

**Research
Paper**

An Alternative Architecture for Intelligent Tutoring Systems: Theoretical and Implementational Aspects

PETROS A. M. GELEPITHIS and ROBIN GOODFELLOW
Kingston University, UK

ABSTRACT

In this paper some theoretical aspects of an alternative architecture for intelligent tutoring systems are highlighted, a prototype research tool in the domain of work-cell scheduling is introduced, and suggestions for future theoretical and implementational work are made. In particular: first, we briefly justify our claim that the processes of understanding and explanation are the central processes shaping the cognitive structures of learners and tutors. Second, the implications of this proposal for ITS design are made explicit. Third, we introduce the elements of a prototype representing three levels of understanding and implementing the explore mode of interaction between the learner and tutor models.

KEY WORDS Intelligent tutoring architectures Cognitive structures
Understanding Explanation Workcell scheduling

INTRODUCTION

Most 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. A slightly different approach is adopted by Clancey (1987), who chooses to identify an ITS with an ES plus a Teaching Knowledge module. It should be noted that 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; nevertheless, such an endeavour has not so far been undertaken. An alternative architecture for ITSs has been presented in Gelepithis and Goodfellow (1992). In this paper some theoretical and implementational aspects of that architecture are highlighted and suggestions for future work are made.

THEORETICAL ASPECTS

Teaching and learning (including apprenticeship) are not 'one-way processes.'¹ It is *not* the case that the teacher teaches only and the learner learns only. *Learner-tutor duality* refers to the fact

¹ This paragraph has been abstracted from the introduction in Gelepithis and Goodfellow (1992).

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 process. For example, 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. 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) aim to provide an approximate representation of the learner's (correspondingly, tutor's) cognitive state specific to the learning situation at hand. Therefore, the traditional domain expertise module should be substituted by two separate sets of knowledge bases: one each for the learner and tutor models, respectively. Similarly, the traditional pedagogic expertise module should be substituted by two separate pedagogic expertise subsystems (PESs), one each for the learner and tutor models, respectively, composed of a set of pedagogical knowledge bases.

The questions now are: (1) What processes, if any, shape the cognitive state of learners and tutors? (2) What implications for ITS design do such processes have? (3) How can such processes be implemented? Our response to the first question amounts currently, to little more than the following:

Claim: The processes of understanding and explanation are the central processes shaping the cognitive structures of learners and tutors.

This claim is based on the following points. First, explanation in the form of theories, models, frameworks, etc. constitutes the core of any 'theory' both in the hard and soft sciences. Second, understanding and explanation are necessary elements of commonsense reasoning. Third, the process of understanding, not the end result of that process, is characterized by its reducibility to primitives (Gelepithis, 1985). Such a reducibility makes understanding a shaping force of the conscious human cognitive structures and, by virtue of its structured (i.e. multiple levels of understanding corresponding to multiple levels of premises-based descriptions) and systematic nature, an indispensable compass in the design of complex, intelligent systems (Gelepithis, 1990).

Our answer to the second question is based on the widely accepted point that the processes of understanding and explanation are closely related. In ITS design terms such a close relation implies that these processes should be reflected in an Understanding/Explanation (U/E) subsystem constituting the heart of both the learner and tutor models.² This close relation, coupled with the configuring nature of understanding, strongly suggests that the U/E subsystem should be both the processing and the interfacing structure for each of the models. Consequently, on the basis of our analysis so far, each model will be composed of: (1) a set of domain and general knowledge bases; (2) a set of pedagogical knowledge bases; and (3) the U/E subsystem.

The functioning of the U/E subsystems is based on a correlation between specific types of understanding and explanation³ providing the framework in which the learner (or tutor) will build up his or her understanding and the teacher (or learner) will provide his or her explanations, respectively. Such a correlation may be represented in a matrix form, and can be used to determine the respective viewpoints of learner and tutor during an interaction. For an example of an actual matrix (correlating three types of explanation with three types of understanding) the reader is referred to Gelepithis and Goodfellow (1992). The overall function of the understanding/explanation matrix is as follows:

² We define a learner (correspondingly, tutor) model as a representation of all the cognitive states specific to the learning situation at hand for that learner (tutor).

³ Types of understanding and explanation is an interesting, open topic. For an initial dip into the latter the reader is referred to Chandrasekaran *et al.* (1987).

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 (ie: 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. . . . 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 that 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 (*ibid.* p. 34).

Constrained by space limitations we state only some of the key directions for future work: (1) enrich the processing characteristics of the U/E subsystem; (2) refine the design of the U/E matrix; (3) investigate the relation of understanding to the lower level processes of comprehension, and (4) investigate the relation of understanding to learning (our contention is that learning is subservient to understanding).

IMPLEMENTATIONAL ASPECTS

To demonstrate the feasibility of our ideas we developed a prototype in the framework of a research grant⁴ aiming, among other things, to identify generic computer-based educational tools for advanced manufacturing systems. The domain of workcell scheduling was selected as it involves higher cognitive skills (e.g. planning). Existing scheduling practice often utilizes computerized job aids which represent schedules as Gantt-chart diagrams. The interface for the prototype is based on such a job-aid (BICC's⁵ ACIT system).

Requirements for the learning process involved in workcell scheduling have been derived from a 'best-practice', human-centred model of apprenticeship learning (EEC-Delta project D1104; 1991), characterized by: (1) links between on- and off-site learning; (2) learner responsibility for learning; and (3) having its basis in purposeful task-directed activity. These requirements have been incorporated into the following scenario. Exemplary data representing good practice in the use of the scheduling aid is acquired from real-work operations by experts in the workcell. The experts add comments to the examples, relating them to principles of good scheduling practice. Learners explore the data, experimenting for themselves and relating their results to the good practice and comments of the experts. The learners then feed back what they have learned or discovered into the real-work environment.

There are thus two roles for the user of the learning system: domain expert and learner. The primary objective of the domain expert is to solve problems in the domain. That of the learner is to explore the domain. The objectives may, however, be shared (e.g. an expert may look for an optimal solution by exploration, or a learner may explore by attempting to solve problems).

⁴ EEC-Delta project D1104, 1989-1991.

⁵ Consortium partner.

The forms of representation of the domain knowledge will therefore not only reflect the differing understanding levels of learners and domain experts but also support the translation of one into the other.

Representation of domain knowledge supports three levels understanding besides the graphic representation of the Gantt chart:

- (1) *Numerical* data representing the numbers of parts to be made, the times of setup and running on the different machines, etc.;
- (2) '*Concept*' used by operators to represent and describe different entities and states in the domain, such as 'largest batch', 'bottleneck process', etc.;
- (3) *Textual* describing particular states and the experimental knowledge an expert brings to bear on them, e.g. that a particular changeover between machines has been proved to cause problems.

In relation to the subsystems of the learner and tutor models the developed prototype has: (1) a shared domain and general knowledge; (2) inferred PESs; and (3) a partially implemented U/E subsystem. The shared knowledge base is subdivided into static and dynamic areas, the former incorporating the basic numeric, concept, and text data and the latter containing the data relevant to current exploration and problem solving.⁶ The PESs are inferred implementations of aspects of the Understanding/Explanation subsystem. The system's overall strategy is to integrate two types of output: one (from the learner model) a representation of the exploration process of the learner with an inferred level of understanding, the other (from the teacher model) a representation of the problem-solving approach of the expert, with an appropriate level of explanation. The combined outputs are used to constrain the production from the static database (i.e. the numeric database above) of examples of problems and solutions in the domain which relate to both learner's and teacher's perception of the current problem-state. System behaviour is initiated by the U/E subsystem of one or the other model, according to the mode of interaction (Explore or Problem-solve) selected by the user. The initiating U/E subsystem generates an account of the current problem-state at the highest viewpoint level (numeric, concept, text) available to it, or at a level specified by the user. This account is compared with whatever can be generated by the other model, at its own highest viewpoint level, and the difference between the two accounts determines the breadth of the subsequent search for related examples in the static database.

The most important limitation of the prototype to date is that it only implements interaction between the learner and tutor models in the explore mode. Future work will extend the implementation to the problem-solve mode and to other significant modes of learner-teacher interaction such as 'introduce' and 'monitor'. Other directions for further work will attempt to extend the basic architecture to cater for a full PES, extend the implementation of the U/E matrix, and develop equivalent versions of the prototype in industrial domains such as electronic and mechanical fault diagnosis.

REFERENCES

- Chadrusekaran, B., Tanner, M. and Josephson, J. (1987). Explanation of the role of control strategies and deep models. In Hendler, J. (Ed.), *Expert Systems: the User Interface*, Ablex, Norwood, NJ.
- 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.

⁶ The manner in which modifications are made to the databases is governed by predefined modes of interaction reflecting the objectives of the user. For example, exploring the static databases (explore mode), finding a solution to a particular problem (solve mode), etc.

- EEC Delta project D1104 (1991). *Deliverable 5*. Requests for copies to Greater London-Enterprise, 63-67 Newington Causeway, London SE1 6BD. UK.
- Elsom-Cook, M. (1990). Guided discovery tutoring. In Elsom-Cook, M. (Ed.), *Guided Discovery Tutoring*, Chapman and Hall, London.
- Gelepithis, P. A. M. (1985). *The nature of human understanding: human primitives*, RTP-2, School of Information Systems, Kingston Polytechnic, UK.
- Gelepithis, P. A. M. (1990). A new design philosophy for complex, intelligent based information systems development. International Conference in Cybernetics, Birkbeck College, London, UK, 22 September.
- Gelepithis, P. A. M. and Goodfellow, R. (1992). An alternative architecture for intelligent tutoring systems. *Interactive Learning International* 8(1), 29-36.
- 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, Milano, Italy.
- 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: **Dr Petros A. M. Gelepithis**, School of Information Systems, Kingston University, Penrhyn Road, Kingston-upon-Thames, Surrey KT1 2EE, UK. Tel: +44 (81) 547 7695 Fax: +44 (81) 547 1457.