# Intelligent Tutoring Tools in a Computer-Integrated Learning Environment for Introductory Numeric Disciplines

**Ashok Patel,** *De Montfort University, UK* **and Kinshuk,** *German National Research Center for Information Technology*

*(Innovations in Education and Training International, 34(3), August 1, 1997, pp200-207, ISSN 1355-8005)*

## SUMMARY

Research in the field of Intelligent Tutoring Systems has failed to provide any substantial or viable systems that could be used in real academic environments. This situation appears to be the result of two factors: first, the failure to identify clearly the objectives and the scope of such systems; and second, the continuously shifting technological platform on which such systems are built. This paper examines the possible objectives for the development of tutoring systems and presents an approach adopted by the Byzantium project. It describes a model of computer integrated learning environment (CILE) and discusses the role of an intelligent tutoring tool (ITT) within this model. The paper also considers the potential of the Internet for various learning environments. Based on our experience of designing and implementing four ITTs that have the same look and feel (but which address diverse subject areas) the paper suggests a possible extension of the Byzantium approach to the Internet through the conversion of ITTs into intelligent tutoring applets (ITAs).

## INTRODUCTION

Technological advancements in education have received much attention in the last few years. Researchers have explored the areas such as intelligent tutoring systems, expert systems, hypertext and multimedia as a substitute for a human teacher. It appears that the much attended user modelling aspect of this research has failed to address the diversity and richness of the educational environment - even in terms of their basic properties. The range of possibilities ( such as nursery to post-graduate schools, adult education, distance education) and diverse subject disciplines are indicative of the size of the problem, let alone identifying and catering for individual differences (Kinshuk & Patel, 1996).

Traditional intelligent tutoring systems (ITS) designers have attempted to provide tutoring facilities that try to satisfy all of the student, teacher, curriculum and institutional needs. However, this has proved to be an immense task. An increased understanding of this immensity has brought about increasing recognition of two concepts. The first relates to the general acceptance that knowledge has a contextual component and that the context provides a principled way to cluster, partition and organize knowledge (Brezillon & Abu-Hakima, 1995). The second relates to the acceptance of the role of a student’s common sense and general problem-solving abilities in the learning process - some researchers arguing that human tutors virtually never provide the sort of explicit diagnosis of student misconceptions that is sought to be provided in a traditional ITS (Cumming, 1991).

It is also important to examine the suitability of diverse technologies (including computers) in an educational context. It would be useful to identify the differences between computers and other educational technologies. In a learning context, a video clip accompanied by audio on a multimedia computer is no different from seeing it on a video player, unless the computer offers some opportunity for interaction. The value of such interaction, apart from the morale boosting confirmation of a correct action, lies in the province of *learning by mistakes*. This demonstrates that the *know-how* type of knowledge is more suitable to computer based learning (Patel & Kinshuk, 1996a).

With the increasing demand for a broad based appreciation of multiple disciplines, there is an increasing pressure on academics to ensure the transfer of adequate skills to all students - some of whom may be less motivated than others. While staff resources for lecturing are relatively less affected by this increased demand, the tutorial aspect of teaching requires a growth that cannot be met due to insufficient funding. Increased reliance is placed on self-study, which poses no great problems for well-motivated students; however, it is the remaining students for whom an adequate level of knowledge transfer has to be ensured either directly through the teacher as an instructor or indirectly through the teacher as a facilitator for adequate knowledge construction. The resource intensive tutorials generally deal with the operational (*know-how*) aspect of knowledge and are suitable for software tutors as noted above. This opportunity of harnessing computers for education creates a need for interactive software tutors and provides the basis for the work reported below.

## THE DEVELOPMENT OF PROTOTYPES

A consortium of six British universities (known as the Byzantium Project) obtained substantial funding from the UK's Teaching and Learning Technology Programme (TLTP) in order to develop tutoring software for introductory accounting. The funding was obtained on the strengths of DOS-based prototype software to support the teaching and learning of Marginal Costing. The prototype was developed in order to demonstrate the utility of applied intelligence to student learning in introductory numeric disciplines and to date has been used by more than 3000 students (Wilkinson-Riddle and Patel,1992).

Unlike traditional intelligent tutoring systems (that at times attempt to outperform a human tutor and claim improved learning), the Byzantium project has a modest aim - that of producing intelligent tutoring tools (ITTs) that extend a lecturer's scope. This is achieved by horizontally partitioning some of the teaching activities e.g. supervising the development of operational skills, and assigning them to a computer tutor. It should be recognized that computers, though quite powerful, are only one of the educational technologies that can be applied in a learning environment. This recognition in itself causes a shift away from comprehensive ITSs to ITTs that can be mixed and matched with other educational technologies and human teachers in various configurations in order to create a computer integrated learning environments (CILE) to suit classroom-based, open and distance learning strategies.

## THE BYZANTIUM MODEL OF COMPUTER INTEGRATED LEARNING

One approach to computer integration within a learning environment is reflected in the Byzantium model of CILE. The aims of the approach are:

- to use computers and humans for what they are (*currently*) good at;
- to employ useful software tools within the overall learning environment consisting of human teachers and educational technologies;
- to add a level of applied artificial intelligence to the software tools in order to provide a degree of support to students thereby enabling them to work by themselves;
- to let intelligent tutoring evolve (from practically useful applications) in a bottom up fashion (through vertical and horizontal integration) rather than be designed top-down;
- to understand the economics of the learning environments and be concerned with assessment and course management as they consume substantial human resource - the system should provide economies of staff time in all areas of the teaching cycle, e.g. exposition, example setting and grading work (Wilkinson-Riddle & Patel, 1992);
- to appreciate the economics of software production and recognize that tutoring software is a joint cognitive system (Dalal and Kasper, 1994) involving a student, tutoring software and a human teacher whose involvement may be greater or lesser depending on capabilities of the software and the manner of its integration into the curriculum. In such joint cognitive systems, it is more economic at times to increase a student’s understanding of the tutoring software or let a human teacher handle certain aspects of the teaching than expend huge effort and expense in trying to design a comprehensive ITS that attempts to comprehend predispositions and mental processes of students with different personalities and backgrounds;
- to acknowledge that various educational technologies have their own strengths and that a learning environment benefits synergistically from an appropriate use of multiple resources (the converse also being true in the learning environment becoming suboptimal through inappropriate use of the educational technologies) and therefore the usefulness of any resource can only be studied in the context of systems employing it (Patel & Kinshuk, 1996c).

## THE EVOLUTION OF COMPLEXITY

It should be recognised that a usable, true and substantial ITS is, intrinsically, a highly complex system. Achieving this level of complexity by designing it top-down is almost an impossible task, that is complicated further by the continuously evolving technology. It would, therefore, be useful to view the highly complex system as being made up of smaller entities which are networked - the connectivity producing the necessary flexibility and synergy. When this smaller entity has a degree of intelligence, it is an intelligent system of a lower order. To distinguish it from the conventional understanding of an ITS, it is called an Intelligent Tutoring Tool (ITT).

It is hoped that when sufficient ITTs are ready, a second level of integrative tools will become possible (Patel & Kinshuk, 1996b). This integration can occur in two directions: vertical and horizontal. *Vertical integration* in a Ranking ITT allows holding and comparing results of different instances of an ITT, for example, comparing four different investment proposals in the presence of a ranking factor like the investor's attitude to risk. *Horizontal integration* in a Linking ITT allows use of multiple tools to solve a given problem - for example, a Marginal Costing ITT providing contribution figures for maximising profit in a Linear Programming ITT. Thus, while a simple Linking ITT is a network of Basic ITTs, it can be extended along the dimension of aggregation granularity to include other Ranking or Linking ITTs. The advantage of this approach is that the necessity of interpreting complex interactions and the construction of complex feedback messages is eliminated. A student who has failed to grasp any of the intermediate steps drops to an appropriate granularity level where the simple feedback messages are adequate.

The emergence of a higher degree of intelligence through connectivity is demonstrated by combining the ITTs with a teaching support tool (TST) like the *Marker* software which enables the testing of a student's interpretation for an examination-type narrative question and allows the system to discriminate between correct solution, incorrect interpretation but correct method, and incorrect solution.

An individual ITT is thus envisaged as an autonomous entity possessing rudimentary intelligence and then connected in a decentralised network enabling the emergence of a more sophisticated tutoring system. Even if this vision fails to materialise, the individual ITTs have their own utility and the total effort is not wasted. N order to appreciate better the strengths of an ITT, some aspects of their design are discussed next.

## USING APPLIED ARTIFICIAL INTELLIGENCE IN TUTORING TOOLS

An ITT is designed around a software engine which stores and processes knowledge rules. Its main features are (Kinshuk, 1996):

a. a *knowledge base* containing the conceptual rules and processing information. It consists of two sections that together embed common didactic knowledge and domain didactic knowledge;
b. a *student model* which records a student's progress towards a complete solution. It contains four basic items:
   - values entered by a student;
   - value status; for example, differentiating a zero value because of a blank variable, calculated zero or data not processed due to infinity condition;
   - status of the edit fields filled in by students, such as blank or intrinsically dependent (to begin with) and independent or dependent variable when filled;
   - user preferences of generative aspects of the interface; for example, in a Capital Investment Appraisal (CIA) ITT, students can choose whether the discount

factors are given by the system, filled by using a 'Table' or calculated using a formula;

c. an *expert model* which links to the knowledge rules. This derives correct outcomes and records how they are derived. In the case of a narrative question, the local expert model is based on a student's interpretation of data while the remote expert model in the *Marker* software provides correct interpretation - their combination enabling *Marker* to assign a partial score for an answer based on a correct method but an incorrect interpretation of given data, demonstrating emergence of a higher degree of overall intelligence from the connection of the rudimentary intelligence of an ITT to the *Marker* Teaching Support Tool (TST);
d. a *tutoring module* which links students to the various parts of the engine through its user interface and gives advice based on the work done so far. It allows a student to adopt a different route to the solution than the expert model;
e. a *user interface module* which provides interaction between an end-user and various parts of an ITT;
f. a *level selector* which determines the functionality that is made available to a user through the interface. The levels are:
   i. the *student level* which allows the use of the ITTs for learning without being able to create examples for others to use;
   ii. the *lecturer level* that enables instructors to create examples or templates for use by the students and to save model answers used by the *Marker*. Marking of students' assignment work is also possible at this level. An 'auto-solve' option is provided at this level that solves whatever it can as each independent variable is input to enable adjusting of inputs to obtain a desired scenario;
   iii. an *administrator level* in which the global management of marking schemes and students' data is possible. This functionality was subsequently taken out of individual ITTs and centralized in the *Marker* to enable summarized results over a number of ITTs;

g. a set of *enhanced features* which embed functionalities such as:
   i. a *random question generator* which randomly picks variables and assigns random values within specified bounds and then derives the solution by applying its knowledge rules. Thus, an ITT need not contain any data bank and all students can get individual questions but a lecturer can create a bank of questions to be used by all the students, if so desired;
   ii. a *dynamic feedback system* that generates feedback messages dynamically based on the information received from a tutoring module. The software's advice to a student is based on the work already done by that student and what the student should best do next.

In the case of an examination type question - given in a narrative form and requiring a student to interpret data, identify given values and derive a solution - the lecturer's model answer held on the *Marker* software serves as the overall (*remote*) expert model conveying the correct interpretation and the correct method while the application (*local*) expert model tests correctness of the method employed by the student. The combination of remote and local expert models enables the *Marker* to give a partial score for an answer based on a correct method but a wrong interpretation of given data. This clearly demonstrates the emergence of a higher degree of overall intelligence from the connection of the rudimentary intelligence of an ITT to the *Marker* software.

The implementation of applied artificial intelligence enables an ITT to generate random examples and assist students in solving them. The Byzantium ITT has two main modes of operation: the interactive mode and the assignment mode. In the *interactive mode*, the software does not let a student enter a wrong value and provides immediate feedback, as found in a student modelling technique known as model tracing - the difference being that a student's outcome is monitored rather than the process (rule) employed. Such dynamic feedback is considered essential for learning as it prevents (at source) any incorrect mental associations being made by a student (Patel & Kinshuk, 1996b).

In the *assignment mode* (which is meant for testing) there is no immediate feedback and the software allows any values to be entered. Feedback on their correctness is given after the work is marked. This is static feedback. As Routen (1992) observes, "There are advantages with both forms of student monitoring. Static feedback perhaps is less obtrusive ... while dynamic feedback prevents students from making gross errors and getting completely lost."

## THE STRUCTURE OF AN INTELLIGENT TUTORING TOOL

Patel (1995) describes the variables on the screen as empty containers connected in a network of inter-relationships. They represent an instance of their associated concepts. Figure 1 demonstrates a partial network of 7 out of a total of 14 variables involved in *Marginal Costing* based upon the following equations:

$R = Q * P$ ........ **R**evenue = **Q**uantity * **P**rice

$V_T = Q * V_U$ ........ **V**ariable (**t**otal) cost = **Q**uantity * **V**ariable (**u**nit) cost

$C_T = Q * C_U$ ........ **C**ontribution (**t**otal) = **Q**uantity * **C**ontribution (**u**nit)

$C_T = R - V_T$ ........ **C**ontribution (**t**otal) = **R**evenue - **V**ariable (**t**otal) cost

Any value can be entered in any variable, provided the whole network remains consistent. The intermediate variables need not be filled in. This approach allows mental calculations and does not force a rigid path to the solution. Provided the value being entered is correct, the ITT assumes the intermediate steps to be well within a student's conceptual knowledge boundaries. If the value is incorrect, then the ITT guides the student in a graded manner. This advice is not based upon some hard-coded set of instructions. A student model is applied here to improve the student learning.

According to the terminology used by Self (1988), corrective, elaborative and evaluative aspects of the student model are used. The advice given by the software to the student is based upon the extent to which that student has been able to solve the problem so far and, in view of that, what that student should best do next. The corrective behaviour of the software tells the students whenever they make mistakes in calculations. If they make a mistake second time, elaborative behaviour comes into action, which suggests the various relationships by which certain values can be calculated.

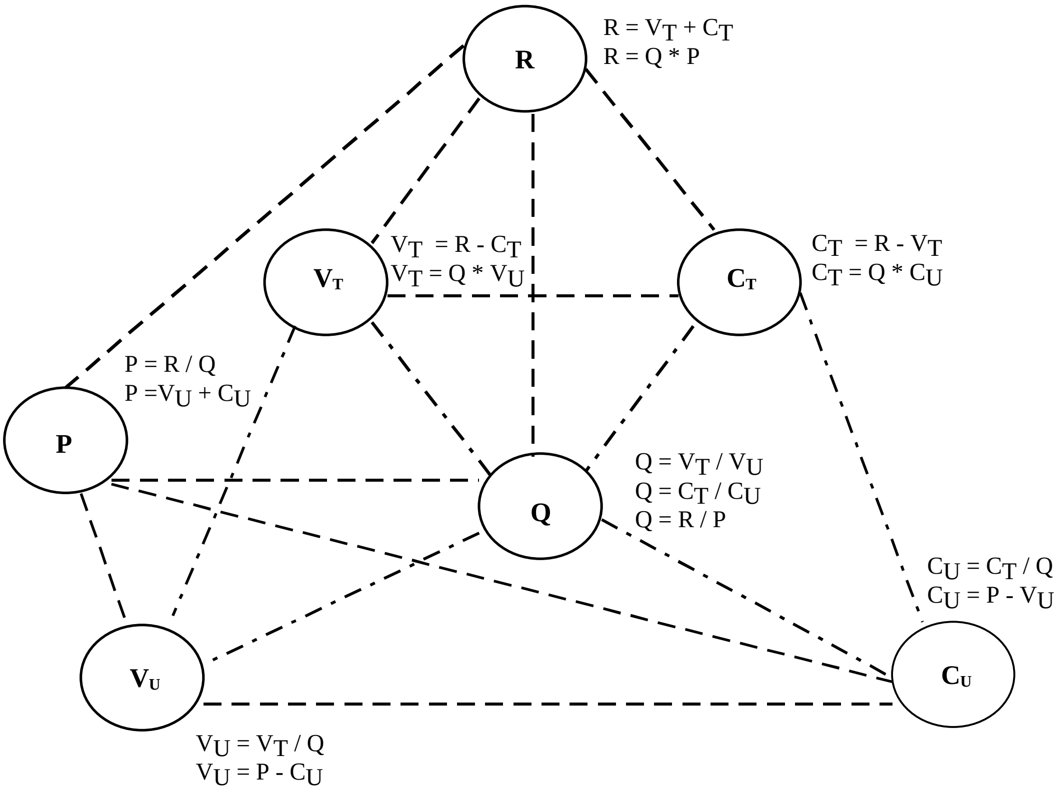


*Figure 1: A subset of the network of inter-relationships for Marginal Costing*

The suggestion depends on what the student has done so far. If enough information is not available on the screen to calculate a certain value, the software guides the student to fill in other values before attempting to complete that particular value. The evaluative aspect of software is featured in *assignment mode*, which is designed to assess the gain in a student's knowledge.

Diagnostic, predictive and strategic aspects of the student model are not considered to any significant degree in the software, the reason being that the learning topic has been 'sliced' into such small modules that they become the most basic units of that topic. A 'Road to London' paradigm is adopted - which is more concerned with 'what to do next' rather than 'how did I come here' (Patel, 1995). Coupled with dynamic feedback (which prevents any mistakes in the first place), this paradigm keeps on reinforcing the correct connections throughout the network of inter-relationships and eliminates the need for the above aspects.

## INTERFACE DESIGN

The interface of the Byzantium ITT includes various features, such as:

- a context-sensitive help link to textual description of topics;
- file and printing management; and

- efficiency/learning support tool - which may be either general, such as an on-screen calculator, or application specific, such as the discount factor table in the CIA ITT.

The user-interface is mouse-driven and employs only push-buttons, scrollbars and edit controls - so as to make the interaction as simple and instinctive as possible. A push-button is also activated by a keystroke of the letter highlighted on its label. This allows for mixing the mouse and keyboard events to enhance a user's efficiency and comfort. All data entry is done through an edit control, suitably masked to prevent entry of illegal data. The interface also reserves a certain area on the screen for interactive feedback, so users do not have to search the whole screen for the information that they require. As students prefer greater use of diagrams and pictures (Hazari & Reaves, 1994), graphical representation of data is provided wherever possible and the on-screen tutorial text is augmented with illustrations and replicas of screens. A screen-shot from the Capital Investment Appraisal (CIA) module is presented in figure 2.

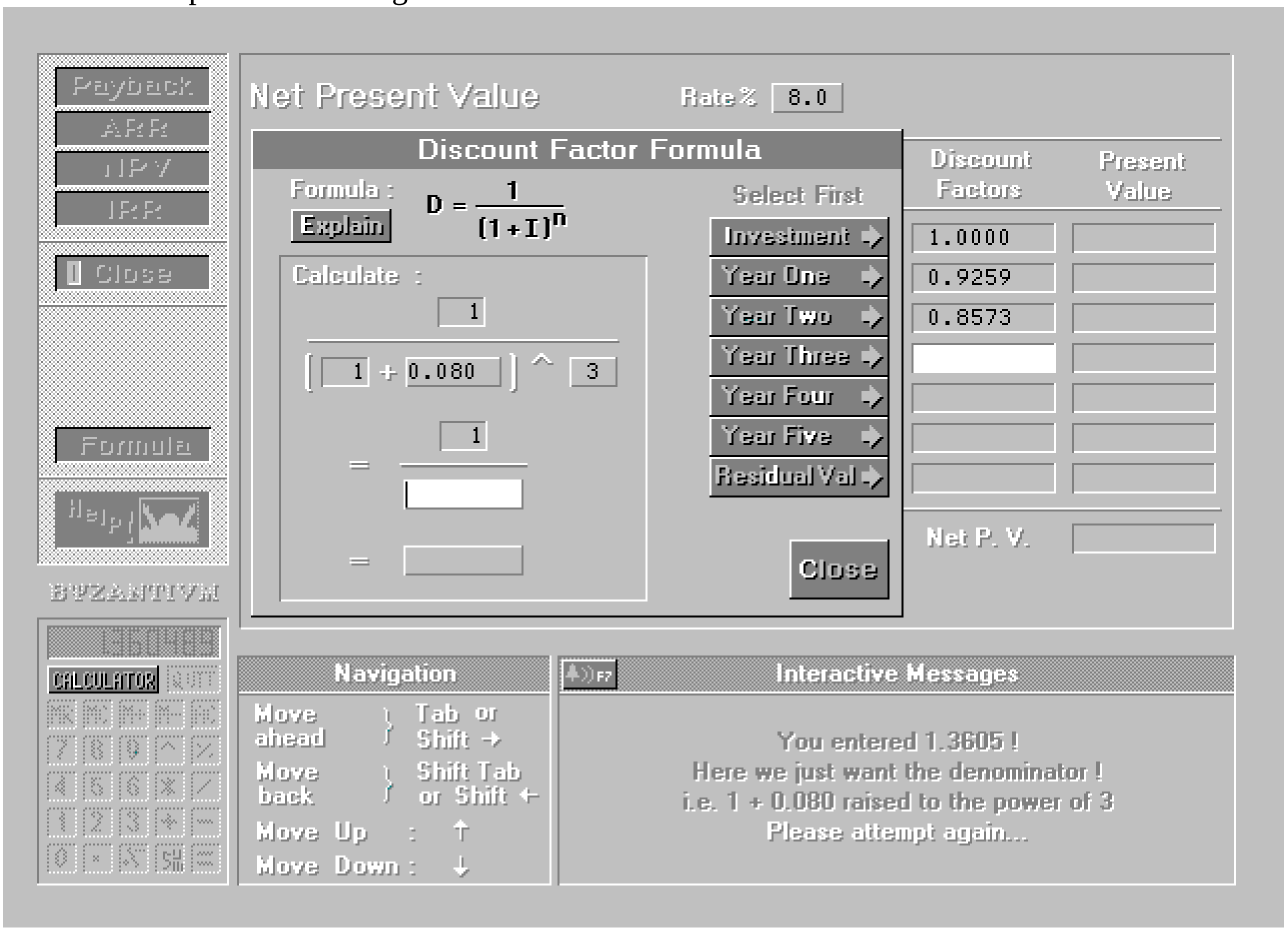


*Figure 2. A screen-shot from Capital Investment Appraisal module*

## FUNCTIONAL CONSIDERATIONS

The Byzantium ITT is designed with the consideration of high visibility, logical decomposition and visible dependencies. It does not provide ready solutions and waits for a student to execute an action before providing a hint. The ITT accepts a correct entry, regardless of the state of student outcomes model, allowing a student to carry out some operations mentally. It forces an intermediate step calculation only in case of an invalid entry - signifying that a student has not fully grasped the chain of relationships.

The ITT also provides support for decomposing complexity, that is, a student who is not confident about using a formula can call up a formula interface in the *Capital Investment Appraisal* module, which has the formula decomposed into its various constituents. A student can obtain expert help in filling each of the constituents and can work in stages to derive a result of that formula - in the context of the data to which the formula is to be applied.

In the first level of integration, the ITTs are linked to Teaching Support Tools (TSTs) like the *Marker* mainly through the data driven programming paradigm, as each ITT, through its file structure, informs the *Marker* about its fields.

## EXTENSION OF THE BYZANTIUM APPROACH TO THE INTERNET

The development of the four diverse ITTs provided necessary information to formulate the methodology to construct a general purpose tutoring tool builder. The builder, and the interactive dialogue with a subject teacher, will produce the ITT for any numeric discipline (Kinshuk & Patel, 1996c). The ITT approach can be implemented on the Internet through the creation of intelligent tutoring applets (ITAs). The Internet allows these knowledge entities, in the form of ITAs, to be held and accessed in a structured manner.

With the appropriate structuring parameters, the ITAs created by different teachers build up to a large inventory of accessible knowledge that can be utilized by all the teachers in various configurations of single or multiple ITAs to suit more advanced applications. The students will need to download the general purpose ITA modules only once, as they could be used by any ITA. In order to run each ITA, the students would need to download the appropriate ITA configuration file, created by a teacher and made available on the Internet. A possible structure of an ITA is shown in figure 3.

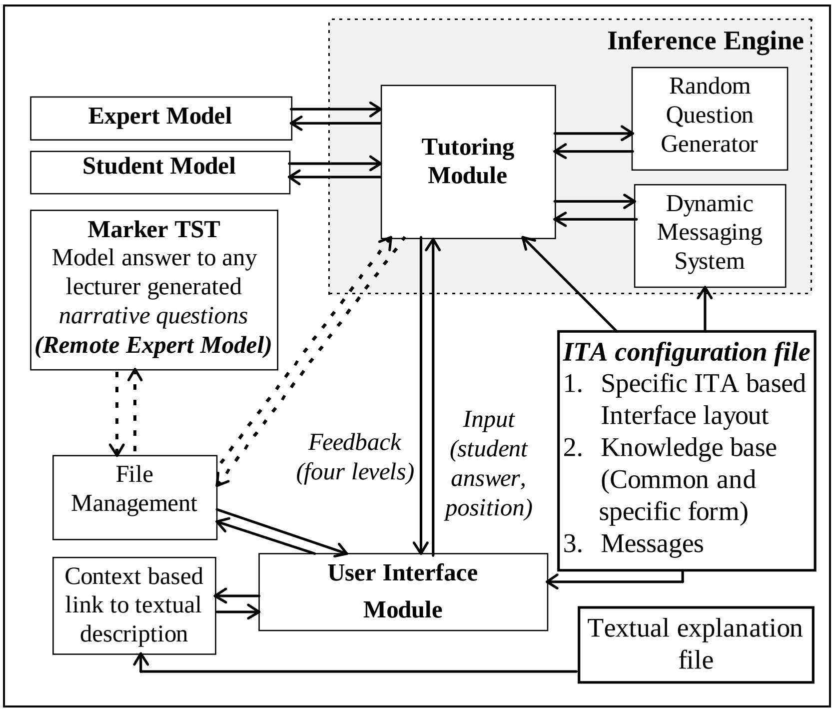


*Figure 3.Structure of an intelligent tutoring applet*

The ITAs will enable the students to use these tutoring tools interactively using standard web browsers while obtaining the same feedback and guidance as provided by an ITT (Patel & Kinshuk, 1996b). The indexing of the ITAs would require an approach that is concept-based (Brusilovsky et al., 1996) where the ITAs are classified according to all the subject domains where they may find an application and may fall in more than one classification due to the overlapping of the concepts in various disciplines. The possibility of linking various ITAs in a configurable teacher designed ITS allows the extension of the ITT concept to more complex applications. However, the methodology could only be firmly established after more prototypes are built and tested on the Internet.

## CONCLUSION

The Byzantium model of CILE does not attempt to change the process of education in any radical way. It recognizes the strengths of a human teacher and is concerned with extending a teacher's scope by providing intelligent tutoring tools that will efficiently teach introductory material and remove some of the burdens associated with assessment. This approach ensures better and more qualitative utilization of the human teacher's time and effort. It is realistic in its demand on the computer software and therefore more likely to succeed in integrating computers into the curriculum. Nevertheless, it still maintains a vision of developing a much more complex tutoring system out of small and useful ITTs.

The Internet offers a great opportunity for collaboration among teachers and students. It offers the possibility of building up a large inventory of useful knowledge entities and enables division of labour and sharing of expertise. It also offers facilities like newsgroups which may be utilized as a discussion forum by students with or without a teachers' involvement - thereby providing an opportunity for peer-to-peer learning and social interaction. Accompanied by facilities like e-mail and file transfer, it provides a valuable platform for distance learning. With the emphasis that there has been on hypertext and hypermedia in both tutoring systems and Internet applications, an important aspect of learning in numeric disciplines (namely, learning through problem solving) has not been adequately addressed. We believe that the ITTs as tutoring system, and even more so the ITAs, will fill this gap.

## REFERENCES


**Brezillon P. & Abu-Hakima S.** (1995). Using knowledge in its context: Report on the IJCAI-93 workshop. *AI Magazine*, Spring, 87-91.

**Brusilovsky P., Schwarz E. & Weber G.** (1996). A Tool for developing hypermedia-based ITS on WWW. ITS'96 Workshop on Architectures and Methods for Designing Cost-Effective and Reusable ITSs, Montreal, June 10th 1996.

**Cumming G.** (1991). Using artificial intelligence to achieve natural learning. In Lewis R. and Otsuki S. (eds.) *Advance Research on Computers in Education*, Elsevier Science Ltd, Amsterdam.

**Dalal N. P. & Kasper G. M.** (1994). The design of joint cognitive systems: the effect of cognitive coupling on performance. *International Journal of Human-Computer Studies,* 40, 677-702.

**Hazari S. I. & Reaves R. R.** (1994). Students preferences toward microcomputer user interfaces. *Computers & Education*, 22(3), 225-229.

**Kinshuk** (1996). Computer aided learning for entry level Accountancy students. PhD Thesis, De Montfort University, England, July 1996.

**Kinshuk & Patel A.** (1996). Intelligent Tutoring Tools - Redesigning ITSs for adequate knowledge transfer emphasis. In Lucas C. (ed.) *Proceedings of 1996 International Conference on Intelligent and Cognitive Systems*, Institute for Studies in Theoretical Physics and Mathematics (IPM), Tehran, 221-226.

**Patel A.** (1995). Intelligent Tutoring Tools: An evaluation of TLTP Byzantium's approach to software design for a computer integrated learning environment in introductory accounting, Project report, De Montfort University, UK, September 1995.

**Patel A. & Kinshuk** (1996a). Knowledge characteristics: Reconsidering the design of Intelligent Tutoring Systems. In Behrooz A. (ed.) *Proceedings of Knowledge Transfer - KT'96*, University of London, London.

**Patel A. & Kinshuk** (1996b). Applied Artificial Intelligence for Teaching Numeric Topics in Engineering Disciplines. In Diaz de Ilarraza Sanchez A. & Fernandez de Castro I. (eds.) *Lecture Notes in Computer Science*, 1108, Springer, Amsterdam, 132-140.

**Patel A. & Kinshuk** (1996c). Artificial Intelligence in learning and assessment - A supportive software for the teaching of Accounting. In Overmyer S. & Gornostaev J. (eds.) *Proceedings of the 6th East-West International Conference on Human-Computer Interaction,* ICSTI, Moscow, 1-11.

**Routen T.** (1992). Reusing formalisations of legislation in a tutoring system. *Artificial Intelligence Review*, 6, 145-159.

**Self J.** (1988). Student models: what use are they? In Ercoli P. & Lewis R. (eds.) *Artificial Intelligence Tools in Education,* North Holland, Amsterdam, 73-86.

**Wilkinson-Riddle G. J. & Patel A.** (1992). Applied research in the development of a software methodology enabling the use of unstructured data in management accounting CAL systems, CTI-AFM annual conference, Bradford , March 1992.

## BIOGRAPHICAL NOTE

**Ashok Patel** is the Software Director of the Byzantium Project, a consortium of six universities that have been funded under the Teaching and Learning Technology Programme of the UK's Higher Education Funding Councils.

**Kinshuk** was a research assistant in the Byzantium project and is currently a Post-Doctoral Research Fellow at the German National Research Center for Information Technology.

Address for correspondence: Dr Patel can be contacted at the CAL Research and Software Engineering Centre, 8.1-8.3, Bosworth House, De Montfort University, Leicester LE1 9BH England. Tel / fax: (44) 116 257 7193, e-mail: apatel@dmu.ac.uk

Dr Kinshuk's current address is: GMD FIT - German National Research Center for Information Technology, Schloss Birlinghoven, D-53754 St. Augustin, Near Bonn, Germany. Tel: (49) 2241 14 2144 Fax: (49) 2241 14 2065, e-mail: kinshuk@gmd.de