an unusual claim, but one that needs to be established if the centrality of understanding for the structuring of human cognition is to succeed. The second line of research readily suggests itself: to investigate the psychological plausibility of the proposed architecture in selected application domains.

ACKNOWLEDGEMENT

This research has been supported by grant D1004 (Delta Project) of the Commission of the European Communities, DGXIII, Telecommunications, Information Industries and Innovation.

REFERENCES

- Anderson, J. (1987). Methodologies for studying human knowledge. *Behavioral and Brain Sciences*, **10**, 467–505.
- Chadrsekaran, B., Tanner, M. and Josephson J. (1987). Explanation of the role of control strategies and deep models. Hendler, J. (Ed.), In *Expert Systems: the User Interface*, Ablex, Norwood, NJ.
- Carbonell, J. (1970). AI in CAI: An artificial-intelligence approach to computer-assisted instruction. *IEEE Trans. on Man-Machine Systems*, **MMS-11**, No. 4, 190–202.
- Clancey, W. J. (1987). Methodology for building an intelligent tutoring system. In Kearsley, G. P. (Ed.) *Artificial Intelligence and Instruction: Applications and Methods*. Addison-Wesley, Reading, MA.
- Elsom-Cook, M. (1990). Guided discovery tutoring. In Elsom-Cook, M. (Ed.), In *Guided Discovery Tutoring. A Framework for ICAI Research*, Chapman and Hall, London.
- Gelepithis, P. A. M. (1985). *The Nature of Human Understanding: Human Primitives*. RTP-2, School of Information Systems, Kingston Polytechnic, Surrey, UK.
- Gelepithis, P. A. M. (1986). Conceptions of human understanding: A critical review. *Cognitive Systems*, **1-4**, 295–305.
- Gelepithis, P. A. M. (1991a). Invariants for artificial intelligence and cognitive science. Ninth International Workshop of the European Society for the Study of Cognitive Systems, Vienna, 2–5 July.
- Gelepithis, P. A. M. (1991b). The possibility of machine intelligence and the impossibility of human-machine communication. *Cybernetica*, **XXXIV**, No. 4.
- Goodfellow, R. (1990). *Design and Functionality of a Demonstration Prototype Learning System for Workcell Scheduling (ADEM)*. Internal Report, IR-7, for the EEC-Delta Project D1104 (ARTISAN), September.
- Goodfellow, R. and Gelepithis, P. A. M. (1991). *Learner Expert System Specifications*. Internal Report, IR-9, for the EEC-Delta Project D1104 (ARTISAN), February.
- McIntyre, A. (1991). *Implementing the ARTISAN Research Tool*. KT memo N. 91-1, 18 February, Knowledge Technologies NV, Pelikaanstraat 28, 1000 Brussels, Belgium.
- Murray, W. R. (1989). Dynamic instructional planning in the BBN1 blackboard architecture. In Jagannathan, V., Dodhiawala, R. and Baum, L. S. (Eds.), *Blackboard Architectures and Applications*. Academic Press, New York.
- Ohlsson, S. (1987). Some principles of intelligent tutoring. In Lawler, R. and Yazdani, M. (Eds), *Artificial Intelligence and Education: Learning Environments and Intelligent Tutoring Systems*. Ablex, Norwood, NJ.
- Pitacco, N. and Cerri, S. A. (1989). *SNAKE: A Shell for Building Intelligent Tutoring Systems*. Unpublished report, Dida*Lab R&D Laboratory, Dida*El srl, Via Lamarmora 3, 20122 Milano, Italy.
- Self, J. A. (1987). The application of machine learning to student modelling. In Lawler, R. and Yazdani, M. (Eds), *Artificial Intelligence and Education: Learning Environments and Intelligent Tutoring Systems*. Ablex, Norwood, NJ.

- Self, J. A. (1988). Bypassing the intractable problem of student modelling, *Proceedings of Intelligent Tutoring Systems Conference*, Montreal, Canada.
- Sleeman, D. (1987). Pixie: A shell for developing intelligent tutoring systems. In Lawler, R. and Yazdani, M. (Eds), *Artificial Intelligence and Education: Learning Environments and Intelligent Tutoring Systems*. Ablex, Norwood, NJ.

Address for correspondence: Kingston Polytechnic, Kingston-upon-Thames, Surrey KT1 2EE, UK.