Intelligent Teaching Assistant Systems

Kalina Yacef
School of Information Technologies, University of Sydney
kalina@it.usyd.edu.au

Abstract
Traditionally, Intelligent Tutoring Systems (ITS) are dedicated to learners. They help them learn at their own pace, following a curriculum tailored to their individual needs and receiving individualised feedback. Intelligent Teaching Assistant Systems (ITAs) are dedicated both to learners and teachers. Their aim is to facilitate the whole teaching/learning process by helping the teacher as well as the student. There has been an increasing interest to integrate the teacher as an end-user of the ITS. This paper presents the characteristics and discusses the architecture of ITAs. We illustrate the concept of ITA with two systems that we designed.

1. Introduction

Intelligent Tutoring Systems (ITS) have been more often dedicated to learners than to teachers. The teacher is generally perceived as the administrative manager of the tutoring system (authoring systems) and sometime is one of the designers of the system. But it is not very common to read details about the triangular pedagogical relationship between the student, the teacher and the system, or about the teacher’s features in an ITS [4,5,13]. However, there has been, over the past decade, an increasing interest in the teacher’s role and his/her integration as a target user of the ITS [4,8,11,12,13].

The original aim of ITS is to help learners learn. An additional way to help learners is to help the teachers or instructors to teach better and more efficiently. This is particularly needed when teachers are a scarce resource. The pedagogical expertise and the face-to-face interactions they provide are extremely valuable. In domains where complexity, risk and lack of automation are present, they may even be irreplaceable[7]. Some Intelligent Tutoring Systems include the teacher as a final user of the system and we call them Intelligent Teaching Assistant systems (ITAs). Under this broad type of ITS, we can find more detailed names such as pedagogical assistant[9] or cooperation and collaborative learning [2,9].

ITAs are dedicated both to learners and teachers. Their purpose is to support the educational or training process in an intelligent way by assisting the teacher in his/her tasks as well as helping learners to learn. They can take a significant load off the teachers, assist them in tedious or complex tasks, keep track of the students’ results, report problems whilst helping learners to practice at their own pace in an adapted environment, receiving feedback and tailored exercises.

With an ITA, the teacher remains present in the learning process. Assisting the whole learning process rather than replacing the teacher and treating him/her as a target user is the key philosophy of an ITA. Teachers remain in control of the teaching and are seconded with an ITA.

This paper gives an overview of ITAs. Section 2 presents the concept of Intelligent Teaching Assistance. Section 3 presents the general architecture of an ITA. Section 4 discusses the new triangular didactic architecture that ensues from their use. Section 5 then briefly shows two examples of ITAs, and then comes the conclusion.

2. What type of assistance can an ITA provide to the teacher?

If we classify intelligent educational systems according to their degree of autonomy towards the teacher, we obtain the following three classes: (1) Those that are fully autonomous and replace the teacher; (2) Those that assist the teacher, whilst remaining autonomous in their use: the student can practice without the teacher being present; (3) Those that assist the teacher and are not autonomous: the teacher must remain present during their use.

Whereas the early literature on ITS focused on systems of type (1), we can now distinguish the two other types, which we call Intelligent Teaching Assistant Systems. We will give an example of each type of ITA in the last section of this paper.

We have found there are several areas in which an ITA can provide assistance, especially in the areas of pedagogy, monitoring, analysis and synthesis of individual and group results.

Help in diagnosis and assessment of learning: an ITA can help making a more precise, systemic and/or faster diagnosis of the learning. This is the case of PepiDiag, the cognitive diagnosis tool in Pepite, an assistant system for mathematical skill diagnosis [4].

Help for generating tailored material for a particular student: the ITA can assist the creation or adaptation of exercises/questions, taking into account the learning stage and difficulties of the learner. This is particular relevant in open-ended and/or highly complex domains where
assessment of the learning and follow-up exercises are too complex. The ATC-TutorHelp focuses on this feature [13].

Help for monitoring one student during the execution of an exercise: an ITA can provide careful monitoring of the execution of an exercise and free the teacher of this tedious task whilst reporting the results in a meaningful way to him/her. This is a feature of the ATC-TutorHelp.

Help for analysing or synthesizing results: the ITA can collect data from the students’ interactions with the system and report them to the teacher in various forms to increase the visibility on students’ learning. PepiProfil, in Pepite, aims at synthesizing the diagnosis results for each student so that teachers can analyse them [4]. In the Logic-ITA, teachers can visualize students results [10] from both an individual and a group point of view.

Help in creating/defining the ITS: authoring tools and semi-authoring tools. [10,11].

In addition to these types of assistance, there are two broad dimensions that an ITA may target:

Reducing the quantity or the length of burdensome tasks that can be automated or facilitated. If the teacher’s workload is reduced, s/he is more available for tasks where his/her invaluable human expertise is in demand.

Improving the quality of the teaching process, by providing new or better tools and feedback to the teacher. If s/he has better insight on students’ learning and more means to provide better teaching material, the teacher can deliver better and more focused teaching.

3. Architecture of an ITA

ITA have two different target users: the teacher and the learner. As a consequence, its architecture is slightly different. An ITA includes the same modules as an ITS: an expert module, a pedagogical module, a student module and a student interface. In addition, it has a teacher interface and a teacher module, which comprises of pedagogical assistance, monitoring and analysis/synthesis modules. These are shown below.

![Figure 1 – Architecture of an ITA](image)

Figure 1 – Architecture of an ITA

Obviously, the teacher’s module and interface are the most substantial differences with learner-centred ITSs.

The Teacher’s module varies depending on the tasks that the ITA targets. However we have identified three main components, in addition to the teacher interface.

The Pedagogy component, already present in the classical ITS architecture, is opened up to allow the teacher’s input. It is enriched with modules that contribute to the provision of pedagogical material: authoring elements, which define the overall ITS parameters, generation of tailored material (exercises, questions, feedback, explanations) that is delivered to a particular student. The student diagnosis can also be the source of assistance to the teacher.

The Monitoring component contains the knowledge and procedures needed to monitor thoroughly a student or a group of students during their session with the ITS. The Monitoring component can be used to record and provide summaries to the teacher, alert him/her if some specific incidents occur. It really is useful if the teacher’s presence is normally required during students’ practice. For example, in the ATC-TutorHelp, the system monitors that students do not breach any air traffic control rules and regulations, that tasks are attended to in a specific time and so on. It also checks that the exercise stays within the initial training objectives. If something goes wrong or if the exercise becomes too difficult, then the system will alert the teacher.

The Analysis/synthesis component includes the knowledge and procedures for analysis (identification of trends and problem areas, both at an individual and a group level) and synthesis (compilation and visualization of results, both at an individual and group level, under various criteria and presentation types).

Each of these modules may increase the complexity of the interface, depending on how active the role of the teacher is: relatively passive (visualize results), moderately active (accept or not the suggestions of next training objectives and exercises made by the system), or very active (e.g. modify the student’s diagnosis data, modify the training objectives). The teacher’s interface is drawn across both the ITS and teacher’s module because the teacher may be able to modify data in the ITS (e.g. in the student model).

The ITS module contains the traditional components generally accepted in the ITS literature, with some variations introduced by the presence of the teacher.

4. General didactic architecture with an ITA

Figure 2 illustrates the new triangular didactic architecture introduced by an ITA. The teacher still retains his/her existing interactions with the students. These can be of any sort and in both directions: classroom teaching, demonstrations, monitoring, one-to-one sessions, questions and so on. They are labelled ‘1’. Students may use an ITS to help them learn and practice. Here the whole range of Computer-Assisted Learning tools applies and obviously the deepness of the interactions depends on the tool and its computer intelligence. It should be noted that the “ITS” component may or not be autonomous, i.e. the teacher’s...
presence may or not be required. These interactions are labelled “2”. Students also interacts between each other, as labelled “3”, sometimes aided by the ITS [3,6].

The ITA provides an additional level of interaction, circled with a grey shadow on the figure. The “Teacher’s side” of the ITA oversees the ITS, helps defining the curriculum and other teaching parameters, monitors the use of the ITS by the students and reports results to the teacher. It enriches the didactic process as all the information resulting from the student’s use of the ITS are provided to the teacher, who in turn can teach better.

Figure 2. Triangular didactic architecture
Naturally, each interaction affects the others. For example, the data that the teacher obtains from the ITA (4) influences his/her future interactions with the students. Let us take the example of a recurrent mistake made by the students. The teacher will certainly address the underlying misconception in his next lectures or briefing sessions with the students.

5. Two examples of ITAs: the ATC-TutorHelp and the Logic-ITA

We illustrate here the concept of ITA by briefly presenting two systems that we designed. The ATC-TutorHelp [13], an ITA of class 3 and the Logic-ITA, an ITA of class 2 [10].

5.1. ATC-TutorHelp

Overview of the domain
The domain of this ITA is Air Traffic Control Training. Air Traffic Control (ATC) a highly complex domain: extremely dynamic, with a constant and very high risk, a large number of inter-connected variables (pilots, controllers, weather, computer systems, and so on) and a constant uncertainty due to the slight inaccuracy of measure instruments. All these factors make the full automation of ATC, hence its training process, impossible. Here more than ever, cooperation between both humans and computer systems is indispensable [7].

ATC-TutorHelp targets the final stage of ATC training. Here, students practice on simulation exercises (the equipment is the same as in real context, only the input data is simulated), where they learn to develop their operational skills, i.e. the ability to apply the knowledge and the rules learned previously. All of this is done under the close monitoring of an instructor (we therefore replace “teacher” by “instructor” in this section), whose duties are very time-consuming. S/he needs to monitor cautiously that the student does not breach any ATC rule but also that the exercise remains aligned with the original pedagogical objectives (exercises may deviate extensively depending on the student’s actions. For example an easy, low traffic exercise could become too difficult if the student initiates many poor actions). Exercises may last several hours and involve briefing and debriefing sessions.

The preparation of tailored exercises is unusually long (several hours or even days for one exercise created from scratch). This is due to the fact that the instructor reasons in terms of pedagogic situations whereas the tools s/he can use require entering basic physical data such as plane details, weather conditions and so on. It requires an important mental exercise and several trials to calculate what these physical data should be to generate eventually the desired traffic situations.

Overview of the system
The ATC-TutorHelp targets mainly these two areas, which were the two most time-consuming for the instructors: the monitoring of exercises and the conception of tailored exercises. A secondary aim is to allow the instructor to visualize the student’s results under various angles. The ATC-TutorHelp is placed above the high-fidelity simulator and performs the following tasks:

- it generates training objectives that are adapted to the individual student, according to MOST, a model for operational skill training which uses performance measures in context, the latter being composed of traffic situations and workload [14]. The instructor can delete, modify, or add new ones. Once these objectives are selected, the ATC-TutorHelp retrieves one or several exercises matching these objectives in the exercise database.

- it monitors the running of the exercise. It records performance measures for evaluating the student. If a problem arises, it stops the exercise, provides a message to the student and alerts the instructor. The ATC-TutorHelp also shows a summary of the performance measures of the current exercise to the instructor. This way, the instructor may supervise several students at a time and the data that is recorded in the student model, which will be used for the generation of next training objectives.

Architecture of the system
We will discuss the architecture in terms of the elements described earlier.

In the Teacher’s module, the Pedagogical component is in charge of identifying a list of training objectives for the next exercise session. These training objectives are expressed in terms of topics and contextual elements (type of traffic situations and workload), and they are determined by the student model data, which records performance...
measures with these contextual elements, and MOST, a Model for Operational Skill Training. For each skill, MOST identifies two major stages, after the beginning level: (a) the construction of situational patterns, where the student learns to practice the skill in various situations at a relatively low workload and (b) the development of automaticity, where the student practices this skill under an increasing workload and, inherently, in parallel with a large number of other tasks. The instructor may choose, accept, change the training objectives suggested by the system. The next step is to select an exercise that corresponds the closest to the training objectives.

The Monitoring component observes the student’s actions and checks that rules are respected and that tasks are performed in a timely manner. It also checks that the exercise does not become too difficult for the student. It provides the instructor with a summary showing the performance measures for each of the training objectives for the current exercise. It alerts the instructor if an incident occurs and stops or not the running of the exercise.

The Analysis/synthesis component displays chart showing the performance measures according to selected contextual elements. The system does not infer anything but displays the information in a way that is meaningful to the instructor. For example, the chart showing the performance measures against types of situations may reveal specific problems such as calculation problems or wrong usage of the special ATC calculation tools, or misjudgment of the vertical dimension for example. The chart of performance measures against workload shows the impact of increasing workload on the student’s performance. If this performance does not drop dramatically when the workload is high, it can be deduced that automation for the current skill is taking place.

In the ITS module, the Expert module is a rule-based system. It is able to generate a good “solution” to the exercise. However it does not generate all the possible solutions: firstly there can be a very large number of them, with the space and time factors, and secondly it is not really important. Controller trainee may develop different strategies, different reasoning, as long as they ensure safety, efficiency and expeditiousness of the air traffic.

The rest of the Pedagogical component comprises of the curriculum, which is a network of “topic” nodes with a pre-requisite structure. Each node comprises of a topic-task, a set of situations in which to practice this task and a threshold of workload.

The Student module is an overlay on the curriculum structure, except that each node also contains the current level of skill acquisition for the task, the history of performance measures with their respective contextual information, and a list of context-problems.

The ATC-TutorHelp does not allow the students to practice alone, but relieves the instructor of several very time consuming tasks and provides advice to the instructor for follow-up exercises. As it was only built as a prototype on a narrow portion of the overall ATC activity, there is no quantitative evaluation available. However, the feedback that we received from ATC instructors on this concept was very positive: they highly value the fact that the relationship instructor-student is preserved and that the control of the teaching remains theirs. Beyond the reduction of their workload and a more adequate tool to teach with, they also see the benefit of the capitalisation of exercises and a more systemic way to teach and evaluate students.

5.2. The Logic-ITA

Overview of the domain

The problem targeted here is quite different: alleviate the disproportional ratio between teachers and students in large classes, leading to lack of individualised learning and to the students’ general feeling of being lost in the mass. The domain chosen is formal proofs in propositional logic.

Overview of the system

The Logic-ITA really acts like a human teaching assistant. It has expertise of the domain and has a good pedagogical expertise that allows it to provide intelligent feedback and suggest exercises adapted to the students’ level, assess each part of the exercise and compile the results for the teacher.

The domain being suitable for an autonomous ITS, we started building a practice tool (the Logic Tutor [1]). Then we added the student model and the teacher’s components to make it an ITS. So the Logic-ITA now consists in an autonomous ITS and two tools for the teacher’s use: the LT-Configurator allows him/her to set up the parameters of the ITS and the LT-Analyser collects and displays the students’ results at a group level as well as an individual level [10]. It allows the students to practice alone but the teacher is still monitoring their results and problems.

Architecture of the system

In the Teacher’s module, the Pedagogical component is the LT-Configurator, which manages all the teaching configuration settings and material. It contains a Curriculum Sequencing Authoring Tool and a Management tool of Exercises. The curriculum sequencing in the Logic-ITA is semi-dynamic. The teacher defines students’ learning levels and sketches the high-level sequencing of the progression through the levels. The finer grain of training objectives will be decided on the fly with the data stored in the student model. In regards to the management of exercises, the teacher can create, add, and modify exercises, as well as adding comments and hints to them. S/he may choose to make partial solutions visible to the students. A basic exercise generator is also available to generate new exercises.

The Monitoring component does not exist as such since the Logic Tutor is an autonomous system.
The Analysis/synthesis component is the LT-Analyser, used by the teacher to investigate the class’s results, levels and problems. Results are stored in a database connected to MS Excel. The teacher can then use the features of the software to query the database and visualise graphics. For example the teacher may look at the performance results of the class on all exercises involving a particular rule, or the distribution of levels among a given group of students, and so on.

In the ITS module, the Expert module contains the expertise of the system in propositional logic. It checks the correctness of students’ answers dynamically and provides proper feedback when errors occur, thanks to a database of common mistakes, where each mistake is stored with its feedback and hints.

The Student module is a mix of error and overlay models. It records all the attempted exercises along with the mistakes, and stores the student’s level, performance in each exercise, results for each rule used and general student information.

The ITS part of the Pedagogical component contains the high-level curriculum sequencing, as parameterised in the LT-Configurator. It also contains the mechanism for deciding dynamically of the next training objectives using the data stored in the student model and retrieving appropriate exercises. The student is able to agree with the selection of exercises or may choose another exercise or may create a new one, thus leaving him/her in control of the learning. During the exercise, when a mistake is detected, it re-explains the correct usage of the rule (contextualised with the current formulae), and provides hints when they are available. Lastly, it contains HTML help pages, which can be consulted at any time.

The Logic-ITA helps the teachers by replacing a large part of their expertise during one-to-one exercise sessions, by reducing the marking time (formal proofs are quite tedious for a human to check), by displaying the students’ progress and difficulties in a concise or detailed way. The teacher can pinpoint individual problems as well as identify general trends. The content of the lectures can be adjusted to the class’ current state of learning. Teachers can address common students’ problems and can also identify in the mass the students who need special attention. It has been used in 2001 in a class of 442 students and evaluation showed that students obtained higher results in homework and exam questions on the Logic topic.

6. Conclusion

We have presented the concept of Intelligent Teaching Assistant systems, of which the key philosophy is to assist the teacher rather than replace him/her and treat him/her as a target user. Whilst there is no obvious limitation on the type of domains these could be used, the full benefit of ITAs comes up in domains where the human teacher is a scarce or distant resource. The link between intelligent teaching assistants and learner-centered intelligent tutoring systems is worth exploring to identify the benefits of each approach. We think that they both have a very distinct role to play. Perhaps the debate about the choice of building an autonomous ITS or an ITA could help as a starting point for elaborating a design methodology for such systems. If the requirements, benefits, limitations, and conditions of use for each type of system are formalised, the design of new systems can be simplified.

References