Towards an Intelligent Knowledge Based Tutoring System for Foreign Language Learning

Marios C. Angelides and Isabel Garcia
Information Systems Department, London School of Economics and Political Science

Intelligent Tutoring Systems have found a wide range of potential applications across a multitude of disciplines and subject areas. The Intelligent Spaniard is an Intelligent Knowledge Based Tutoring System designed to assist intermediate level students with their learning of Spanish grammar, by testing their knowledge of regular and irregular verbs. This paper describes the system's architecture which takes the form of three rule-based expert systems working in synergy with each other. The Intelligent Spaniard has been developed as a domain-independent generic tutoring system, hence, its knowledge bases can be replaced with knowledge bases containing the vocabulary and grammar of other foreign languages.

Introduction

A Tutoring System classified as Intelligent must pass three tests of intelligence [Burns and Capps, 1988]. First, the system must know the subject matter well enough to be able to draw inferences or solve problems in the domain of application. Second, it must be able to deduce a user-learner's approximation of the domain knowledge. Third, the tutorial strategy must allow the system to implement strategies that reduce the difference between the expert and the student performance.

Therefore, at the foundation of an Intelligent Tutoring System one expects to find three special kinds of knowledge: domain knowledge, student knowledge, and tutoring knowledge. A major consideration with Intelligent Tutoring Systems is how these three kinds of knowledge are embedded in the system architecture. Consequently, the anatomy of an Intelligent Tutoring System is as is shown in Figure 1.

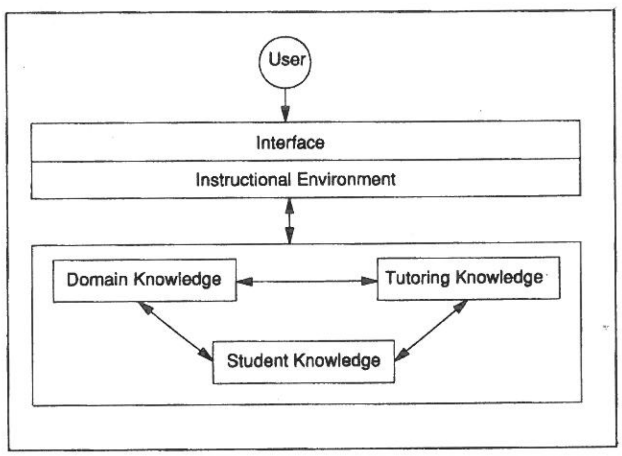

Figure 1: The Architecture of an Intelligent Tutoring System

Intelligent Tutoring Systems have been developed over the years in many areas [Angelides and Doukidis, 1990] such as mathematics, geography, electronics, medicine, foreign language learning, especially grammar, programming languages, chemistry, etc. Intelligent Tutoring Systems have been used in foreign language instruction for some time with varying degrees of success [Swartz and Yazdani, 1992]. Intelligent Tutoring Systems that attempt to model the user-learner allow for more adaptive, individualised instruction and can deliver goal-based and remedial instruction to lead the learner through the material.

Knowledge Based Expert Systems have been substantially used to develop Intelligent Tutoring Systems [Doukidis et al, 1988]. They have proved to be useful [Sleeman and Ward, 1988] in representing domain-specific expertise to be consulted by a student, in choosing an appropriate problem for the student to solve, in diagnosing missing conceptions and misconceptions in the actions of a student engaged in problem-solving and in explaining the reasoning behind advice given or misconceptions diagnosed.

The purpose of this paper is to present the Intelligent Spaniard, an Intelligent Knowledge Based Tutoring System designed to assist students whose Spanish vocabulary is of intermediate level, but who may have difficulties with Spanish grammar. The assumption underlying the use of the system is that the user possesses the required vocab-ulary and as such, the age of the student is of no relevance. The system aims to assist students with their learning of Spanish grammar, by testing their knowledge of regular and irregular verbs. Help is provided in the form of explanations on the formation of tenses, examples on the use of tenses and hints on possible reasons for any mistakes made. The users will hopefully learn from their mistakes and thereby extend their knowledge and skills. The system is not intended as a replacement to classroom teaching, but as a supplementary aid to assist students who need the extra tuition and guidance which cannot be provided by the teaching staff. The paper gives first an overview of Intelligent Tutoring Systems and then describes the Intelligent Spaniard and the pedagogical strategy that it follows to provide tutoring.

Intelligent Tutoring Systems

The Domain Model

The first key place for intelligence in an Intelligent Tutoring System is in the knowledge that the system has of its subject domain. There have been three approaches [Anderson, 1988] in encoding knowledge into the domain model which gave rise to the three different types of domain models.

The first approach, which gives rise to a black box model of the domain knowledge, involves finding a method of reasoning about the domain that does not actually require codification of the knowledge. A black box model is one that generates the correct input-output behaviour over a range of tasks in the domain and so can be used as a judge of correctness. However, the internal computations by which it provides this behaviour are either not available or are of no use in delivering instruction. Such a domain model can be used in a reactive tutor that tells the students whether they are right or wrong and possibly what the right move would be. This kind of tutoring is called surface-level tutoring. The second approach, which gives rise to a glass box model of the domain knowledge, involves reasoning about the domain by applying codified knowledge. A glass box model is a standard knowledge based systems approach to reasoning with knowledge. Because of its nature, the emerging system should be more amenable to tutoring than a black box model because a major component of this expert system is an articulate representation of the knowledge underlying human expertise in the domain. For tutoring systems to be effective, it is not simply enough to understand the knowledge in the domain but also the way by which this knowledge is deployed and the restrictions levied on it. The third approach, which gives rise to a cognitive model of the domain knowledge, involves making the domain model a computer simulation of human problem solving in the domain of application.

The Student Model

The second key place for intelligence in an Intelligent Tutoring System is in the knowledge that the system infers of its user-learner. An Intelligent Tutoring System infers a student-user's current understanding of the subject matter and uses this individualised knowledge to adapt instruction to the student's needs. The component of a Tutoring System that represents the student's current state of knowledge is called the student model.

The input for diagnosis is garnered through the interaction with the student. The particular kinds of information available to the diagnosis module depend on the overall Intelligent Tutoring System application. This information could be answers to questions posed by the Intelligent Tutoring System, moves taken in a game, or commands issued to an editor. This information is sometimes complemented by the student's educational history. The output of the diagnostic module, i.e. the product of diagnosis, depends on the use of the student model. Nevertheless, it should reflect the student's current knowledge state. Some of the most common uses for the student model include advancement of the user to the next curriculum topic, offering unsolicited advice when the student needs it, dynamic problem generation, and adapting explanations by using concepts that the student understands. A student model can be classified partially according to its structural properties and partially according on properties of the input available to the diagnosis module [VanLehn, 1988]. This classification has three dimensions. One relates to the input (bandwidth), and the other two are structural properties (target knowledge type and differences between student and expert) of the student model.

Bandwidth: The input to the diagnosis unit consists of various kinds of information about what the student is doing or saying. From this, the diagnosis unit must infer what the student is thinking and believing. The bandwidth dimension is a rough categorisation of the amount and quality of the input information.

Target knowledge type: Solving problems requires some kind of interpretation process that applies knowledge present in the domain model to solve the problem. This interpretation underlies the second dimension in the student model, the "type" of knowledge in the student model. There can be two types of knowledge: procedural and declarative. Procedural knowledge is further subdivided into two subcategories: flat and hierarchical. Hierarchical representations allow subgoaling, flat ones do not. Both declarative and procedural knowledge may be represented as a hierarchical tree.

Differences between Student and Expert: Because students will move gradually from their initial state of knowledge towards mastery, student models must be able to change from representing novices to representing experts. Most Intelligent Tutoring Systems use the same knowledge representation language for both the domain model and the student model. Conceptually, an Intelligent Tutoring System has separate representations of its domain knowledge and of its student knowledge. Nevertheless, the student model is represented as the domain model plus a collection of differences. There are basically two kinds of differences: missing conceptions and misconceptions. A missing conception is an item of knowledge that the domain model has and the student model does not. Conceptually, the student model is a proper subset of the domain model. Such student models are called overlay models. With overlay models, a student model consists of the domain model plus a list of items that are missing. To model misconceptions an Intelligent Tutoring System employs a library of predefined misconceptions and missing conceptions known as the bugs library. In this case, the student model consists of an overlay model plus a list of bugs. This system performs student diagnosis by finding bugs from the library that, when added to the overlay model, yield a student model that fits the student performance.

The most common diagnostic techniques that have been used with existing Intelligent Tutoring Systems are model tracing, plan recognition, issue tracing and expert systems. With model tracing the diagnostic program traces the execution of a domain model and compares it to the student's activity. With plan recognition the diagnostic program infers a unique plan tree that spans the student's actions from a hierarchical decision tree representation of the domain knowledge. Issue tracing works by analyzing a short episode of problem solving into a set of microskills or issues that have been used or missed by the student during the episode. The idea of the expert systems approach to student diagnosis is to provide diagnostic rules for all the situations that arise.

The Tutor Model

The third key place for intelligence in an Intelligent Tutoring System is in the principles by which it tutors students and in the methods by which its applies these principles. Tutor models may incorporate many different instructional techniques. However, regardless of how tutorial interactions are conducted, a tutor model must exhibit three characteristics [Halff, 1988]:

[1] It must exercise some control over curriculum, that is, the selection and sequencing of material to be presented to the student and some control over instruction, that is the process of the actual presentation of that material to the student.

[2] It must be able to respond to student's questions about the subject matter.

[3] It must be able to determine when students need help in the course of practising a skill and what sort of help is needed.

Some tutors are primarily concerned with teaching factual (declarative) knowledge and inferential skills. These are the expository tutors. They teach students a body of factual knowledge and the skills needed are to draw first-order inferences from that knowledge. Some tutors are primarily concerned with teaching skills and procedures that have application outside the tutorial situation. These are the procedural tutors. Tutors of this kind are concerned with teaching the procedures that manipulate factual knowledge. As a result, procedural tutors function much more like coaches. They present examples to exhibit problem-solving skills, and they pose exercises for purposes of testing and practice.

The problem of curriculum can be broken down into, formulating a representation of the material in the domain model and selecting and sequencing concepts from that representation. In addition, a tutor model must also incorporate some form of propaedeutics, that is knowledge which is needed for enabling learning but not for achieving proficient performance. Curricula in Intelligent Tutoring Systems serve several functions:

[1] A curriculum should divide the material to be learned into manageable units. These units should address at most a small number of instructional goals and should present material that will allow students to master them.

[2] A curriculum should sequence the material in a way that conveys its structure to students.

[3] A curriculum should ensure that the instructional goals presented in each unit are achievable.

[4] Tutors should have mechanisms for evaluating the student reaction to instruction on a moment-to-moment basis and for reformulating the curriculum.

Propaedeutics serve to support performance up to an intermediate level. The underlying assumption is that skilled performance will be achieved only with practice. As a result, they serve, firstly, to relate theory to practice, secondly, to justify, explain, and test possible problem solutions, thirdly, as a stepping-stone to more efficient problem-solving strategies and, fourthly, as strategies for management of the working memory during intermediate stages of learning.

The tutor model may use different methods to deliver a curriculum. These methods cover initial presentation of the material, ways of responding to students' questions and the conditions and content of tutorial intervention. The methods used to present material depend on the subject matter and the instructional objectives of the Intelligent Tutoring System. Expository tutors mainly use dialogue as the form of communication whereas procedural tutors use examples and coached exercises to develop those skills. Instructional modelling, the use of worked examples or guided practice, is a prime vehicle for introducing students to procedures that they must learn. Effective answering of questions is related to the difficulty of natural language comprehension and generation which has been described as the Achilles' Heel of any effort on Intelligent Tutoring Systems development. Tutorial intervention is needed in order to maintain control of the tutorial situation to protect the student from inappropriate or incorrect learning, to keep the student from exploring paths that are not instructionally useful, and to speed the course of instruction. This involves devising rules for deciding when or when not to intervene and formulating the content of the intervention.

User Interface

Interface techniques affect two aspects of Intelligent Tutoring Systems [Miller, 1988]. First, they determine how students interact with the Intelligent Tutoring System. Second, they determine how students interact with the domain that is being tutored, through either a simulation of the domain or a direct connection to the domain itself. This interaction is generally tied closely to the tutorial component of the system so that actions in the domain are analyzed and acted upon. Based on the overall structure and orientation to the user, i.e. the perceived relationship between the user and the domain addressed by the computer system, interfaces can be divided into two groups: the interface allows users to become direct participants in the domain, or the users control the domain by instructing an intermediary to carry out actions in the domain.

In first-person interfaces or direct manipulation interfaces, the user has a feeling of working directly with the domain. These interfaces allow users to carry out desired computations by manipulating objects. Such interfaces are designed so that the actions and objects relevant to the task and domain map directly to actions and objects in the interface. The underlying mechanism behind such systems are almost always icons, i.e. small pictures on the screen which when selected by the user trigger some action. Although first-person interfaces appear to offer significant advantages to users, some aspects of the system's functionality may not be self-evident to an inexperienced user. In such cases, the Intelligent Tutoring System may have to explain the different capabilities of the system to a user. Second-person interfaces which allow the system to describe an arbitrary set of system objects in terms of the shared properties of those objects, give the user more precise control over system functionality than first-person interfaces do. With second-person interfaces, users interact with the domain by giving commands to a computerised intermediary, which then carries out the desired actions.

The Instructional Environment

The Environment Module refers to the Environment part of the system which is responsible for specifying or supporting the activities that the student does and the methods available to the student to do those activities [Burton, 1988]. It defines the kind of problems the student is to solve and the tools available for solving them. In the SOPHIE I [Brown at al, 1982] troubleshooting environment, the activity is finding a fault in a broken piece of equipment, and the primary tool available to solve the problem is the ability to ask in English for the values of measurements made on the equipment. The environment part of SOPHIE supports these activities by providing a circuit simulation, a program to understand a subset of natural language, and the routines to set up contexts, keep history lists, etc. The Environment includes some aspects of help that the system provides to the student while he is solving problems but does not include those forms of help that one would classify as requiring intelligence; these are dealt by the tutor in the box. The Instructional Environment in many ways defines the way the student looks at the subject matter [Burton, 1988].

The Intelligent Spaniard

The teaching strategy adopted by the Intelligent Spaniard is that of providing tasks for the student and evaluating his responses in order to diagnose any misconceptions in his answer. The system takes the student through a number of tests of increasing difficulty while monitoring the student's progress. The system then identifies the student's strengths and weaknesses and adapts its teaching strategy accordingly. If a misconception is found the test is temporarily suspended to provide extra tuition on a particular misconception in the hope of clarifying any difficulties the student may have, before returning back to the test. This monitoring process will continue until the system has proof that the student has reached a predefined level of knowledge and skills. This adaptive method of teaching provides a high level of individualisation. In addition, by keeping a record of a student's past interactions the system can identify an individual student should the student decide to interact with the system again and adapt its teaching strategy accordingly to suit the individual needs of the student.

The pedagogical strategy supported by the functionality of the system is best described as a continuous four-stage cycle of didactics, perception, diagnosis and remediation. The first stage, didactics, concerns tutoring the student with the domain knowledge. This task is carried out by the tutor process model. It uses information of what is being taught, from the domain knowledge model, who is being taught, from the student overlay model, and adapts the teaching strategy to suit the individual needs of the student.

The second stage, perception, concerns the actual level of understanding of the subject matter achieved of the user. This information is incorporated by the student process model in a student overlay model which represents a historical profile of the student's past performance. The student process model carries out this task continuously by reference to the domain knowledge model and to a set of overlay rules.

The third stage is diagnosis. The student process model checks a student answer for correctness against the domain knowledge model. If the answer is incorrect the student process model checks the student input against the mal-rules, an extensive library of misconceptions and errors representing possible deviations a student can make from the correct answer, to find a possible explanation for the mistake. If such an explanation is found, the student process model may call for relevant remedial action to be triggered.

The last stage in the cycle is remediation. This stage represents an attempt by the Intelligent Spaniard to clarify any misconceptions a student may have. If a misconception is identified the tutor model provides extra tuition specifically designed to test a student on a particular misconception. The aim is to clarify problems as they appear and thus not leaving the problem to deepen. The tutor process model applies the teaching strategy with the knowledge from the remedial knowledge model that has been suggested by the mal-rule that proves the misconception.

Consequently, four types of knowledge are incorporated into the tutoring system:

[1] The domain knowledge to be taught including examples and explanations, along with remedial domain knowledge that relates to possible misconceptions a student may display.

[2] Tutoring knowledge.

[3] Student overlay knowledge (this type of knowledge is organised by a set of overlay rules) including those misconceptions diagnosed for the student.

[4] A set of mal-rules that prove common misconceptions in the domain.

Three rule-based expert systems have been integrated in the Intelligent Spaniard to manipulate these four forms of knowledge and carry out the various tasks included in the four stages. The first expert system is responsible for didactics and remediation. As such it carries out all the necessary tutoring processes. Firstly, based on the student overlay model, it decides which test and to which part of the test it should take the student next. Secondly, after a misconception is diagnosed, it provides the necessary tutoring with the remedial knowledge. The second expert system is responsible for perception. As such it builds the student overlay model. The expert system decides what the level of mastery of the student is with regard to the contents of a test and allocates overlay marks in the student overlay model. The third expert system is responsible for diagnosis. As such, it diagnoses misconceptions in the student's answer and calls for remedial action.

The knowledge bases which the three expert systems manipulate are rule-based and the reasoning performed by all three of them is achieved via forward chaining (also known as data driven inferencing). The method works by taking all true facts and examining each rule to verify if all the IF parts of the rule are true. If they are then all the THEN parts are also true. With this new extended list of true facts the process is repeated until either a suitable goal has been derived or no more information can be discovered. The Intelligent Spaniard consists of various components shown in Figure 2. The instructional environment manager is responsible for communicating information to the user interface and for managing all the data structures.

The Domain Model

The domain model consists of the domain knowledge which is divided into the domain knowledge model and the remedial knowledge model and the domain process model which is responsible for manipulating the domain knowledge. The domain process model is responsible for providing a question, the correct answer, associated examples and also indicate whether the user input is correct or not. As a result, the domain process model has been programmed to function as an information retrieval mechanism.

The domain knowledge model is held in two text files. The first contains the questions for each test as well as the correct answers and a list of relevant examples for each and every one of the questions. The examples will be made available to the student on an if-needed basis. Below is a segment from this text file. The first line is the question to be displayed to the user. The word in the brackets is the verb that has to be put in the correct tense by the user. The second line is the correct answer. The remaining sentences are a list of examples which the student may ask to see, or the tutor may decide to display depending on the output of the student process model.

```
Maria (HABLAR) sobre el accidente
habla
Mi madre viaja todos los meses a Francia.
Mi padre canta todos los domingos en misa.
El chico juega en el patio.
```

The second file contains explanations for the formation of tenses. There is an explanation for each

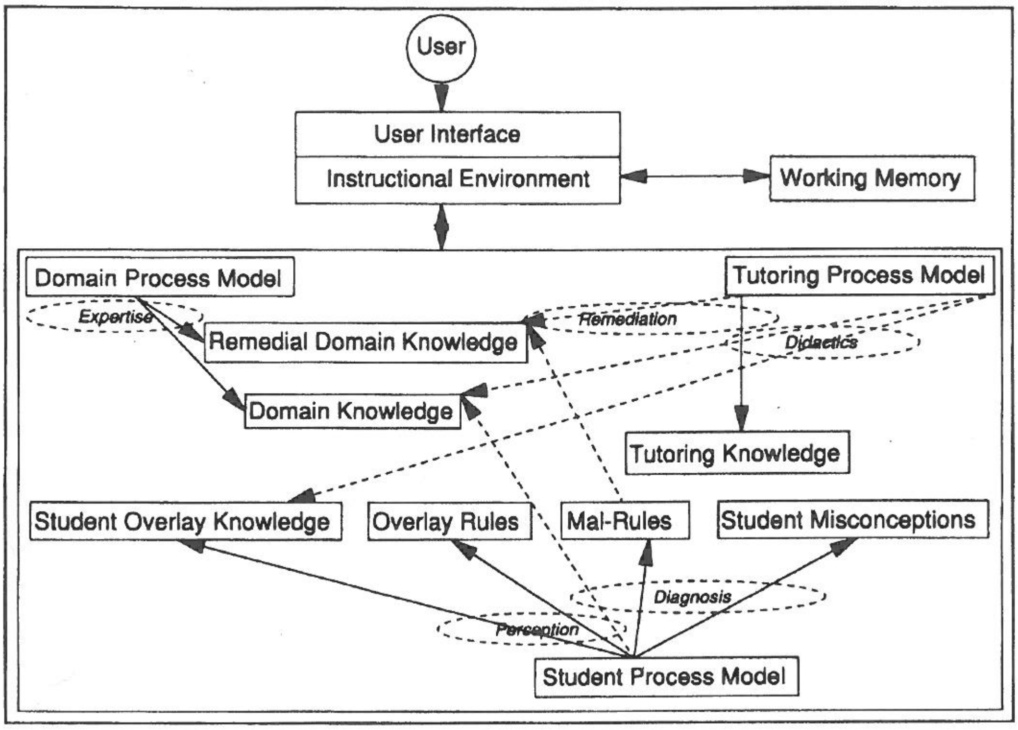

Figure 2: The Architecture of the Intelligent Spaniard

tense for each conjugation of verb. No explanations are given for irregular verbs as there are no grammatical rules to follow. The explanation below is an extract from this file.

The Present Tense : AR Verbs
The stem is formed as follows :-
Infinitive minus -ar
eg habla- (Hablar)
To the stem the add the following endings:
1st Person singular : -o
2nd Person singular : -as
3rd Person singular : -a
1st Person plural : -amos
2nd Person plural : -is
3rd Person plural : -an

The remedial knowledge model is held in a single text file that consists of questions, answers and examples. The form is similar to the first file of the domain knowledge model, the difference being that the first line contains the name of the mal-rule the question relates to. Below is an extract from this file.

RULE-5
Yo (CANTAR) en un grupo de rock.
canto
Hablo todos los das con Luis.
Escucho el aradio por las maanas.

Once a mal-rule has been fired, the test is temporarily suspended, and the tutoring process model switches to remedial mode whereby a remedial question like the one above is posed to the user. The student resumes with the test once the question has been completed. With respect to explanations provided during remedial tutoring, the same explanation file is used for the remedial domain knowledge model as with the domain knowledge model.

The Student Model

The student process model infers the student's current understanding of the subject matter. In doing so, it not only infers the student's knowledge but also any misconceptions the student may have. The model uses two knowledge sources. The overlay rules which decide which overlay mark to give the user for his answer and the mal-rules which prove a misconception in the user input and also indicate to the tutor process model what remedial action to pursue, i.e. which remedial questions to ask. The overlay rules are called every time a student answers a question. By reference to these rules the student process model allocates an overlay mark to the student between -2 and 2, with -2 being very bad and 2 being excellent. The student process model decision to allocate a particu-lar mark takes into account the number of attempts made by the student at a particular question, the number of examples seen by the student, if an explanation was seen and whether the answer had to be provided by the domain process model. It is these states that are stored in the student overlay model and drive the interaction.

The overlay rules are held in a text file and are represented as a set of IF THEN rules. That part of the student process model that fires these rules is the inference engine of the second expert system, whose function was described earlier in this section. Given below are two such rules.

RULE-8	RULE-9
IF	IF
AV = 1	AV = 1
XV = 0	XV = 1
SV = 0	EV = 0
TV = 1	SV = 0
PV = 0	TV = 1
THEN	PV = O
2	THEN
	1

Six variables are taken into account by these rules. 'AV' depicts the number of attempts a student has made at any one question. It may take a value between 1 and 3. 'XV' depicts the number of examples a student has requested to see. It may take a value between 0 and 3. 'SV' stands for spelling. If the student spells the word correctly it is set to 0 else it is set to 1. 'EV' is set to 0 if no explanation on the tenses has been requested by the student, else it is set to 1. 'TV' stands for the time taken to provide an answer to the question. There are three possible time spans: 1 for up to one minute, 2 for over one minute but less than three minutes and 3 for over three minutes. Finally, 'PV' is set to 0 if the answer is not provided to the student, else it is set to 1. Depending on the values of these six variables the final state is awarded between -2 and 2.

The student overlay model is a historical profile of a student's past performance when using the system. The model depicts the exact subset relationship between the student knowledge and the domain knowledge. The model serves as an advancement mechanism for the next question or test. The states awarded to an individual student are stored after the course of interaction in a text file. This text file is searched for by the system at the beginning of each session and its contents are retrieved to the system's student overlay model data structures. This will help the system resume tutoring from where the student left the system the last time he interacted with it. If the system does not find this file, then it assumes that the student will be a novice user.

The first line in this text file lists the student's name. Each consequent line in this file contains the states for each individual question for a particular test. A line contains eight numbers (since there are eight questions per test) between -2 and 2. If a question has not yet been attempted the state is set to 0. Currently the file consists of sixteen such lines representing the sixteen tests implemented. Below is a possible extract from such a text file.

```
Isabel
   2 1 2 1 1 1 1 2
   1 -2 -1 -1 -2 1 1 1
   -2 -2 -2 0 0 0 0 0
   0 0 0 0 0 0 0 0
```

The mal-rules are an extensive library of common misconceptions in the domain. These represent possible deviations a student can make from the correct answer. The student process model checks a student answer for correctness against this library, when his answer does not agree with the solution of the domain process model. If the answer given is incorrect the student process model runs through the mal-rules to prove a misconception. If a match is found the student process model will indicate to the tutoring process model to which part of the remedial domain knowledge, additional questions relating to this misconception can be found. The tutoring process model then takes the student through these questions.

The mal-rules are held in a text file and are represented as a set of IF THEN rules. That part of the student process model that fires these rules is the inference engine of the third expert system, whose function was also described earlier in this section. Given below are two such rules. The first IF clause is the correct answers and the remaining words are possible deviations.

RULE-1	RULE-2
IF	IF
ATLEAST 1	ATLEAST 1
habla	habla
hábla	hables
hablá	hablas
THEN	hablabas
Accents	hablo
	hablarás
	THEN
	Tenses

If the student has confused the tenses then the tutor may advise him to see an explanation on the formation of tenses to help clarify the problem. If the tutor detects a problem with accents the advice would be to check carefully accents again, or if there is some confusion on the verb endings the tutor suggests an example. In total, there are seven different misconceptions that the Intelligent Spaniard can detect and offer some advice on. The mal-rules for each test are kept in a separate text file which is accessed by the system at the beginning of a test and disposed of after the completion of the test.

A mal-rule relates to the remedial domain knowledge model via the mal-rule name. Once a mal-rule has been fired, i.e. a misconception has been proved, the student is notified that a mistake has occurred which was caused by a particular misconception. This results to remedial action been undertaken which involves providing relevant explanation or examples and additional questions that are relevant to this misconception. All the misconceptions that a student exhibits during the course of interaction are stored in a separate data structure during the course of interaction and are kept in a separate text file after the end of interaction. This text file contains a history of problems a particular student has experienced. A possible extract from such a file is given below.

```
8

Tenses
RULE-11
checked
Accents
RULE-10

pending
Accents
RULE-14
pending
```

The number depicts the number of questions contained in a particular test. If no misconception has occurred for a particular question then the file has a blank line. Otherwise the name of the misconception is listed followed by the name of the mal-rule that proves it and whether it still has to be cleared away, i.e. the 'pending' status, or has already been clarified, i.e. the 'checked' status. It is possible that more than one misconceptions may arise for any particular question. Each question being separated by a blank line.

If the Intelligent Spaniard detects a student is experiencing great difficulty with a particular problem, i.e. the same misconception arises continuously, the system prompts the student to seek further help from a hu-man tutor. The Intelligent Spaniard is not intended as a replacement to classroom teaching but as a supplementary tool to help students who need extra help and guidance. The human teacher is still to be the main source of tuition.

The Tutor Model

The tutor model consists of the tutoring knowledge and the tutoring process model which is responsible for manipulating the tutoring knowledge. The function of the tutoring process model is to decide which question to display next, in the context of a test, by applying a set of tutoring tactics to the student knowledge retrieved from the student overlay model. The student overlay model holds the marks a student has been allocated for each question attempted. The marks lie between -2 and 2, -2 being very bad, 2 excellent and 0 not attempted. The tutoring process model attempts to identify deficiencies, e.g. -2, -1 or 0s, in the student knowledge in the student overlay model and then test the user on these.

The tutor model is implemented as a rule-based expert system whereby the inference engine represents the tutoring process model, and the rules in the knowledge base are the tutoring tactics. The tutoring tactics are held in a text file as IF-THEN production rules which can have any number of IF statements and THEN statements. Given below are two such rules extracted from the file:

RULE-6	RULE-7
IF	IF
ATLEAST 1	2
2	1
0	THEN
-1	3
-2	
THEN	
2	

The first number represents the question number. The remaining numbers are possible states in the student overlay model regarding a question. All the IF clauses must be true for the THEN clause to be true except when there is an ATLEAST statement. On these occasions there must be at least as many true IF clauses as the number following the ATLEAST word.

Another useful function which the tutor model serves is to provide remedial tuition after a misconception has been diagnosed. When the student process model diagnoses a misconception in the user answer the test is temporarily suspended and the tutoring process model takes the student through an additional set of question(s) which has been specifically designed to put the student on test with the particular misconception. On completion of this remedial action the student resumes with the test. The aim of the system is to clarify a misconception as soon as it appears.

The Intelligent Spaniard sets a specific and attainable goal for the user-learner and follows a certain strategy to achieve this goal. The goal is for the student to achieve at least 90 % of the states in the student overlay model equal to 2 and the remaining 10 % equal to 1. When this goal has been achieved a certain level of competence has been achieved by the student and the student can gain nothing more by continuing with the use of the system.

The Instructional Environment

The instructional environment manager is solely responsible for the retrieve and save functions. At the beginning of the session, the instructional environment manager is responsible for loading the domain knowledge, the tutoring knowledge, the overlay rules, the mal-rules and the user's overlay knowledge and misconceptions from the appropriate text files and for constructing the relevant program data structures. During the course of interaction the instructional environment manager is responsible for communicating input-output to and from the user interface. After the end of interaction, it is responsible for saving the updated student overlay model and the student misconceptions in the appropriate text file for use in a future session.

The User Interface

The user interface is of great importance as it determines the use of the system. The users will most probably not be computer literate, therefore, an unattractive, confusing and poor quality interface may discourage users. A fast, efficient and simple to understand user interface, is required. The interface has been implemented using TURBO VISION which is based on Object Oriented Programming (OOP). The result is a windows-based interface. One window is used to display the question, another for explanation on the formation of the tenses and another the examples. No help facility on the use of the system needs to be made available as using the system should be self-explanatory. There is an option of using either a pointing device, i.e. a mouse, or the keyboard, which-ever the user feels more comfortable with. A user manual is provided with the system.

Working Memory

This holds information on the current status of interaction including current information pertaining to the student, i.e. current test and question number, etc.

Man-Machine Interaction The following is a brief description of a consultation with the Intelligent Spaniard. On calling the system, after the initial title page a menu window is displayed. The window consists of a menu bar and a statusline. The menu bar has two options, the first of which is a file option. This has three further options: retrieve (to retrieve the user file), save (to save the user file) and exit (to quit the system). The second menu bar option is a verb option which has two further options: regular or irregular verbs. These represent the two topics which the user may choose to be tested upon.

Regular verbs are categorized into three conjugations as listed below.

1. First conjugation verbs (AR verbs).
2. Second conjugation verbs (ER verbs).
3. Third conjugation verbs (IR verbs).

For each conjugation of verbs there are four tenses to be tested - the present, the preterit, the imperfect and the future. Each test is based on one conjugation and tense. A test question consists of a sentence in which the verb, given in capitals and enclosed in parentheses, is to be put by the student in the correct tense. The tests are placed in order of increasing difficulty, as are the questions within a particular test. Irregular verbs can not be divided into conjugations but only into different tenses. The form of the questions is similar to the form adopted within the regular verb tests.

When the user makes his choice of option, if he has not yet retrieved the text file containing his overlay knowledge, the system asks whether the user has used the system before. If the user has used the system before, this file is then retrieved by the system, otherwise a new file is created.

When tests commence, questions are displayed one at a time. The student may have up to three attempts at the question (the number of attempts may be very easily modified). Explanations on the formation of the tenses and examples are available on an if-needed basis. If the answer to a question provided by the student is correct the tutor advances the student onto the next question. If the answers provided on the first two attempts are incorrect, the mal-rules are checked after each attempt and hints on the possible reason for the mistake may be given in addition to some necessary remedial action. If a misconception is found the Intelligent Spaniard suspends the test temporarily and poses additional question(s) to the user which are related to the misconception diagnosed. The student may have three attempts at the remedial question and examples and explanations are available for the Regular Verb tests. However, there are no mal-rules with remedial tutoring. Therefore the user will simply be told whether the answer is correct or not. The correct answer is displayed if the student fails to provide it after the third attempt. This is also the case with the main tests. On completing the question the student is taken back to the original test and resumes with the question which he left of. On completing a test question, the student overlay model is updated based on the student's performance.

The system will guide the user through all the tests until either the student wishes to conclude the interaction with the system, or the system decides that the user has attained the system goal and no further gains can be achieved from continuing with the interaction. For a sample interaction with the system see Appendix I.

Concluding Discussion

Intelligent Tutoring Systems have the potential of a wide range of applications across a multitude of disciplines and subject areas and may affect learning and training both in schools and businesses. The most direct and obvious promise of this field is the production of systems which will be useful in helping people acquire various forms of expertise through individualised learning [Angelides and Doukidis, 1990). This new kind of educational device is regarded as a very important learning resource which the instructors can place at the disposal of their students to foster learning.

The Intelligent Spaniard is an Intelligent Tutoring System designed to assist students with learning Spanish Grammar, by testing their knowledge of regular and irregular verbs. The system is targeted at students with an intermediate level of command of Spanish vocabulary. The functional aim of the system is to reduce the gap between its own knowledge of Spanish grammar and the student's knowledge by trying to impart to the student as much of its own knowledge of Spanish Grammar as possible. The Intelligent Spaniard achieves this goal by following an explicit teaching strategy that guides the user through a series of tests each containing a set of questions and by continually assessing the student and diagnosing errors in his input in order to enable the system to tailor its questioning, explanations and remedial advice according to the needs of the individual user.

The architecture of the Intelligent Spaniard follows the general architecture of an Intelligent Tutoring System, nevertheless, adapted and enhanced to reflect the specific principles on which the Intelligent Spaniard is built. Hence, the main components in the Intelligent Spaniard's architecture are a tutor model, a student model and a domain model. The tutor model comprises the Spaniard's tutoring knowledge which is represented as a set of IF-THEN rules and the tutoring process model which manipulates these tutoring rules. The tutoring process model is implemented as a forward-chaining mechanism to work with the tutoring rules.

The student model comprises the overlay rules which are represented as a set of IF-THEN rules, the mal-rules which are also represented as a set of IF-THEN rules and the student process model which manipulates both the overlay rules to produce the student-user's overlay model and the mal-rules to diagnose misconceptions in his input. The student process model consists of two forward-chaining mechanisms, one to work with the overlay rules and one to work with the mal-rules.

Finally, the domain model comprises the domain knowledge which is represented as a set of tests each containing a number of questions along with their answers and relevant examples and explanations and the domain process model whose purpose is to manipulate these. The domain process model mainly serves as information retrieval mechanism.

The Intelligent Spaniard was not developed to take over the role of a human teacher. Instead, it was developed to provide the students with a complementary facility for practising their Spanish grammar skills which should have already been practised in the classroom. Explanations provided during the course of interaction should be sufficient enough to help learn or remind Spanish grammar rules.

The adaption of the system to be more flexible by allowing the user to ask what-if questions and to initiate their own area of investigation would provide a very interesting ground for a long-term research and devel-opment in the future. However, because of the nature of the domain of application, enhancing the system towards this direction may at this stage of research and development into Intelligent Tutoring Systems be overambitious. Currently, the system has a very limited knowledge about the subject. It stores questions and answers, but it cannot use this knowledge to answer arbitrary or hypothetical questions or tailor its explanations to suit the user needs. As a result, the system, being in control, tests the student on those topics on which he is diagnosed to be in need of help.

Short-term research and development to enhance the system performance may be undertaken in several areas. With student modelling, the mal-rules may be extended to uncover a wider range misconceptions. Currently the system is able to uncover seven classes of misconceptions only. The user interface may be enhanced to support a much more "natural" conversation between the student and the system, beyond the short answer or menu-driven interactions of the Intelligent Spaniard. In addition, the topics covered by the Intelligent Spaniard can be extended to cover other areas of Spanish grammar such as adjectives, pronouns and adverbs. Another improvement may be made in the form of providing assistance on the meaning of question sentences, for those students that may be struggling with Spanish vocabulary. Possible implementation would involve an on-line dictionary of the vocabulary used, that would enable the system to give the meaning of a word in the mother language of the student.

A major limitation of the system is that there exist no explanations for irregular verb tests. This is due to the fact that irregular verbs do not follow any grammatical rules. Furthermore, no examples are given for irregular verb tests as this would involve giving away the correct answer. A further improvement that can be made is in the form of providing explanations based on particular verbs rather than on each of the tenses.

The Intelligent Spaniard has been developed as a generic system. This would allow the system to offer tuition in other European languages. No extra programming will be needed to implement this idea as the data structures and the mechanisms for manipulating them are domain-independent. The only requirement would be the addition of the domain knowledge. This may be a part of a bigger project whose objective would be to transform the Intelligent Spaniard to become an authoring shell for European languages. The system would act as a general purpose shell which, when supplied with the database specific to a language, could teach grammar for that language. Hence, the system would provide a facility enabling the rapid development of an Intelligent Tutoring System in the user's own choice of language.

References

1. J.R. ANDERSON (1988) The Expert Module. In *Foundations of Intelligent Tutoring Systems* (M.C. POLSON and J.J. RICHARDSON, Eds.), pp. 21-53. Lawrence Erlbaum Associates, USA.
2. M.C. ANGELIDES and G.I. DOUKIDIS (1990) Is There a Place in OR for Intelligent Tutoring Systems? *Journal of the Operational Research Society* 41 (6), 491-503.
3. J.S. BROWN, R.R. BURTON and J. de KLEER (1982) Pedagogical, natural language and knowledge engineering in SOPHIE I, II and III. In *Intelligent Tutoring Systems* (D.H. SLEEMAN and J.S. BROWN, Eds.), pp. 227-282. Academic Press, USA.
4. H.L. BURNS and C.G. CAPPS (1988) Foundations of Intelligent Tutoring Systems: An Introduction. In *Foundations of Intelligent Tutoring Systems* (M.C. POLSON and J.J. RICHARDSON, Eds.), pp. 1-19. Lawrence Erlbaum Associates, USA.
5. R.R. BURTON and J.S. BROWN (1982) An investigation of computer coaching for informal learning activities. In *Intelligent Tutoring Systems* (D.H. SLEEMAN and J.S. BROWN, Eds.), pp. 79-98. Academic Press, USA.
6. R.R. BURTON (1988) The Environment Module of Intelligent Tutoring Systems. In *Foundations of Intelligent Tutoring Systems* (M.C. POLSON and J.J. RICHARDSON, Eds.), pp. 109-142. Lawrence Erlbaum Associates, USA.
7. G.I. DOUKIDIS, M.C. ANGELIDES and J.L. HARLOW (1988) Towards an Intelligent Tutoring System for Pascal Programming. *Education and Computing* 4 (4), 273-286.
8. H.M. HALFF (1988) Curriculum and Instruction in Automated Tutors. In *Foundations of Intelligent Tutoring Systems* (M.C. POLSON and J.J. RICHARDSON, Eds.), pp. 79-108. Lawrence Erlbaum Associates, USA.
9. J.R. MILLER (1988) The Role of Human-Computer Interaction in Intelligent Tutoring Systems. In *Foundations of Intelligent Tutoring Systems* (M.C. POLSON and J.J. RICHARDSON, Eds.), pp. 143-189. Lawrence Erlbaum Associates, USA.
10. D.H. SLEEMAN and R.D. WARD (1988) Intelligent Tutoring Systems in Training and Education: Prospects and Problems. In *Research and Development in Expert Systems* 5 (B. KELLY and A. RECTOR, Eds.), pp. 331-343. British Computer Society Workshop Series, UK.
11. M.L. SWARTZ and M. YAZDANI (1992) *Intelligent Tutoring Systems for Foreign Language Learning: The Bridge to International Communication.* Springer-Verlag, Germany (1992).
12. K. VANLEHN (1988) Student Modelling. In *Foundations of Intelligent Tutoring Systems* (M.C. POLSON and J.J. RICHARDSON, Eds.), pp. 55-78. Lawrence Erlbaum Associates, USA.

APPENDIX I

Student:	a:\Spaniard
Spaniard:	Introduction Window
Student:	ALT V
Spaniard:	Have you used this package before?
Student:	Yes
Spaniard:	The user file has not yet been retrieved.
	The file will now be retrieved.
Spaniard:	Hello Isabel.
	The lesson will now begin.
	Nosotros (HABLAR) de la situación en Russia.
Student:	hablamos
Spaniard:	The answer is incorrect.
	An explanation may be of some use.
Student:	OK
Spaniard:	THE IMPERFECT TENSE : AR VERBS
	The stem is formed as follows:
	Infinitive minus -ar
	eg. habl- (Hablar)
	To the stem add the following endings:
	1st Person singular - aba
	2nd Person singular - abas
	3rd Person singular - aba
	1st Person plural - ábamos
	2nd Person plural - abais
	3rd Person plural - aban
Student:	OK
Spaniard:	You are having some difficulty with tenses.
	Extra questions will be supplied to help you.
Student:	OK
Spaniard:	Ayer (CELEBRAR) el cumpleaños de nuestra madre.
Student:	celebrábamos
Spaniard:	The answer is correct.
Student:	OK
Spaniard:	Nosotros (HABLAR) de la situación en Russia.
Student:	Hablábamos
Spaniard:	The answer is correct.
Student:	OK

Spaniard: Tú (TRABAJAR) en un banco ¿no?
Student: trabajabas
Spaniard: The answer is correct.
Student: OK
Spaniard: De pequeños nosotros (TIRAR) piedras a ventanas.
Student: Examp
Spaniard: Faltábamos de clase muy a menudo.
Student: OK
Spaniard: Do you want another?
Student: Yes
Spaniard: Cantábamos en el coro durante cinco anos.
Student: OK
Spaniard: Do you want to see another?
Student: No
Spaniard: De pequeños (TIRAR) piedras a ventanas.
Student: tirábamos
Spaniard: The answer is correct.
Student: OK
Spaniard: Usted (FALTAR) del grupo pere tuvimos que marchar.
Student: ESC
Spaniard: Do you want another test?
Student: No
ALT X
Spaniard: Have you saved your file?
Student: No
Spaniard: Saving user file.

Received: August 15, 1992
Accepted: March 4, 1993

Address for correspondence:
Marios C. Angelides
Lecturer in Information Systems
Information Systems Department
The London School of Economics and Political Science
Houghton Street
London WC2A 2AE
phone: (071)955-7628
fax: (071) 955-7385

Marios Angelides is a Lecturer in the Information Systems Department at the London School of Economics. He holds a B.Sc. degree in Computing and a Ph.D. in Information Systems both from the London School of Economics. His research interests include Intelligent Tutoring Systems, Gaming-Simulation and Multimedia/Hypermedia. He has published several papers in the above areas and he is the co-author of the book, Lisp: From Foundations to Applications published in 1988. He is Vice-Chairman of IFIP's (International Federation for Information Processing) Working Group 9.5: Social Implications of Artificial Intelligence Systems and he is the Chairman of the Second International IFIP Conference on Opportunities and Risks of Artificial Intelligence Systems which will take place in London in 1993.

Isabel Garcia is a researcher in the Information Systems Department at the London School of Economics. She holds a B.Sc. degree in Economics with special subject Computing from the London School of Economics. Her research interests include Intelligent Tutoring Systems, Knowledge Based Expert Systems and Computer Aided Education applied to Foreign Language Learning.