University of
Bedfordshire

Title   Adaptive Intelligent Tutoring for Teaching
           Modern Standard Arabic

Name   Dawod Kseibat

Adaptive Intelligent Tutoring for Teaching Modern Standard Arabic

Dawod Kseibat

A thesis submitted to the University of Bedfordshire, in partial fulfilment of the

requirements for the degree of Doctor of Philosophy

June 2010

ABSTRACT

The aim of this PhD thesis is to develop a framework for adaptive intelligent tutoring systems (ITS) in the domain of Modern Standard Arabic language. This framework will comprise of a new approach to using a fuzzy inference mechanism and generic rules in guiding the learning process. In addition, the framework will demonstrate another contribution in which the system can be adapted to be used in the teaching of different languages. A prototype system will be developed to demonstrate these features. This system is targeted at adult English-speaking casual learners with no pre-knowledge of the Arabic language. It will consist of two parts: an ITS for learners to use and a teachers' tool for configuring and customising the teaching rules and artificial intelligence components among other configuration operations. The system also provides a diverse teaching-strategies' environment based on multiple instructional strategies. This approach is based on general rules that provide means to a reconfigurable prediction. The ITS determines the learner's learning characteristics using multiple fuzzy inferences. It has a reconfigurable design that can be altered by the teacher at runtime via a teacher-interface. A framework for an independent domain (i.e. pluggable-domain) for foreign language tutoring systems is introduced in this research. This approach allows the system to adapt to the teaching of a different language with little changes required. Such a feature has the advantages of reducing the time and cost required for building intelligent language tutoring systems. To evaluate the proposed system, two experiments are conducted with two versions of the software: the ITS and a cut down version with no artificial intelligence components. The learners used the ITS had shown an increase in scores between the post-test and the pre-test with learning gain of 35% compared to 25% of the learners from the cut down version.

DECLARATION

I declare that this thesis is my own unaided work. It is being submitted for the degree of Doctor of Philosophy at the University of Bedfordshire.

It has not been submitted before for any degree or examination in any other university.

Name of candidate: Dawod Kseibat

Signature:

Date:

DEDICATION

This thesis is dedicated to my mother, Rabia, and my late father, Saad Kseibat, who have raised me to be the person I am today. To my wife, Hana, and my kids, Blgeas and Hashem. Thank you for everything.

ACKNOWLEDGEMENTS

LIST OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AI: Artificial Intelligence.

ANF: Average number of feedback.

ANH: Average number of hints.

ANN: Artificial Neural Network.

AS: Arabic spelling.

ASR: Automated speech recognition.

AUD: Audio.

AWD: Average number of working days.

AWT: Average working time.

CBM: Constraint-Based Modelling.

CM: Course manager.

CMS: Course Management System.

CSBM: Case-Based Modelling.

DAT: Discovery average time.

DESC: Description.

DFD: Data-flow diagram.

DK: Domain knowledge.

DUALF: Dual feminine.

EAG: Examination average grade.

Eg: Engagement.

ENG: English.

ERD: Entity relation diagram.

FEM: Feminine.

FF: First form.

Fpdf: First-person dual feminine.

Fpdm: First-person dual masculine.

Fppf: First-person plural feminine.

Fppm: First-person plural masculine.

Fpsf: First-person singular feminine.

Fpsm: First-person singular masculine.

GUI: Graphical user interface.

Hs: Help-seeking.

ICALL: Intelligent computer-assisted language learning.

ILTS: Intelligent language tutoring systems.

ITS: Intelligent Tutoring System.

IND: Index.

IOVs: Input/output variables.

LDB: Learning database.

LI: Learner-interface.

LMS: Learning Management System.

MSA: Modern Standard Arabic.

MT: Machine Translation.

NL: New language.

NLP: Natural Language Processing.

PI: Practice interface.

Picture: PIC.

Pr: Performance.

PLUR: Plural.

PLURF: Plural feminine.

QAG: Question average grade.

R: Reading.

RQ: Reading questions.

RP: Reading part.

RQT: Reading question type.

RQFA: Reading question first answer.

RQTA: Reading question third answer.

RQSA: Reading question second answer.

RQRA: Reading question fourth answer.

RT: Root.

SCORM: Shareable Content Object Reference Model.

SF: Second form.

SLG: Successful login.

Spdf: Second -person dual feminine.

Spdm: Second -person dual masculine.

Sppf:	Second -person plural feminine.
Sppm:	Second -person plural masculine.
Spsf:	Second -person singular feminine.
Spsm:	Second-person singular masculine.
TI:	Teacher-interface.
TLTS:	Tactical Language Training System.
TOF:	Type of feedback.
TOH:	Type of Hints.
Tpdf:	Third -person dual feminine.
Tppm:	Third -person plural masculine.
Tppf:	Third -person plural feminine.
Tppm:	Third -person plural masculine.
Tpsf:	Third -person singular feminine.
Tpsm:	Third-person singular masculine.
TRNS:	Transliteration.
TUs:	Teaching units.
Video:	VID.
VN:	Verbal noun.
VS:	Values.
VSi:	ith Values.
WBITSs:	Web-based intelligent tutoring systems.
WWW:	World Wide Web.

CHAPTER 1: INTRODUCTION

1.1   Background to the research

An Intelligent Tutoring System (ITS) refers to educational software containing some artificial intelligence components. ITSs are computer-based instructional systems that employ theories from three disciplines: Education, Computer Science and Psychology. The interaction between education, computer science and psychology is shown in Figure 1.1. *Education* represents the teacher and the teaching strategy of the ITS. *Psychology* represents the learner interaction with the ITS in the form of the learner model. *Computer Science* represents the techniques and methods used by the ITS in the learning process. The goal of ITSs is to engage the learners in continuous reasoning activities and interact with them based on the understanding of their behaviour.
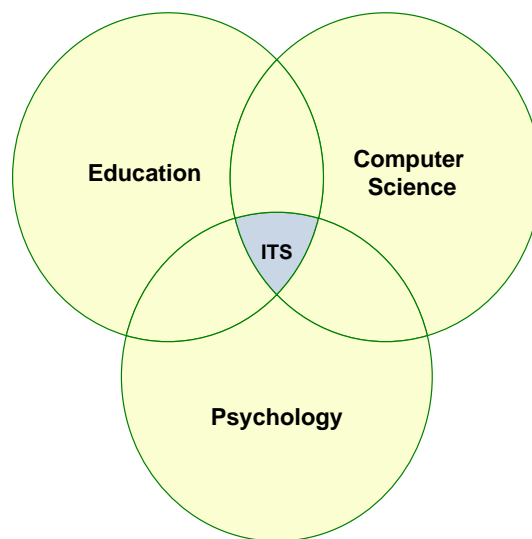
Figure1.1: Intelligent tutoring system interactions

ITSs have been implemented successfully in the teaching of different subjects including mathematics, physics, medical informatics, and computer science (Anohina, 2007). The advent of the Internet, the digital multimedia has made the Web a preferred platform for the delivery of the learning materials since it provides a rich context for self-paced instruction. Web-based tutoring systems overcome the

challenges posed by the traditional paper-based and classroom learning such as location and time by exploiting the various advantages of Internet technologies (e.g. classroom independence and platform independence). Web-based tutoring systems facilitate different forms of learning: just-in-time learning, on demand learning, any time learning, and lifelong learning. Web-based ITSs represent a way to integrate the intelligence of an ITS with the advantages of WWW applications.

Computers are widely used in the teaching of natural languages since they can make the learning process dynamic and rich (Levy and Stockwell, 2006). Intelligent language tutoring systems (ILTS) and intelligent computer-assisted language learning (ICALL) are disciplines that apply the technology of Artificial Intelligence (AI) to enhance the language learning and teaching (Godwin-Jones, 2009). AI allows for more varied and personalised interaction with the user. Although ICALLs are very useful, there is still great potential in the new technologies which has to be explored. Previous research has identified several approaches to the learning and teaching of foreign languages such as Natural Language Processing (NLP) (Shaalan, 2005; Farghaly and Shaalan, 2009), Automated speech recognition (ASR) and Machine Translation (MT) (Abdel Monem *et al.*, 2008).

Most of the ILTS and ICALL systems use NLP. However the applications of these technologies have largely focused on the acquisition of grammatical structures with little attention given to analysing students' interaction with the system. Although some of these systems are rather promising, additional research efforts are required in order to overcome the limitations of these systems.

1.2   Natural languages

In this research an adaptive tutoring system for the teaching of Arabic will be developed. The system will be designed to be adapted for the teaching of different languages (e.g. French) based on the concept of pluggable domain introduced later in this thesis. The next section reviews both the Arabic and French languages' grammar. The general structure of the natural language grammar used in this research is shown in Figure 1.2.

Arabic is a Semitic language, and it has been used since the 4th century. Arabic script is cursive and most characters are connected to each other. The same Arabic characters can take different shapes depending on their position in the text (i.e. beginning, middle, end, or alone). Also, the Arabic alphabet contains few letters that does not exist in other languages or cannot be easily pronounced except with training and practice (e.g. Arabic letters (ق) and (ظ), pronounced in English like 'Khaa' and 'Thaa').

The Modern Standard Arabic (MSA) or *Al-Fus-ha* is the formal form of the Arabic language that is understood by all Arabic speakers and it is the form used by the mass media (newspapers, television, or radio). Arabic is the language of the Qur'an, the holy book of Islam and it is widely used throughout the Muslim world. More than 300 million people around the world speak the language (Cutshall, 2007).

French is a Romance language descended from Latin. It is often the language used in diplomatic work as it is an official language of the United Nations.
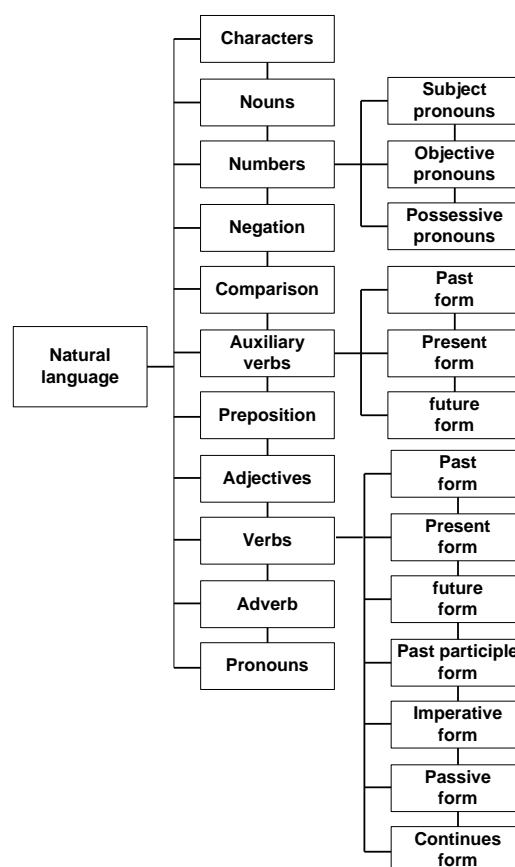
Figure 1.2: The general structure of the natural language grammar

There are 28 characters in the Arabic alphabet. These characters are read and written from right to left. There is no distinct upper and lower case letter forms in this language. Both printed and written Arabic are cursive and each individual character can have up to four distinct forms based on its position within a word (i.e. beginning, middle, end, alone). Figure 1.3 presents different forms of the Arabic character "Kaaf" (in Arabic: ك). The present Arabic numerals' system is Hindu-Arabic originally invented by the Hindus in India in the 4th century BC. Since then this system spread to the Middle East in about the 9th century AD, where it was used by the Arabs.

| End | Middle | Beginning | Alone |
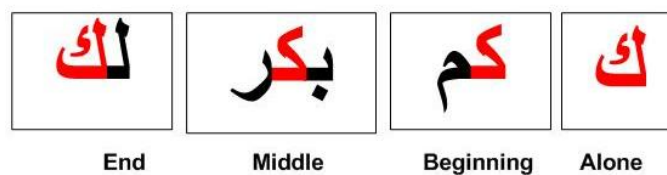
Figure 1.3: The different forms of the Arabic character "Kaaf" based on its position

French has 26 letters similar to the English language where most of them have different pronunciation. The language has several accents which are á, è, ù, and é.

Arabic nouns as with the French have two genders: feminine and masculine. The feminine form in Arabic is formed by adding (the ta marbuta; in Arabic ة) to the end of masculine form of the noun. Moreover adding the prefix (al; In Arabic: ال) to the beginning of the noun (e.g. كتاب) change its format from indefinite to define form (e.g. الكتاب) (i.e. like "the" in English language). On the other hand the French language has two definite articles (i.e. le or la) depends whether the noun is masculine or feminine.

With reference to the English language, Arabic and French verbs can take different forms such present, past and continuous (see Figure 1.2). Most verbs in the Arabic language have three letter roots (i.e. like "to draw" in English; in Arabic: رسم pronounced "resim"). Auxiliary verbs in both Arabic and French precede the main verb and they are used to assist the main verb. They are also used in the negation form and for asking questions.

Arabic and French prepositions are used to link nouns and phrases to other words in the sentence. Prepositions usually come before nouns like the preposition "around" in the sentence "around the house" (in Arabic: حول المنزل). In French the preposition "from" in the sentence "he is from Paris" come before the noun Paris (in French: de Paris).

Arabic adjectives follow the noun they modify as the adjective "big" in the sentence "big house" (in Arabic: سيارة كبيرة ; in French: grande voiture). An adverb is used to modify phrase or clause to indicates place, cause or manner like "always" (in Arabic: دائما; in French: tout le temps). A pronoun is used to substitute a noun or noun phrase such as "I" (in Arabic: انا; in French: je) and "He" in English (in Arabic: هو ; in French: il).

This research will be limited to languages that are similar in structure to those discussed above. Any other languages that do not fit this structure will be out of this research scope.

1.3   Motivation

The motivation of this research can be summarised as follows:

i. To develop a generic natural language teaching framework that can be adapted to teach different languages. This framework will comprise of new design approaches to create Intelligent Language Tutoring Systems. The framework also aims to reduce the time and cost of building such systems.

ii. To overcome the limitations of the current Arabic language teaching systems (Shaalan, 2005), which lack the adaptability and the intelligence that can be provided by the proposed framework. Such systems cannot provide personalised instructions or feedback to learners (i.e. without the intervention of human teachers) whilst performing learning tasks.

iii.   To provide Arabic language teaching to a large number of learner populations around the world who need to learn Arabic for different reasons such as business or religion.

1.4   Research Aims and Objectives

Research Aims:

I.   To provide a positive learning experience based on a comprehensive Arabic language teaching environment for adult learners. The learning environment supports multiple teaching strategies which informed by reflective practices. This will help to create educational techniques that recognise different aspects of learner's personality and knowledge.

II.   To accomplish the learning and teaching of this environment through the combination of fuzzy logic and generic rules. The environment will provide support that reflects the learner's knowledge and personal disposition. The aim is to help learners to acquire knowledge and skills that are compatible with their understanding and problem-solving skills.

Research Objectives:

The objectives of this research are:

1. To investigate the current research into intelligent tutoring systems with emphasis on language learning systems
2. To analyse the requirements for developing a framework for intelligent language learning systems using artificial intelligence features
3. To design new techniques for customised learner modelling and teaching strategies using artificial intelligence methods
4. To construct a framework for developing a web-based ITS with the following features:

- Has the ability to provide differentiated instruction using multiple instructional strategies (MIS). Practices, extra tutoring, prerequisites, feedback and hinting are embedded in the design of the instructional strategies. Generic rules are developed to perform the selection and modification of the attributes and features of each instructional strategy. This will provide a framework of instructional capability by integrating Artificial Intelligence into natural language learning systems to assist the learner to learn the teaching materials and the concepts presented.

- Able to determine the learner's learning characteristics using Stereotypes' learning modelling technique. Multiple fuzzy inferences (Stereotypes) are used to represent each stereotype since fuzzy logic provides human-like evaluation of the learner different characteristics such as performance and help-seeking. This provides a great advantage over other techniques which deal with human problems.

- Possesses a design model for an independent domain (pluggable-domain) for foreign language tutoring systems in which the system can be adapted to teach a different language with minimal changes required. The design model is based on the reusability of templates (i.e. lessons and question) and a general Domain Knowledge structure that can be used in the teaching of other languages. Moreover a conversion process is used to provide mapping between the source and target languages.

5. To evaluate the developed prototype in an educational environment.

1.5 Research methodology

The design approach of the research takes into consideration the aims and objectives of this work covering literature review, system design, implementation, and evaluation of the results and analysis. Previous research, systems and techniques developed for designing ITSs in general are reviewed with focus on ILTSs and ICALLs. The research problem is defined in response to the proposed aims and objectives. An ITS for teaching Arabic is formulated based on the typical

components of ITSs: the student model, the teacher model, and the domain knowledge.

The research also involved a study of the Arabic and French languages' structures and features as part of building the system's domain knowledge.

The processes involved in this research are illustrated in Figure 1.4. These processes are system interaction, learning, learner's stereotyping, system support and the evaluation as discussed in the next sections.
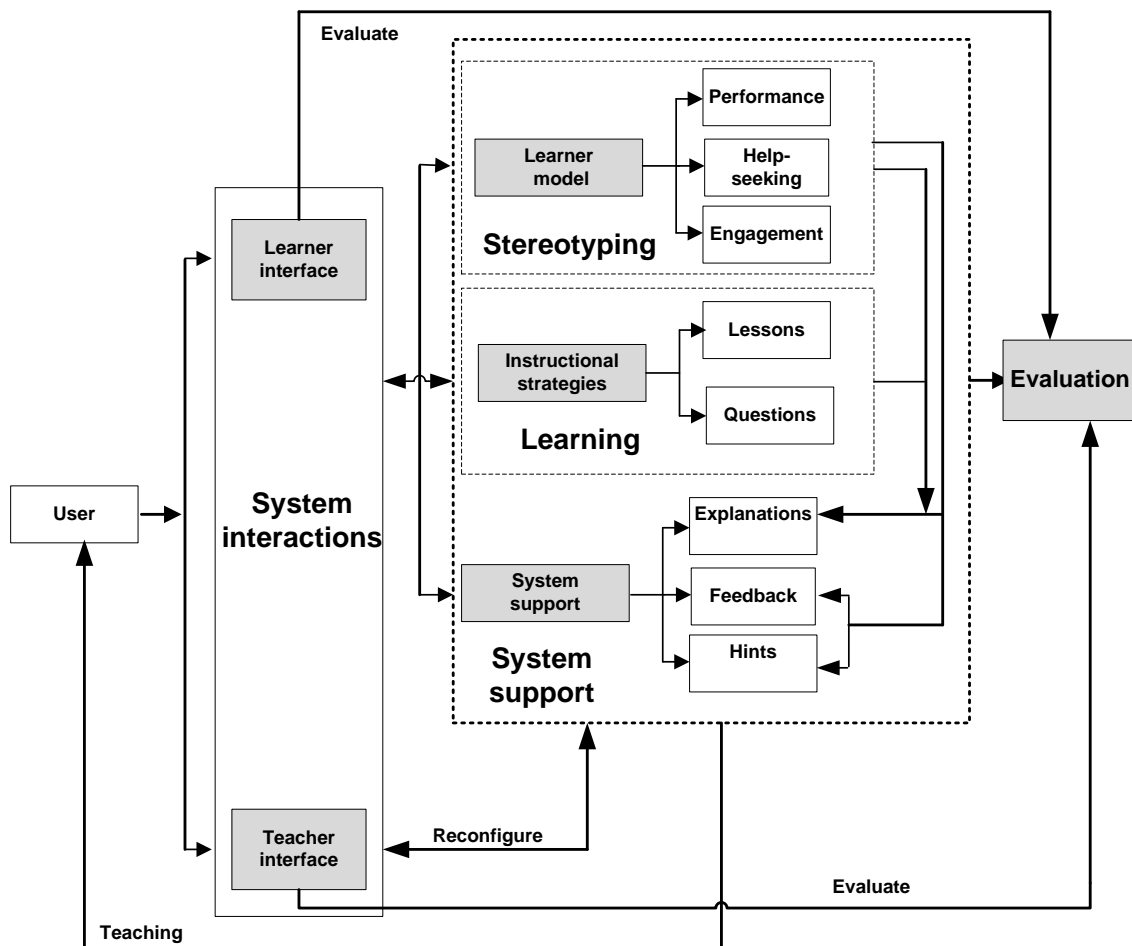
Figure1.4: the processes of the research

I.   The system interactions process provides communications between the user (i.e. Learner or teacher) and the tutoring system via a graphical user interface (GUI) (see Figure 1.4). The learner interfaces will be used to present teaching materials

8

in different formats (i.e. text, pictures and videos), examinations and system support. The teacher interface will allow the teacher to reconfigure various components of the tutoring system (i.e. learner model and instructional strategies). Since the developed system is a prototype and the research will be fundamentally looking at the design features therefore Human Computer Interaction (HCI) principles will not be considered.

II. The learning process is responsible for providing teaching to the learner via learner interface (see Figure 1.4). Previous research revealed that most of the natural language tutoring systems, Intelligent Computer Assisted Language Learning (ICALL) and Intelligent Language Tutoring System (ILTS), are grammar-driven and their selection of the teaching materials follow curriculum sequencing and task-based approaches (Shaalan, 2005; Amaral, 2007). Therefore a new framework for selecting the proper instructional strategy will be developed in this research. The framework is based on the contents and attributes of the current teaching materials (i.e. lessons or questions) such as difficulty and learning levels. The feature of each instructional strategy can be reconfigured by the teacher via teacher-interface at runtime.

III. The stereotyping process assesses the learner's previous and current interaction. The learner model will be based on stereotype learner modelling (Chrysafiadi and Virvou, 2008) implemented as multiple-fuzzy inferences since fuzzy system provides human-like evaluation of specific characteristics of the learner (see Figure 1.4). The design of each fuzzy inference will be based on a reconfigurable design that can be altered to be adapted to different learning situations.

IV. The system support process will provide on-demand explanations, feedback and hints. The goal is to optimise of the process of giving some help to the learner without letting them rely completely on that help (Baker, 2007). This process will be based on the complete characteristics of the learner profile. On the other hand the explanation provides information on the current instructional strategy and the learner profile (see Figure 1.4).

V. In the evaluation process, all the system components will be implemented using PHP scripts, and MySQL database running on an Apache server. The key problems to be addressed in the implementation of the tutoring system are:

- Defining the data storage for various components of the systems teaching materials such as lessons and examinations.

9

- Defining the data storage for learner.
- Developing various algorithms for handling system control components such as fuzzy inference and instructional strategies.

Various experiments will be carried out to evaluate the system. System performance will be evaluated by experimental results, statistical analysis and learners' questionnaires. A paired t-test will be used to indicate the significant differences between the pre-test and the post-test achievements (Chien, 2008). Moreover the learning of the tutoring system will be tested by teachers and their experience will be recorded by the teachers' questionnaires.

1.6   Organization of the Thesis

The following chapters are organised as follows:

Chapter 2 surveys previous and related work in the area of ITS and its applications. The relevant research is divided into different topics: Web-based intelligent tutoring systems, CALL, ICALL, and ILTS systems, Fuzzy systems, Learner modelling, Help-seeking and gaming, and authoring tools and shells.

Chapter 3 describes the overall design of the tutoring system which includes the modes of interaction with users, system support, design of the instructional strategies, system architecture, pluggable domain, and curriculum design.

Chapter 4 introduces the design of the Course Manager (CM) and its components which includes the learner and the instruction models.

Chapter 5 describes the overall design of the system knowledge and its components which includes the learning database and the domain knowledge (DK).

Chapter 6 introduces the concept of the pluggable domain as well as highlights the advantages and disadvantages of this concept.

Chapter 7 summarises and discusses the implementation of the components of the tutoring system.

Chapter 8 presents the evaluation of results and discusses the finding of the evaluation. Pre-test and post-test analysis combined with questionnaire responses are used to evaluate the tutoring system.

Chapter 9 discusses and summarises the main contributions of this research and proposes some future work related to this work.

CHAPTER 2: RELATED RESEARCH

This chapter presents a review of the literature related to this research. Moreover it provides a finding of the current theoretical and methodological contributions to the field of natural language intelligent tutoring systems.

2.1   Background

While existing Intelligent Tutoring Systems (ITSs) vary in their structure, they typically consist of at least three basic components or subsystems. Hartley and Sleeman (1973) described the requirements of an ITS for the first time. An ITS relies on three components which can be described as follows:

1. The Domain model (or Domain knowledge) that contains the knowledge of certain domain (e.g. Physics).
2. The Learner model that contains the learner knowledge and behaviour.
3. The Pedagogical model that contains the expertise and teaching strategy of the human teacher in the area of the domain. Figure 2.1 shows the components of an ITS. The interaction between the learner and the ITS is provided via a user interface.
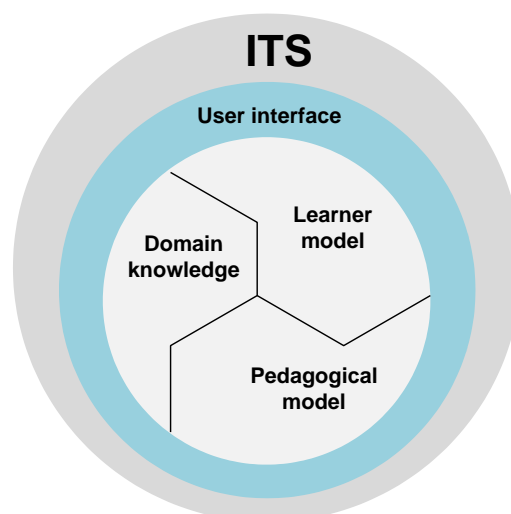
Figure 2.1: The components of an ITS

There has been continuous research in ITS over the past thirty years with some notable successes (Abu Naser, 2008). ITSs have provided a remarkable educational gain for learners from different knowledge domains (Chien *et al.*, 2008). Researchers have investigated ITSs as the means of providing one-to-one tutoring. ITSs have improved learning for students in difficult subject domains such as mathematics (i.e. algebra and geometry) as compared with traditional class-room instruction (Feng *et al.*, 2008).

ITSs have been used in a variety of applications such as virtual reality educational games (Virvou *et al.*, 2005), and physics (Makatchev *et al.*, 2006). Different computational techniques such as artificial neural networks, production systems, Bayesian networks, and fuzzy systems have been used in these systems (Chang *et al.*, 2008).

ITSs contain models and strategies that specify what and how to teach and simulate the teacher's behaviour during the learning process. Broadly defined, ITSs fall into the "problem-based learning" or "learning by doing" categories. ITSs provide learners with a series of tasks to accomplish and as the learner works through the system, the ITS tracks his/her learning and provides him/her with personalised guidance when he/she needs it.

2.2   Instructional strategies

Online pedagogical strategies used to date such as model tracing (Aleven *et al.*, 2006b), coaching, guided discovery and collaborative learning (Gregory and chapman, 2007; Luppicini, 2007) have been derived from traditional teaching strategies (Anderson and Elloumi, 2004).

Strategies like model tracing and coaching usually take full control of the learning process with some variations from one strategy to another on the type and the timing of help provided to learners. However, an advanced learner with some background knowledge may find these strategies restricting and not challenging. Moreover, these strategies may encourage shallow learning which is characterised by the ability to

memorise information. On the other hand the discovery learning strategy which operates by providing the learner with the freedom to work in an unconstrained environment and learning from experience is not suitable for beginners who need more supervision and support (Veermans *et al.*, 2006). The Collaborative learning strategy is based on the philosophy of learning from peers where a number of learners have to be "logged-in" concurrently in order to benefit from this learning strategy (Wheeler *et al.*, 2008).

Learners learn a second language with different ability levels, cultural backgrounds, and learning profiles. They need a supportive learning environment that promotes diversity, learning at various rates, and in different ways. These learners need a variety of tasks, flexible learning options, and challenges to demonstrate their capabilities. They also need to experience differentiated instruction or personalised instruction (Anderson, 2007). Differentiated learning is a philosophy of teaching and learning which is based on the belief that each learner is unique. However, in differentiated instruction learners are not doing exactly the same things at the same time by not changing what is taught but changing how it is taught. Differentiation is an efficient way to offer meaningful instruction by providing challenging contents that meet the needs of learners at their suitable levels (Dunn and Honigsfeld, 2009). In the proposed system the differentiated instruction is adapted for the teaching and learning process.

A successful intelligent natural language teaching system should be able to address different learners learning styles; offer help when needed, provide logical next steps, and give appropriate feedback. Such a system is like a good teacher, one that should be able to treat learners as distinct individuals and provide personalised learning. To create these systems, a mix of expertise is required in fields like Artificial intelligence, education, learning psychology, and computer science. Creating these systems is not an easy task given that many projects never reach production status (Godwin-Jones, 2009).

2.3   E-learning

E-learning is a computer and network-enabled transfer of knowledge and skills. It can be a self-paced and it may include multimedia components like audio, video and animations. Web-Based Training, Computer-Based Training and Internet-Based Training are considered forms of E-learning (Hrastinski, 2008). E-learning is the fastest growing sector of the global education market with higher education expected to grow to $69B by 2015 (Wagner *et al.*, 2008).

As the hardware prices decrease and computer literacy increase more people gain access to the Internet. This provides new trends for the delivery of educational contents by the educational institution. These days more and more higher education institutions implement E-learning to create exciting opportunities for students. However the success of E-learning is dependent on the extent to which the needs and concerns of the groups involved are addressed. Most of the current E-learning systems are still delivering the same educational resources in the same way to learners with different backgrounds (Khribi *et al.*, 2009). Adaptive course delivery is the most common personalisation techniques applied in E-learning systems today which include dynamic course re-structuring and adaptive selection of learning materials and adaptive navigation (Khribi *et al.*, 2009).

As in any courses that are totally electronic, students need to be highly motivated and committed to learning. Without such attitudes, there will be a higher rate of incomplete students' grades and leading to withdrawal (Wagner *et al.*, 2008). This will help to take E-learning beyond static web pages into a teaching that based on the strength and weakness of each student.

Most E-learning systems do not have a student model that allows them to provide one-to-one learning experience without the intervention of human teachers. Furthermore, system's support like personalised feedback and hinting is not provided by most of E-learning systems. However, Intelligent Tutoring Systems (ITSs) provide

all these features in their underlining structure that makes them a better choice for the proposed system in this research.

2.4   Web-based intelligent tutoring systems

Research in ITSs has shown a great interest in Web-based education which has given rise to more Web-Based Intelligent Tutoring Systems (WBITSs) on the Internet (Butz *et al.*, 2008).

WBITSs have the advantage of providing access to the content anytime anywhere. WBITSs provide self-paced instruction for learners who want to move ahead or learners that want extra practice. Moreover, WBITSs engage learners with stimulating content and interactivity, and they offer on-demand access to help and support. In standalone systems, the learner needs to use the same machine if he/she wants to benefit from their previous interaction with the tutoring system (Brusilovsky and Tasso, 2004; Roselli *et al.*, 2008).

Before the start of the new millennium, few ITSs were developed as web-based systems or used learner modelling in a distributed style (Nejdl *et al.*, 2008). However, WBITSs are still widely misunderstood and often associated with a restricted view of e-learning. WBITSs started to appear at the beginning of the 1990's with the emergence of the Internet as a learning tool. With the success of the early models, new systems continued to emerge such as an ITS for a Virtual E-learning Center (Contreras *et al.*, 2006), and Web-based Authoring System (Roselli *et al.*, 2008).

WBITSs require a high degree of adaptiveness to address and serve the vast diversity of the online learners. These learners have differences in their languages, prior education level, skills level and dedication to the learning process. Thus, the approach of this research emphasizes a general structure that is not limited to a certain language, background or group.

2.5   Intelligent Tutoring Systems' support

An ITS can be used to support a variety of tasks to enhance the learning process. Simpson (2002) divides these tasks into two broad areas: *academic* and *non-academic*. The academic task is concerned with the learner cognitive and intellectual states, while the non-academic task provides counselling and support for learners. System support is critical to the learning process since it helps to provide rich context for meaningful instruction and reducing the dropout rate. The support provided by the system requires it to adopt several unique roles depending on the tasks the learner has to manage. These roles include resource providing, advising, moderating, facilitating, guiding and assessing the learner in order to gain knowledge and understanding (Denis *et al.*, 2004). Providing support for learners is a popular way of guiding learners in most ITSs. This support can take different forms such as feedback and hints. Tutoring system support is a way to improve the learning process through continuous assessment of learning results and the quality of learner performance (Guo *et al.,* 2008).

In general, there are two possibilities concerning delivering feedback: an immediate feedback after each step or action in problem-solving and late feedback after submission of a complete solution. In the late feedback a learner chooses the moments of feedback presentation and the system provides feedback about correctness of their previous actions. The analysis of previous research has shown that an ITS typically gives an immediate feedback after each performed action or step during problem-solving, and whether the action or the step was correct or incorrect. A strategy prevents the learner from proceeding along a wrong solution path (Anohina, 2007; Murphy, 2007).

Hints take the form of clue or tip (presented in an indirect fashion) to help learner in problem-solving and can be considered as a form of feedback. Matsuda and VanLehn (2003) conclude that ITSs have relatively simple and inflexible hinting policies that are based on a prescribed problem-solving strategy. ITSs can have more than one layer of hints usually organized from the most general to the most specific. The general hint usually contains the minimum information. ITSs typically

provide a special button or tool, which the learner can use to request hints. In Andes (VanLehn *et al.*, 2005) two functions are provided: one for providing help on incorrect entries, while the other providing hints on the next problem-solving step. Many ITSs are based on a problem solving strategy that require learners to work through problems step-by-step while the ITS provides hints and feedback, to improve the learning process. Several studies have shown this strategy to be effective in helping learners to learn (Crowley and Medvedeva, 2006; Razzaq and Heffernan, 2009).

2.6   Natural languages tutoring systems

Computer-aided language learning (CALL) is a form of computer-based learning that is used to facilitate the language learning process (e.g. Arabic language). The field of CALL has passed through two phases, which can be identified as: (i) the pre-Internet phase that include the workstation-mainframe period (the seventies),(ii) the microcomputer period (the eighties), (iii) the post-Internet phase which include the multimedia period (the nineties), and (iv) the Web-based intelligent tutoring systems period (the late nineties).

Intelligent computer-aided language learning (ICALL) is the study of using Artificial Intelligence (AI) techniques in CALLs and language learning (Levy and Stockwell, 2006). AI research and techniques like Learner Modelling (LM) and Natural Language Processing (NLP) have been a source of contribution to the design and implementation of CALLs and have played a significant role in the development of the systems (Godwin-Jones, 2007). ICALL and ILTS systems provide ideal pedagogy for teaching languages and can be used to assist learners in the development of the language awareness. Without intelligence CALLs are simply other systems that provide language teaching to learners.

Gamper and Knapp (2002) presented a survey of more than forty ICALL systems in which they concluded that most of the systems that use NLP techniques only concentrate on syntax and few contained semantic components. In addition, the application of AI in ICALL systems is not mature yet and additional research efforts are still required. Jia (2004) presented a web-based instruction system for foreign

language learning. The system was based on human-computer dialogue with natural language for the teaching of a foreign language. It described the grammar elements of any expression in a natural language and it can be used in information retrieval. The system relied on text input from the user with no speech recognition capability attached to it.

Some ICALLs use NLP to interpret user input as well as to generate responses. E-Tutor (German Tutor) was a comprehensive language learning environment for German developed by Heift (2004) based on NLP. The system generated error feedback suited to learner expertise. Additionally, E-Tutor provided a "Report Manager" which allows learners to review the work they have done and to redo certain exercises. BANZAI/Robo-Sensei (Ushida, 2006) is interactive software for teaching the grammar principles that are essential to the mastery of the Japanese language, using NLP with focus on sentence production practice. A learner creates his/her own sentences in Japanese in response to real-life situations that include extensive cultural information. The designs of E-Tutor and E-Tutor were based on modular structures that enable re-usability of systems' parts. This design was done by separating learner input into separate programs (modules) that can be run separately. TAGARELA is an adaptive system for individualised instruction of Portuguese developed at Ohio State University and motivated by providing grammar lessons outside of class rooms. The system was based on NLP modules to process input and to decide on the best processing strategy. The system provided intelligent feedback for any type of error (Amaral, 2007).

2.7   Arabic language tutoring systems

Despite recent interest in the Arabic language, few ICALLs and ILTSs have been developed for teaching Arabic language (Shaalan, 2005). ArabVISL is interactive software for self-paced learning of Arabic grammar on the Internet (Nielsen and Carlsen, 2003). The system allows the learners to segment and label an Arabic sentence by using grammatical terminology. The learner chooses an Arabic sentence from pre-analysed corpus and selects the visualisation mode. An Arabic authoring tool was developed at London Guildhall University for teaching various

languages including Arabic (Cushion and Hemard, 2003) based on different teaching activities such as "listen and fill the gaps" and "listen and repeat". The authoring tool allows language teachers to edit their own audio-enhanced websites containing a variety of interactive exercises.

Bush and Browne (2004) cited a meta-analysis that demonstrated how online instruction can be effective for language learning and specific instances in the Arabic language. The work showed how materials can be developed in a cost-effective manner. The authors also discussed the Shareable Content Object Reference Model (SCORM) specification which provides cross-platform delivery via technologies that are Web compatible. By implementing the SCORM specification, any learning materials can run under any Learning Management System (LMS) or Course Management System (CMS) that has implemented the SCORM specification.

Although the proposed system shares some of the characteristics of SCORM such as accessibility and the reusability of the system's templates; however it is developed for natural language teaching based on the concept of a pluggable domain and a unique design of the DK which cannot be reused by other systems. The focus in SCORM is on a sequencing model that shows all or part of the learning contents. The learner model is replaced by the Activity State and Tracking Models that allow the learner' traversal of the learning content to be evaluated (SCORM 2004, n.d.). On the other hand the learning process in the proposed system is based on multiple instructional strategies which include sequencing, pre-requisites and extra tutoring. This is supported by the evaluation of various learner's characteristics through different components such Fuzzy inference and reconfigurable domain-based instructional strategies. The proposed system can also provide intelligent feedback, hints and explanations based on the learner's characteristics all of which issues are not been discussed by SCORM.

In order for digital courses to be interchanged, some metadata standards (like SCORM) have to be generated. Yet this is a complex task, where only partial successes have been reported (Samuelis, 2007) and best practices for doing this are still to be developed (SCORM 2004, n.d.). Although some similarities can be noticed between ITSs and the SCORM specification, there are some differences between

them. In SCORM the learner interaction and the learning path are stored in a text file named the "manifest file". In ITSs these components are embedded within the system and the content of a particular ITS cannot be reused nor interoperate in another ITS (Chew, 2008).

Shaalan (Shaalan, 2005; Farghaly and Shaalan, 2009) developed an ICALL system for learning Arabic based on NLP. The system provided feedback to the learner based on syntax analysis of the learner error and proposed a mechanism of correction by the learner. The Tactical Language Training System (TLTS) is a computer learning system based on speech recognition designed to teach Arabic spoken communication to American English speakers (Johnson and Valente, 2008).

The development of ICALL systems is very expensive; a reason why only few ICALL systems have been used in real applications with the focus on a limited range of languages. In addition, techniques like NLP, automatic translation, speech recognition, and parsing still present technological problems in terms of performance, implementation, and complexity. Considered over-promised and underachieved, developers of these systems have mostly narrowed their aims and goals which resulted in deploying systems rather than research prototypes (Godwin-Jones, 2009).

Although numerous technological innovations have made the process of building Arabic ICALL materials simpler, there is no clear standard for an authoring system that will facilitate the development of an Arabic language system. Also all of Arabic CALLs and ICALLs are limited in nature and generally suffer from serious problems in design and maintenance. In addition, most of the learner models developed for these systems have largely focused on the acquisition of grammatical structures with little attention given to the personalisation process (Cristea, 2005). Arabic NLP in general is still underdeveloped and tools used for other languages are not easily adaptable to Arabic due to the language complexity (Abdel Monem *et al.*, 2008). Furthermore in the presented ICALL systems, the author can only change the contents of the learning materials but cannot change the way the learning materials are taught.

	German Tutor (Heift, 2004)	Robo-Sensei (Ushida, 2006)	TAGARELA (Amaral, 2007)	TLTS (Johnson and Valente, 2008)	Arabic ICALL (Shaalan, 2005; Farghaly and Shaalan, 2009)
Techniques	ICALL	ICALL	ILTS	ILTS	ICALL
Intelligent components	NLP	NLP	NLP	Speech recognition/ Pedagogical agent	NLP / Rule based
Availability	Web based	Web based	Web based	Stand alone	Stand alone
Learner model	Diagnosis model	Not provided	Diagnosis model	Diagnosis model	Not provided
Teaching strategy (instruction model)	No instructor model : Flow of instruction purely based on learner choice	No instructor model : Flow of instruction purely based on learner choice	Task-based approach	Task-based approach / scaffolding	Curriculum sequencing
Feedback	Error feedback	Error feedback	Error feedback	Corrective feedback	Error feedback
Hints	Not provided	Not provided	Not provided	Not provided	Not provided
Explanation	Not provided	Not provided	Not provided	Not provided	Not provided
Language skills	G	R G L	R G L	G S	G
Type of interaction with learner	Supply-type / selection-type.	Supply-type / selection-type.	Supply-type / selection-type.	Supply-type	Supply-type / selection-type.

Language skills key: R: reading; G: grammar; L: listening; S: speech

Table 2.1: Features of some of computer-based language teaching tools

Table 2.1 presents various components of ITSs and ICALLs that have considerable advances over previous systems. A shortcoming of the presented systems is that they cannot be easily altered and applied to another language. Some systems like the German tutor and TAGARELA promote (and encourage) the reusability of their components. This is done by separating the processing of the system input into separate modules (e.g. Grammar modules or Analysis modules) which can be run sequentially. This modular design allows for the teaching of new languages by replacing certain modules of the system by new modules according to the target language. This approach has allowed for the creation of an ICALL system for

learners of the Russian language (Dickinson and Herring, 2008) by reusing many significant components of TAGARELA. However these modular systems are built with the option of sharing only particular modules since the analysis processes and the grammar models vary according to the target language. Moreover these systems are not optimised for performance and the cost of building these systems is usually very high (Godwin-Jones, 2009).

In the proposed work the design of the system allows for the teaching of a new language using the current system's templates (e.g. lessons and questions). That is without replacing these templates the tutoring system can be used to teach a new language (e.g. French) based on the concept of a pluggable domain. A major effort was made to build a general Domain Knowledge that can be invoked by the teacher (i.e. author) to represent the new language domain. This has the advantage of reducing the time and effort needed for the implementation of a new language tutoring system.

2.8   Learner modelling

The term "Learner model" is used in different ways by different researchers. Researchers have been using various synonyms for the learner model such as user model, mental model, cognitive model, conceptual model, and user profile. In the last decade, researchers have explored many different ways to develop effective instructional systems that tailor interactions and learning activities to an individual learner. Researchers have used different techniques to accomplish this task including those from cognitive psychology, instructional science and artificial intelligence.

The learner model represents the system's beliefs about the learner's knowledge and skills, and it guides their pedagogical decision-making. The learner model is updated regularly by data collected from several sources implicitly, by observing learner activities or explicitly by requesting information directly from the learner (Esposito *et al.*, 2004). Adaptive tutoring systems can modify the learning process to best fit learner's needs, characteristics and preferences by discovering the strengths and weaknesses of the learner. The effectiveness of an adaptive tutoring system

depends on how accurate the learner model is (Carmona *et al.*, 2007). Although the learner model is directly related to the domain knowledge model, the design of the instructional strategy usually determines the learner modelling technique.

Traditionally, ITSs' researchers have focused mostly on modelling the learner's cognitive processes during problems solving as in the "model tracing" approach which models the way people think in a teaching environment. Some researchers developed cognitive diagnosis while others employed adaptive remedy based on the learner's current state and the learning difficulties encountered (Mills and Dalgarno, 2007; Dubois *et al.*, 2008). Moreover there is a recent trend in ITS research to involve more than cognition and focus on issues like the learner's transient shifts in attention, emotions, help-seeking and misuse of the system's help known as "Gaming" (Baker *et al.*, 2005; Qu and Johnson, 2005).

The learner model acquires, maintains and stores data that is specific to each learner and accumulates information about the learning history in order to assess the learner's knowledge of the subject or what he already knows. Learner models have been included in several systems to improve the performance and the level of interaction with the system. Much of the success of ITSs is accomplished by using AI methods to represent the learner behaviour and pedagogical judgment of the ITSs (Shin and Chan, 2004). On the basis of these variations the system decides what curriculum should be incorporated and how it should be presented **(**Cristea, 2005; Kazi *et al.*, 2007; El-Barouki, 2008).

Classically, the learner model consists of two components: *Overlay model* and *Bug model*. The Overlay model considers the learner's knowledge as a subset of the expert's knowledge, while the bug model represents the learner's misconceptions based on the triggering of incorrect rules. However, strategies like Constraint-based, Case-based and Stereotype learner modelling have been successfully proven in various ITSs. These strategies are further discussed in the next sections.

2.8.1 Constraint-Based Modelling

Constraint-Based Modelling (CBM) is a learner modelling technique in which the knowledge state of the learner is represented by constraints on correct solutions paths (Mitrovic *et al.*, 2007). When the learner violates a constraint, the tutor gains specific information about the learner. CBM has the advantage of simplifying the building of the domain and the learner models in ITSs. However, CBM is suitable only for domains in which the solution itself is rich in information. Also CBM does not teach the learner to employ good strategies but only to keep out of incorrect states. A number of successful constraint-based tutors have been developed over the years in various domains, such as SQL-Tutor for teaching SQL database language (Weerasinghe and Mitrovic, 2006; Jeremic *et al.*, 2009).

2.8.2 Case-Based Modelling

Case-Based Modelling (CSBM) is another learner modelling technique which solves new problems by using or adapting solutions similar to the learning domain of a past learner history. A case-based reasoning approach to Adaptive Web-based Educational Systems using fuzzy logic is presented by Alves *et al.* (2008). The system adapts its contents according to the learner learning style and individual needs. Rishi and Govil (2008) presented the design of an agent-based distributed Case-Based ITS for online learning. However, complex cases require huge time to design and large quantity of resources since the quality of the system depends on the number of well-defined stored cases.

2.8.3 Overlay model

The overlay model is a learner modelling approach which represents the Learner's knowledge as a subset of the domain expert's knowledge by estimating the mastery of each component in the domain that an expert would be expected to know. SQL-Tutor (Mitrovic, 2004) uses an overlay model for modelling the learner's long-term knowledge. An integrated learning environment has been developed by (Galeev *et*

al., 2006). This environment includes electronic textbook and integrated tutoring system. The adaptation of the system to the learner is based on the overlay model.

2.8.4 Stereotype

A stereotype is a mental device which helps in categorising people based on certain measures. People habitually invoke a stereotype as a basis for expectations about interactions with other people. The system selects the stereotype based on the learner's initial interaction with the system or by assigning a default stereotype to the learner according to his/her actions and slowly replaced it by more individualised stereotype (Chrysafiadi and Virvou, 2008). The initialisation of learner models is based on default assumptions, which are attached to each class or stereotype (Virvou and Tsiriga, 2004). The common approach in stereotyping involves classifying learners into different classes (e.g. novice, advance) based on their knowledge, and as a result of learning, it is usually assumed that the learner makes a progress from a less advanced level to a higher one. Moreover, the learner may, may not learn, or may forget what they already learned; therefore, it is very important to model the transitions from one cognitive state to another. Haake and Gulz, (2008) explore the use of visual stereotypes in virtual pedagogical agents in digital learning environments and the potential impact of such use in education with respect to teaching and learning.

Model tracing, CSBM and CBM are short term learner modelling approaches. In CSBM and CBM only declarative knowledge is represented while in model tracing both procedural and declarative knowledge is represented. Stereotypes and overlays are long-term learner modelling approaches but the two are different in the amount of information they represent. The learner model in the proposed system is based on the stereotype learner model. This enables the system to make a large number of inferences on the basis of a substantially smaller number of observations.

In general learner modelling has many problems like the difficulty in building a learner model, determining what information should be represented in it, and how to use it. Therefore, it is better to design a general learner model that is based on set of

features that can be used in a variety of applications. Various stereotype learner models are based on an abstract classification of frequently occurring characteristics of learners; however, most of these systems move the learner from one stereotype to another, without changing the stereotypes themselves. In the proposed work the design of the tutoring system allows for the modification of its learner model via user-friendly authoring interfaces. Since there is no definitive answer to what the system design should optimally look like, it seems rational to give the teacher some choices in the system design matter (i.e. the ability for non-technical teachers to change the configuration of the system). Finally the design of the learner model is language-independent that can be applied to any language other than Arabic.

2.9   Fuzzy Intelligent tutoring systems

Fuzzy logic has been used in diverse ITSs as decisions are made in a similar way to human teachers. Without complex formulae it utilises a set of rules similar to those a teacher would apply in judging learners' performance or activities. In addition, Fuzzy logic provides flexibility when used to implement mathematical formalisations based on natural language or working with imprecise information (Stathacopoulou *et al.,* 2007; Sanchez-torrubia, 2008).

Nedic *et al.* (2002) designed a fuzzy rule-based decision making system aimed at adaptively adjusting the teaching of a first year engineering course on electrical circuit theory, based on learners' performance. Each learner's performance was based on the membership functions for a particular topic, difficulty and importance levels. A "virtual learner" model which simulated human learning behaviour was developed by Negoita and Pritchard (2004) based on fuzzy logic technologies.

A neural network-based fuzzy model for learner's knowledge representation was presented by Stathacopoulou *et al.* (2004). The model was used to assess the learner's motivational state during the learning process based on his/her observable behaviour and motivational factors. The training of the neuro-fuzzy model and updates of the learner model were based on teachers' experience. Their work showed that teachers' knowledge can be elicited in linguistic form and encoded in

27

the system. The system had the ability to simulate learning as well as forgetting what has been learnt. Even though experimental results showed that the concept of "virtual" learner to be a promising idea, in the proposed system the virtual learner model is not exploited with the exception of using the fuzzy logic technologies. Stathacopoulou *et al.* (2005) proposed a neuro-fuzzy model to encode teachers' knowledge. This was applied to diagnose learners' learning characteristics. The experimental results from testing the model in a learning environment were encouraging, showing good capability of handling uncertainty as confirmed by the advice of five experienced teachers. Chen *et al.* (2006) presented a learning performance assessment scheme by combining a neuro-fuzzy classifier and a fuzzy inference. The inferred learning performance results can be used as a reference for teachers, and provide feedback for learners.

In this work multiple fuzzy inferences are used to represent the learner model (i.e. Fuzzy Stereotyping) and manipulate various learner attributes such as Performance and Engagement. This allows the system to use these fuzzy inferences separately or in conjunction with each other. The design of the fuzzy inferences is based on flexible design and this makes it adaptable to different learning conditions. The assumption is that the visualization of the system design will encourage the teacher to reflect on his/her knowledge.

2.10   Help-seeking and gaming

Help-seeking is a process done by learner. It is aimed at getting help from the tutoring system in order to solve certain problem. Nevertheless, it can be done ineffectively in ways that are not beneficial to their learning process. Although several studies have investigated the effectiveness of Help-seeking in computer-based learning and have proposed new solutions, yet they introduce new problems (Baker *et al.,* 2006). For example, help was not permitted without first attempting to answer the question which led learners to answer randomly in order to get help and 'game' the system (Baker*, 2007; Mavrikis, 2008).*

Learner gaming is behaviour aimed at obtaining correct answers and advancing within the curriculum quickly and easily with little dedication to the educational content and has been correlated with poor learning (Baker *et al.*, 2004). A learner typically games the tutoring system in order to obtain correct answers easily and advance within the tutoring system with little effort and dedication to the learning process. A classifier model that can identify if a learner is 'Gaming' the system was introduced by Baker *et al.* (2004). Results of experiments from three data sources showed successful recognition by the model of learners who 'Gamed' the system. Another system was introduced by Baker *et al.* (2006) which gave gaming learners additional exercises focused on exactly the materials learners avoided through gaming. The finding showed that this technique has been effective in large parts as it offered additional learning support (Walonoski and Heffernan, 2006; Baker, 2007).

Comparable results from designing preventative strategies are described by Murray and VanLehn (2005). It is suggested that in order to develop a model with beneficial interaction, it is better to provide help to the learner based on the quality of his/her previous interactions with the system (Van de Sande and Leinhardt, 2007; Mavrikis, 2008). Different approaches towards remediation of this behaviour (i.e. gaming and misuse of system help) were proposed; however, these approaches focused on active intervention of the system which resulted in unreasonable penalization of non-gaming learners given that different studies have shown that only a minority of learners game the system.

2.11   Intelligent tutoring systems as authoring tools

Authoring tools can simplify the development of ITSs especially for general groups (including non-programmers) by allowing them to create (i.e. author) contents deliverable to end users of ITSs. An analysis of the research and development of ITS authoring systems provided by Murray *et al.* (2003) in which a categorisation of more than 24 authoring systems was given, followed by a characterisation of types of ITSs that are built for each category. Murray *et al.* (2003) concluded that the research in ITSs authoring tools has demonstrated significant success in limited cases. Murray

argues that it lacks reusability and modularisation with fixed instructions that cannot be individualised for each learner.

Macromedia/Adobe Authorware is a commercial authoring tool for building interactive instructional material that provides good support for multimedia content. The idea of authoring by example is one approach to authoring. Instead of encoding domain expertise and tutoring knowledge, the author demonstrates ideal solutions. The author identifies what the tutoring system should say to the learner at different points of the demonstration (Aleven *et al.*, 2006a).

Various authoring tool such as CTAT (Aleven *et. al*, 2006a), and ASSISTment (Turner *et al.*, 2005) lower the expertise necessary to create a tutoring system by providing teaching methods, various scenarios for learners along with a specification of how the learner's actions and knowledge will be evaluated and assessed. However, these tools are also limited in the scope by which they can be personalised. While other tools such as REDEEM (Ainsworth and Fleming, 2005) represent multiple tutoring strategies that select the appropriate tutoring strategy for a given situation. Pedagogical parameters such as the depth of hints and pre-requisites can be easily authored. Eon system (Murray *et al.*, 2003) is an authoring tool for ITSs that contains a fully integrated set of functions for all aspects of ITS design. Eon provides meta-strategies that combine the authoring of meta-strategy triggers with parameterization values. Since Eon does not include rule-based representation of the expertise therefore it is not suited for representing complex problem solving skills.

Different authoring tools are developed for Cognitive Tutor. One example of these tools is the Cognitive Tutor Authoring Tools (Aleven *et al.*, 2006a) which provides supports for two types of tutors: Cognitive Tutors and Example-Tracing Tutors. Another example is the Cognitive Model SDK developed by Blessing and Gilbert (2008) to allow non-cognitive scientists with no programming experience to produce a cognitive model for model-tracing tutors. However, these tools focused on making cognitive tutor development easy and fast with little attention given to the intelligent components of the systems.

Many authoring tools have been developed with the purpose of reducing the time and cost required to build them. Nonetheless, most of them only permit a single solution path and are not practical in building large ITSs where learners are presented with a varied number of problems (Murray *et al.*, 2003). In addition the learner models of most of the ITS authoring tools are based on learner's performance "Overlay model" yet other learner's characteristics are overlooked as well as most of these systems have not incorporated nor provided authoring for AI modelling techniques such as Fuzzy logic (Koedinger *et al.*, 2004; Ainsworth, 2007). Finally, authoring human languages has not been developed as authoring the natural language is more difficult than other areas of knowledge like physics and programming languages.

2.12   Proposed system vs. Existing solutions

The advent of the Web has made it the preferred platform for delivery of the learning materials. Web-based tutoring systems have the advantages of providing self-paced instructions for learners based on any-time, anywhere and on-demand learning. E-learning delivers web-based educational content to online learners and it is widely used by higher educational institution. However most of E-learning systems are based on static web-based tutoring systems and do not provide one-to-one and intelligent interaction with the students. Moreover these systems lack the personalised system's support such as feedback and hinting.

ITSs have advantages over other techniques of language tutoring systems (i.e. CALL, ICALL, and NLP-based tutoring systems). ITSs can provide personalised and instructions that meet the needs of each individual learner. ITSs can provide intelligent and individualised feedback and hinting to each student. This is done via various components such as the student and the teacher models.

The current language learning systems for teaching Arabic have many weaknesses and limitations. Most of these systems lack the adaptability and intelligence required to ensure effective learning. Thus, a new ITS for the teaching of Arabic is needed.

In order to provide significant learning experiences for the diverse group of online learners different modes of teaching should be presented by the tutoring systems. ITSs designed with a particular class of learners in mind may not suit other learners. Therefore, a flexible or generic design that can be fine-tuned by a teacher who has no programming experience is preferable. Fuzzy logic has been successfully used in ITS as it handles uncertainty and offers mode of qualitative reasoning closer to the teacher's decision making process. It can also be easily modified to improve the learning outcomes. Hence incorporating fuzzy inference for estimating the student learning experience is preferred.

ILTSs are difficult and expensive to build, hence, it is desirable to build an ITS that can be easily adapted to the learning and teaching of different languages since there is no authoring tools or shells yet developed for natural language teaching. This has the advantages of reducing the time and cost required to build these systems.

2.13   Summary

This chapter reviewed the previous work related to this research on ITSs for language learning. Instructional strategies and the advantages of implementing these strategies via web technologies are discussed. System's support like feedback and hints are presented and previous techniques used are discussed. Current ILTSs and ICALLs are investigated and techniques used by these systems are discussed as well as the limitation of the current research is highlighted. Various learner modelling research and techniques are discussed and several example of the current research are presented. Finally several researches in Fuzzy-based ITS authoring tools are investigated and the limitations of these systems are highlighted. The following chapter will discusses the overall design of the proposed ITS in terms of mode of interaction, instructional strategies, the concept of the pluggable domain and the curriculum design.

CHAPTER 3: DESIGN OF THE PROPOSED INTELLIGENT TUTORING SYSTEM

This chapter introduces the overall architecture of the tutoring system. This includes modes of interaction, Instructional strategies, and the concept of Pluggable domain that forms the main parts of the whole system.

3.1   System design

The tutoring system is a prototype Web-enabled ITS for teaching of Arabic language. English is used as a medium of instruction or supporting language during the learning process. The system starts with the basic alphabet and vocabulary then moves to more difficult subjects like adjectives and sentence building in the target language (e.g. Arabic language). The system incorporates multimedia features into the teaching process. The system offers integration of foreign languages' skills such as grammar, reading and listening which enhance the learning process. Each learning unit is associated with a corresponding script, audio and one or more images along with an equivalent video to represent certain language constructs. Several tools are used by the tutoring system. These tools perform different tasks like: displaying contents or files, sending corrections, playing audio or videos, etc. These tools simply execute the learner's commands, which involve one or two instructions written in source code that invokes a specific service from the tutoring system.

In addition to the earlier described functions, the system offers a series of functionalities. These functions do not exist separately but each contributes in a unique way to the overall performance of the system. These functions are described as follows:

a. Monitoring

Monitoring gives advice on demand to the learner, and provides results, reports, explanation, and feedback. The advice is content related and relevant to the learner's situation.

b. Mentoring

Mentoring tracks the learning process without the learner's request. Mentoring provides reminders and analysis to the learner based on information about the learner's situation and previous behaviour. Different routines for diagnosis, data retrieval, and answer analysis are triggered on the basis of this information.

c. Tutoring

In offering the relevant instructional strategies, the tutoring system relies on more complex routines as discussed in more details later in this chapter.

There are two types of users in the tutoring system: learner and teacher. The teacher or domain expert (subject matter expert) is a user with special knowledge or skills in a particular area of endeavour, knowledge been taught by the tutoring system (foreign language teaching). He/she also can use, modify and fine-tune the system. The learner is a user who interacts with the system in order to learn certain language. Each user (i.e. learner or teacher) connects to the tutoring system by means of a login process, which is necessary in order to establish a session.

After the authentication process the user may have access to the tutoring system through a user interface. Two user interfaces are provided by the system: one for learners and another for teachers. The learner-interface facilitates the learning process to the learner while the teacher-interface provides more features to the teacher through which they can fine-tune and control the learning process provided by the tutoring system.

3.2   Mode of interaction

The proposed tutoring system interactions with the learners are presented in two modes: interactive and discovery. In each learning session, all events (i.e. login time,

questions solution and requesting feedback) that happen during each learner interaction with system are recorded in a log file specific to each learner.

3.2.1 Interactive-based learning

This mode encourages the learner to focus on the concepts they do not know based on the selected instructional strategy and the learning goal. To achieve this goal, different teaching units (*TUs*) are introduced to the learner when presenting him/her with new lesson. Based on the placement test result, the tutoring system gives the learner the option to select the appropriate initial learning level (e.g. Beginner, Intermediate or Advanced).

In each learning session, a Lesson model is responsible for selecting the next lesson and the learner is required to study certain *TUs* (practice, extra tutoring or prerequisites) based on the selected instructional strategy. After studying all the required lesson's components (i.e. required by the instructional strategy selected by the tutoring system), the learner must answer a set of questions related to the studied lesson. The tutoring system provides on-demand advice (feedback and hints) while answering questions. Figure 3.1 presents a course map which shows the course progresses from start to finish. The system uses a minimum threshold for the time spent on the required *TU* below which the tutor would not allow the learner to move to the next *TU* (Baker, 2007).
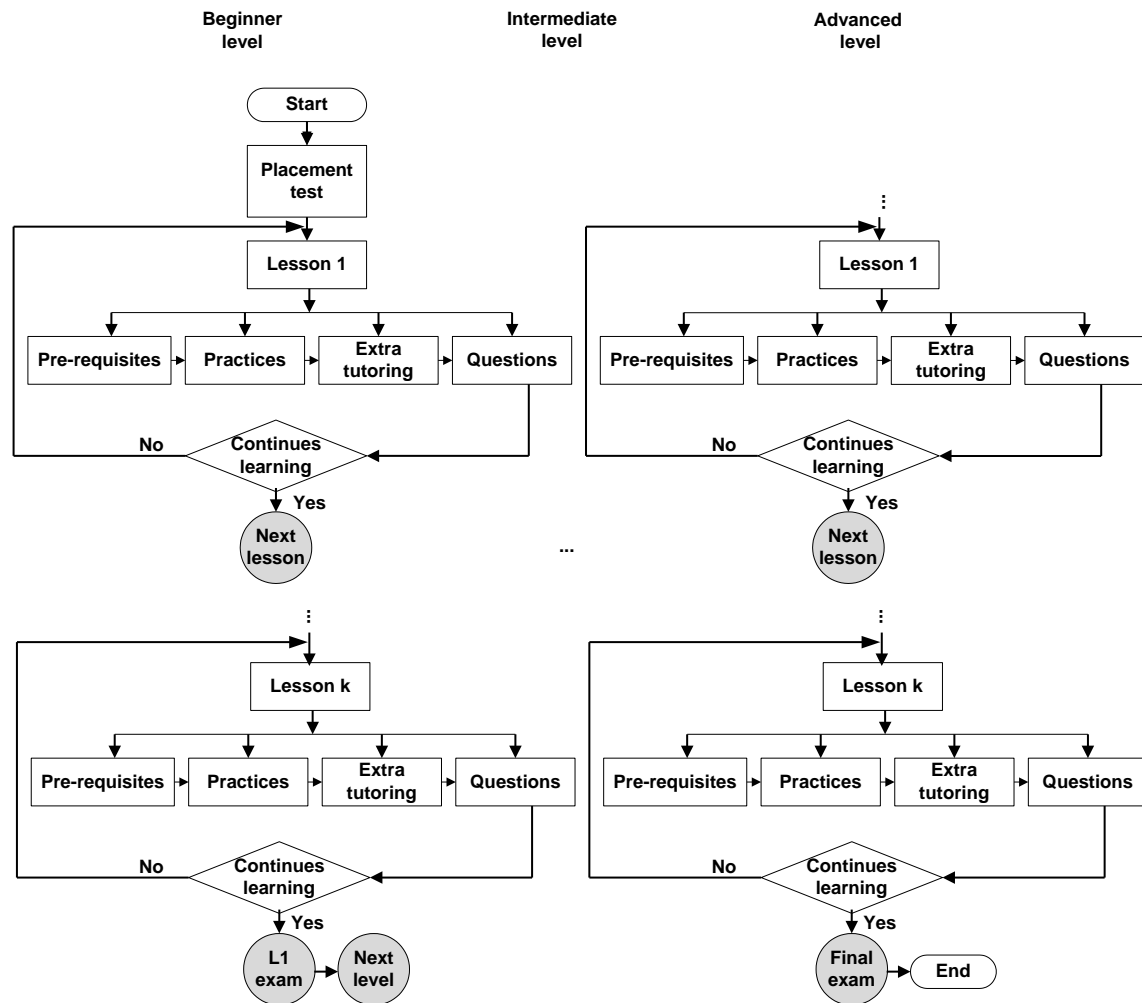
Figure 3.1: The course map

3.2.2 Discovery based learning

Discovery based learning is 'learning by doing' in which the learner interacts with his/her environment by exploring and manipulating the course's teaching materials (Kirschner *et al.*, 2006). The tutoring system provides discovery based interaction for the teaching materials that previously studied by the learner based on the argument that learner should be given some control of the learning process and there should be some forms of guided discovery in which the learner is allowed to make decisions about the learning process with some guidance from the system (Anderson, 2007). Offering learners options encourage them to become more independent (Godwin-Jones, 2009).

3.3   Instructional strategies design

The teaching of the learning materials is based on multiple instructional strategies which can be interactively adjusted by the teacher via the teacher-interface. The tutoring system selects the proper instructional strategy for teaching a certain lesson/question based on its attributes (see Figure 3.2). The selection process will be discussed later on in this thesis. The attributes set includes pedagogical parameters from learning materials. These attributes are stored in "StratFile" file. The complete list of these attributes is shown in Tables 3.1. The structure and the initial values of the "StratFile" file is shown in Table 3.2.

Attributes	Values		
	1st value=0	2nd value=1	3rd value=2
Learning level	Beginner	intermediate	advanced
Difficulty	Easy	Medium	hard
Time	Short (< 4 minutes)	Full (>= 4 minutes)	
Category	Descriptive	Procedural	
Type	Lesson	Question	

Table 3.1: instructional strategy attributes and their values

Instructional strategies	Learning level	Difficulty	Time	Category
1	Advanced	Easy/Medium/hard	Short/ Full	Descriptive/ Procedural
2	Intermediate	Easy/Medium/hard	Short/ Full	Descriptive
3	Intermediate	Easy/Medium/hard	Short/ Full	Procedural
4	Beginner	Easy/Medium/hard	Short/ Full	Descriptive/ Procedural
5	Intermediate/ Advanced	Medium/hard	Short/ Full	Descriptive/ Procedural
6	Intermediate/Advanced	Easy	Full	Descriptive/ Procedural
7	Intermediate/ Advanced	Easy	Short	Descriptive/ Procedural
8	Beginner	Easy/Medium/hard	Short/ Full	Descriptive/ Procedural

Table 3.2: The structure and the initial values of the "StratFile" file
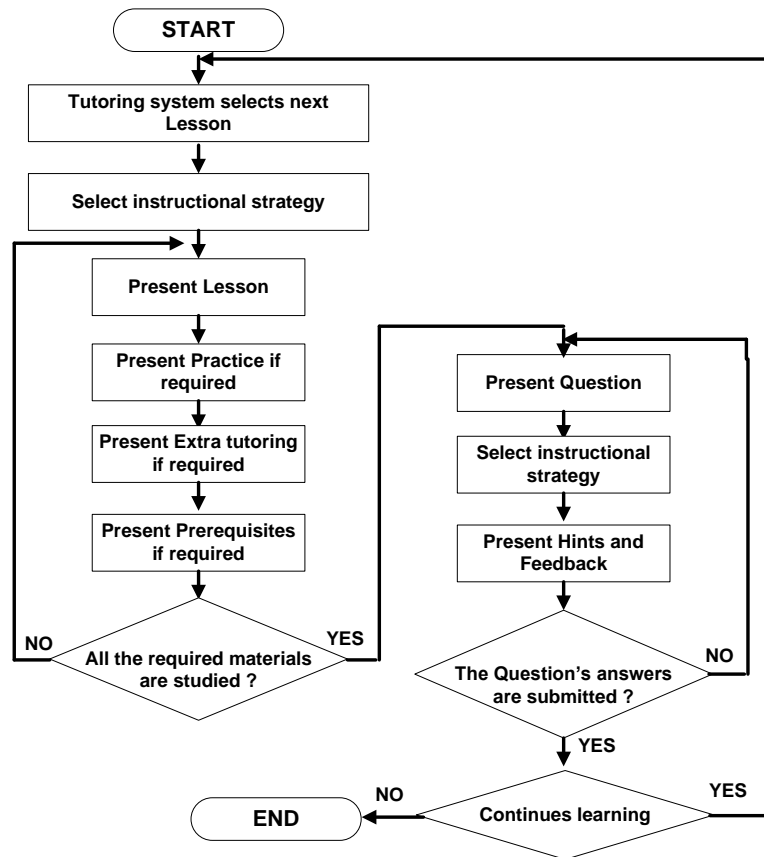
Figure 3.2: The design of the selection process of the instructional strategy

Based on the design of the instructional strategies, each combination of these attributes is mapped into a certain instructional strategy. Two groups of different instructional strategies are developed. Each group contains different strategies and each strategy is pre-defined based on the combination of the input's attributes. The first group is concerned with providing the proper instructional strategies for lessons (see Table 3.3). The second group is concerned with providing the proper instructional strategies for questions (see Table 3.4). Three features (activities) which control the actions of each instructional strategy are embedded in the design of each group. For the first group, strategies 1 to 4, these features are described as follows:

1. Practice. The process of the learner going through (studying) the content of the lesson's practice.
2. Prerequisites. The process of learner going through the content of all the lesson's prerequisites. Pre-requisites are different lessons that precede the given lesson.

38

3. Extra tutoring. The process of the learner going through the content of the lesson's extra tutoring. Extra tutoring represents additional teaching materials provided by the system.

	Strategies	Practice	Prerequisites	Extra tutoring
1	Strategy 1	2	1	0
2	Strategy 2	2	2	2
3	Strategy 3	1	2	2
4	Strategy 4	1	1	2

Table 3.3: Lessons strategies and their features

Table 3.3 Key

Features	Values	Description
Practice	0	Learner not required to study the Lesson.
	1	Learner must study the Lesson's practice.
	2	Learner study the lessons' practice according to a strategy carried out by set of rules which have the following general form: **IF** Performance* **AND** Engagement* **THEN** Action (Mandatory **OR** Optional) **ENDIF**
Prerequisites	0	Learner not required to study the Prerequisites.
	1	Learner must study all the Prerequisites.
	2	Learner study the Pre-requisites according to strategy carried out by set of rules which have the following general form: **IF** Performance* **THEN** Action (Mandatory **OR** Optional) **ENDIF**
Extra tutoring	0	Learner not required to study the Extra tutoring.
	1	Learner must study the Extra tutoring.
	2	Apply extra tutoring to the learner according to strategy carried out by set of rules which have the following general form: **IF** Performance* **AND** Help-seeking* **THEN** Action (Mandatory **OR** Optional) **ENDIF**

* Performance, Help-seeking and Engagement are learner's stereotype.

For the second group, strategies 5 to 8, the features are described as follows:

1. Hint. A short statement in a form of clue or tip presented to the learner upon his/her request.

2. Feedback. One statement or more presented to the learner upon his/her request.

3. Time. The time dedicated to each question.

	Strategies	Time	Feedback	Hints
1	Strategy 5	1	1	1
2	Strategy 6	0	1	0
3	Strategy 7	0	1	1
4	Strategy 8	0	0	1

Table 3.4: The Questions strategies and their features

Table 3.4 Key

Features	Values	Description
Time	0	Learner has unlimited time to answer the given question.
	1	Learner must answer the given question within a given time carried out by set of rules which have the following general form: **IF** Engagement* **THEN** Action (answer within question time **OR** unlimited time to answer) **ENDIF**
Feedback	0	The system provides feedback to the learner.
	1	The system provides Feedback to the learner according to strategy carried out by set of rules which have the following general form: **IF** Performance* **and** Help-seeking* **and** Engagement* **THEN** Hint type **ENDIF**
Hints	0	The system provides Hints to the learner.
	1	The system provides Hints to the learner according to strategy carried out by set of rules which have the following general form: **IF** Performance* **and** Help-seeking* **and** Engagement* **THEN** Hint type **ENDIF**

* Performance, Help-seeking and Engagement* are learner's stereotype

The features of all the instructional strategies are stored in the file called "FeatFile". In the file, each row represents a lesson/question strategy and each column represents a lesson/question strategy feature. The value of each feature in each row is represented as character with a fixed width of 1 byte. Table 3.5 represents the internal structure of the file.

The structure of the 'FeatFile' file is evident from its usage. The tutoring system is responsible for extracting particular information from the file and converting it into a format understandable by the user. The selection process of the proper instructional strategy for certain learning materials, lessons or questions, are based on a set of rules. The design of these rules is based on "reconfigurable" design in which the teacher, who has no background in the programming languages, can adjust the

selection process via the teacher-interface. The teacher can also adjust the feature of each instructional strategy.

Fields	Description
IS	The Instructional strategy
Feature 1	The first feature of the Instructional strategy (i.e. Practice for lesson strategies and time for question strategies)
Feature 2	The second feature of the Instructional strategy (i.e. Pre-requisites for lesson strategies and Feedback for question strategies)
Feature 3	The first feature of the Instructional strategy (i.e. Extra tutoring for lesson strategies and Hint for question strategies)

Table 3.5: The internal structure of the 'FeatFile' file

3.4 Architecture of the tutoring system

The proposed tutoring system is a server-side one written in PHP and HTML. Tutoring systems can be seen as a number of components that can communicate between each others. Components of a tutoring system are based upon the traditional ITS structure proposed by Hartley and Sleeman (1973). A unique feature of the tutoring system is the separation of the domain knowledge from the control process represented by the Course manager (CM). This can make it easier to maintain and modify the system without affecting the system's knowledge and permits the tutoring system to be used for other language domains.

The CM controls the learning process and provides the infrastructure necessary to tailor the presentation of the learning material to each learner. The CM contains two components: the learner model and the instructional model. The learner model is represented as multiple-stereotype model based on flexible fuzzy inference. The stereotype learner model represents the learner's Performance, Help-seeking and Engagement with the tutoring system. The instructional model comprises seven layers responsible for different learning activities. These layers are *Administrator model, Instructional strategies model, Lesson model, Feedback and Hinting model, Pre-requisites model, Extra tutoring model, Questions model, Practice model, Explanation model, and Comments editor*. The CM and its components are

41

described in more details in Chapter 4. Figure 3.3 illustrates the overall system architecture.
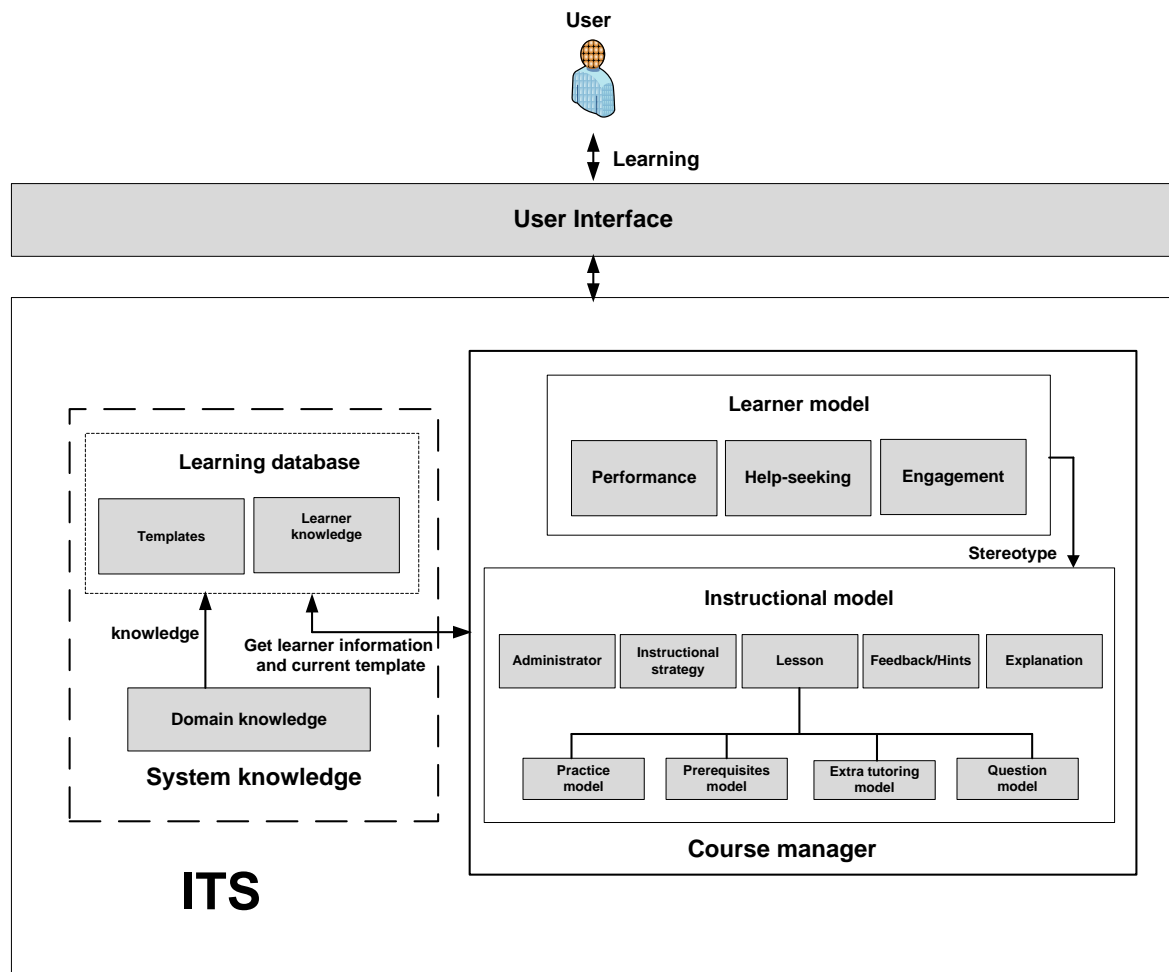
Figure 3.3: The architecture of the tutoring system

The system's knowledge contains two components: the domain knowledge and the learning database (LDB). LDB is a relational database which stores information about the system's templates and the learner knowledge state. The domain knowledge contains the knowledge of a foreign language (e.g. Arabic) broken down into 'topics' and represented as a set of files that will be discussed later on in this thesis.

3.5   Pluggable domain

As illustrated in Figure 3.4, the contents of the learning materials are represented as a set of templates (i.e. lessons, practices, or questions). The information for these

templates is collected from the domain knowledge. The contents of the domain knowledge itself can be altered by adding new materials (plug) or replacing old materials (unplug) by the teacher via the teacher-interface. This can reduce time and storage memory as well as make it easy to use the tutoring system to teach different language.
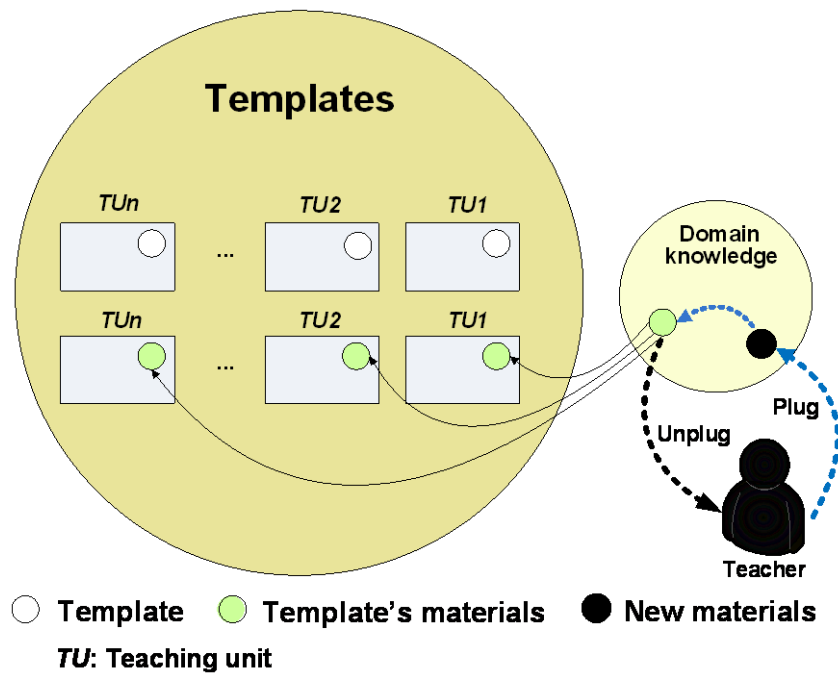
Figure 3.4: The tutoring system pluggable domain

3.6   Curriculum Design

The curriculum aims at engaging learners in active learning by the acquisition of knowledge or skills through direct experience of carrying out a task. The courseware is presented in the form of different Lessons, Questions, Practices, Extra tutoring, and Examination (see Figure 3.5). Given a Lesson (L), there is a Lesson's Practice (R), Lesson's Extra tutoring (Ex), n pre-requisites (Pr_1, Pr_2,…, Pr_n), and Lesson's Question (Q). Each Question has two-level on-demand hints (H_1, H_2) and two-level on-demand feedback (F_1, F_2). Lessons, Practices, Prerequisites, Extra tutoring, Questions, and Examination take the form of text and may include multimedia elements (picture, audio, and video) (Luppicini, 2007). The tutoring system follows the curriculum of Arabic grammar at Libyan primary schools.

43
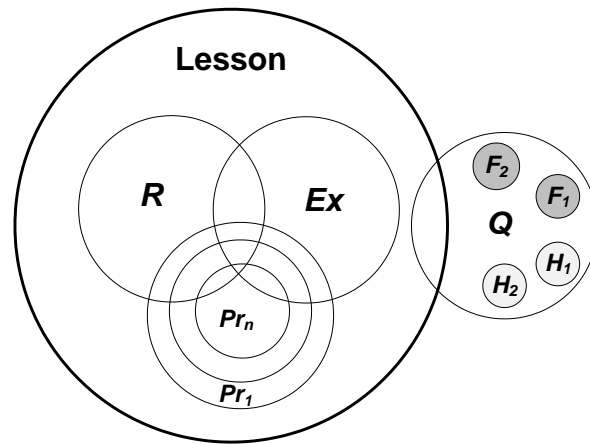
Figure 3.5: The tutoring system courseware

3.6.1   Lesson

The Lesson is a period of teaching of certain learning materials in which learners are taught about a particular language subject or how to perform a particular activity. Within each lesson, the learner finds teaching materials, and learner work, a list of readings, activities for enrichment, and links to multimedia, such as pictures, audio, or videos.

3.6.2   Practice

Practice is a method of learning and of acquiring knowledge by rehearsing the teaching materials taught by the lesson. The learner develops skills in practical manner such as grammar or listening and provides opportunities for learners to expand on what they have learned. Practice provides sample questions and their answers by which learners can asses themselves before answering the required lesson's questions. This influences learners learning by directing their attention to particular aspects of course content (Anderson and Elloumi, 2004).

3.6.3   Extra tutoring

Extra tutoring is a support mechanism in which the tutoring system provides additional teaching materials to certain learners in order to improve their

performance. Based on the selected instruction strategy, the tutoring system decides whether the learner needs Extra tutoring by analysing their previous interaction with the system (Anderson and Elloumi, 2004).

3.6.4   Question

The Question is presented to the learner in order to make an assessment and examine his status of comprehension. Two types of questions: selection and supply types are used by the tutoring system. A learner must answer the lesson's question before he/she can move to the next lesson.

3.6.5   Examination

An examination is a set of questions designed to determine the learner's knowledge or skills at certain learning level. A learner must answer a learning level examination after the completion of all the required teaching materials in that learning level. Before starting the learning process the learner takes a placement test which is designed to test his/her linguistic abilities in order to place them in a class that is appropriate for his/her level.

3.7   Summary

This chapter has presented the overall design of the tutoring system through its various components and the relationship between them. The system contains two learning modes of interactions with users (i.e. learner or teacher): interactive and discovery based learning modes. The design of the instructional strategies of the proposed system is outlined. The system provides multiple instructional strategies for Lessons and Questions. The system selects the proper instructional strategy for teaching a certain lesson/question based on its attributes. The selection of the instructional strategies and the feature of each instructional strategy can be adjusted by the teacher via the teacher-interface. An overview of the tutoring system architecture is presented and its major components are summarised. The system

has two major components: the Course manager and the System's knowledge. The concept of pluggable domain is introduced and the advantages of this concept are discussed. Finally, the courseware which takes the forms of Lessons, Practices, Extra tutoring, Questions, and Examination is explained. The design and the selection of the proper instructional strategy discussed in this chapter represent one of the contributions of this research. This design is based on reconfigurable multiple instructional strategies for both lessons and questions that embrace the learner model reasoning outcomes. The design allows for the introduction of the concepts of the pluggable domain approach to be discussed later in this thesis as another major contribution of this work. The following chapter will discusses the Course Manager and the structure of its main components.

CHAPTER 4: THE COURSE MANAGER DESIGN

This chapter introduces the Course Manager and its components which include the learner model and the instructional model. The learner model represents the learner's characteristics which are Performance, Help-seeking and Engagement. The instructional model contains seven models responsible for making different decisions.

4.1 The Course Manager

The Course Manager (CM) represents the control and the inference mechanism of the tutoring system. The goal of the CM is to promote the learning outcomes by balancing the goal of providing instruction with the goal of maintaining learner interest and motivation. The CM constantly analyses the learner's activities to adapt the learning to each individual learner. It records all the learner's actions and the corresponding system output in a log file. The teacher has access to the log files via the teacher-interface to analyse the learner interaction with the tutoring system.

Each action the learner performs in the learner-interface is sent to the CM which decides how to respond to it. If the submitted action is a solution to a certain problem, the CM sends it to the Learner model, which analyse the solution and updates the system. Based upon the learner request, the CM may also generate hints, feedback, or additional explanations.
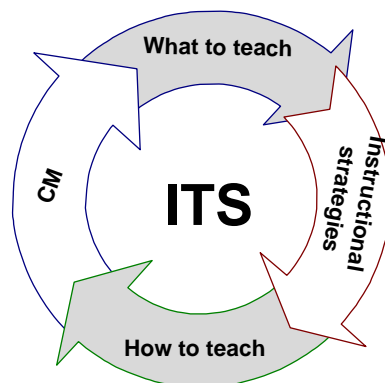
Figure 4.1: The interaction between the CM and the instructional strategies

Multiple instructional strategies based on the philosophy of differentiated instruction are employed in the learning process. The goal is to provide the CM with expertise on the proper strategy of how to teach certain learning materials. The interaction between the CM and the instructional strategies is shown in Figure 4.1.

4.2   Learner model

The learner model is represented using multiple stereotypes implemented as multiple fuzzy inferences. Fuzzy inferences are adapted to represent the learner model since most of the human reasoning is associated with the use of fuzzy rules. Fuzzy logic can be used to model the humans' decision making and common sense (Negnevitsky, 2005). This allows the overlapping of vague concepts and overcomes limitations such as lack of information (Turksen, 2005).

The aim is to imitate the teacher's knowledge acquisition process in evaluating the learner's knowledge and attitude toward the learning process (Wang and Chen, 2007). The advantage of using multiple fuzzy inferences is that it allows the system to use the outcomes of these fuzzy inferences individually or in combination with each other.

4.2.1   Reconfigurable Design

Developing an efficient WBITS that provides measurements to the progress and achievement of learners is difficult process given the different social, cultural and intellectual capital of learners. The challenge is that there is no "one design fits all" each one of these measurements. Therefore it is better to design these measurements based on the concept of "reconfigurable design" that can be fine-tuned by the teacher via the teacher-interface to suit the pedagogical needs and the teacher point of view, creating a more flexible environment.

In order to fulfil these requirements, a general fuzzy membership function $f(x)$ was developed in this research where $f(x):f(x)\rightarrow[0,1]$ is represented by four points *P1*,

P2, *P3*, and *P4*. Different forms of the function $f(x)$ are shown in Figure 4.2. The function *f(x)* is represented in Equation 4.1.

$$f(x) = \begin{cases} \dfrac{x - P1}{P2 - P1} & for \quad Increasing\ section\ (IS) \\[4mm] \dfrac{P4 - x}{P4 - P3} & for \quad Decreasing\ section\ (DS) \end{cases}$$

(4.1)
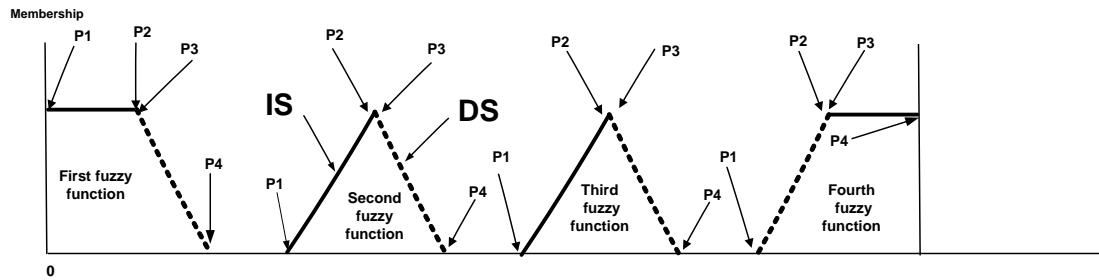
Figure 4.2: Different forms of the general fuzzy membership function

4.2.2   Fuzzy inference

The inference mechanism is based on Mamdani fuzzy inference method since it is successfully used in various ITSs (Mamdani and Assilian). The output of the fuzzy classification system is determined based on the class related to the consequent of the rule that reaches the highest degree of activation. If-then rules are used to formulate the conditional statements that comprise the fuzzy inference. The structure of these rules has the following general form:

Rule i: IF I_1 AND I_2 THEN I_3

where I_1 represents the first input linguistic set, I_2 is the second input linguistic set, and I_3 is the output linguistic set (i.e. stereotype). As a result of two linguistic sets, each of them having four classes, 16 different rules and its conditions are formulated.
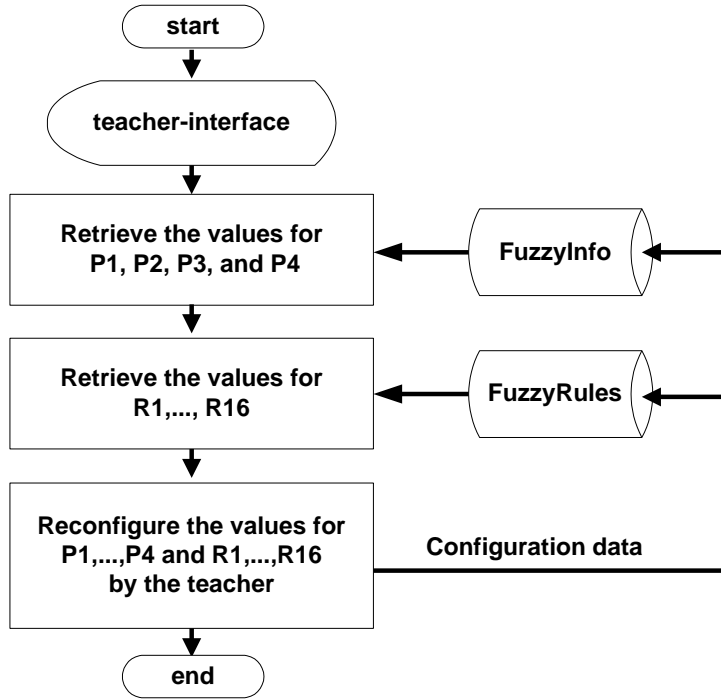
Figure 4.3: The reconfiguration process of the fuzzy membership functions and their rules by the teacher

The information of the fuzzy membership functions are coded and saved in the file "FuzzyInfo" and the information of their rules are coded and saved in the file "FuzzyRules". A sample of the initial values of the file "FuzzyInfo" is represented in Table 4.1. The structures and the initial values of the file "FuzzyRules" are represented in Table 4.2. In the FuzzyInfo file, *IND* is the index of the file, TITLE is the title (name) of each membership function (input or output), *TYPE* is the type of each function while *P1*, *P2*, *P3*, and *P4* represent the starting and the ending points of each function as previously discussed. In the FuzzyRules file, *IND* is the index of the file while the *Ri* is the output of *i*th rule. The reconfiguration process of the fuzzy membership functions and their rules by the teacher is shown in Figure 4.3.

	Type	Title	P1	P2	P3	P4
1	0	Poor	0.00	0.20	0.20	0.40
2	1	Good	0.20	0.40	0.40	0.60
3	1	Very good	0.40	0.60	0.60	0.80
4	0	Excellent	0.60	0.80	0.80	1.00
5	0	Weak	0.00	0.30	0.30	0.60
6	1	Below average	0.50	0.60	0.60	0.70
7	1	Above average	0.60	0.70	0.70	0.80
8	0	Outstanding	0.70	0.90	0.90	1.00
9	0	Underachiever	0.00	0.20	0.20	0.40
10	1	Fine	0.20	0.40	0.40	0.60
11	1	Strong	0.40	0.60	0.60	0.80
12	0	Excellent	0.60	0.80	0.80	1.00
:	:	:	:	:	:	:
36	0	Engaged	0.60	0.80	0.80	1.00

Table 4.1: The structure of the "FuzInfFile" file and
and a sample of its initial values

Table 4.1 key

	Feature	Description
1	Type	Type of the function (i.e. triangle or trapezoidal)
2	P1	The starting point of the increasing section
3	P2	The ending point of the increasing section
4	P3	The starting point of the decreasing section
5	P4	The ending point of the decreasing section

S	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16
1	1	1	2	3	2	2	3	3	2	3	3	4	2	3	4	4
2	1	1	2	3	1	2	2	3	2	3	3	4	2	3	4	4
3	1	2	2	3	1	2	3	3	2	3	3	4	2	3	4	4

R*i*: Rule number *i*; S: Stereotype (1 for performance, 2 for Help-seeking, and 3 for Engagement)
Table 4.2: The structure and the initial values of the "FuzzRulFile" file

4.2.3 Stereotype

A Stereotype represents a structured set of features or characteristics stored as a facet to which a value called 'stereotypes' is attached, or assigned to the learner. These features are Performance (Pr), Help-seeking (Hs) (Baker, 2007) and Engagement (Eg) (Anderson and Elloumi, 2004) (see Figure 4.4). These characteristics represent important factors in indicating the online learner's attitude towards the learning process (Luppicini, 2007). The system collects information about these characteristics from the learner activities during the learning process and tests them against the membership functions of each measurement. Each stereotype is represented by a fuzzy inference to provide an information flow closer

to the human thinking and makes the model friendly and easy to use, especially to non-expert users.

In order to design the membership functions for input/output, sub-ranges of the continuous (numeric) input/output variables (*IOVs*) were defined since all *IOVs* are in the range from 0 to 1. The measurement for all IOVs has four separate membership functions "linguistic variables" defining different ranges. Therefore, any *IOV* has four "truth values" — one for each of the four functions. The defuzzification process (i.e. decide the output class) is based on the Mean of Maximum Method (MOM) (Sivanandam *et al.*, 2007). The values W1 to W5 used in the defuzzification process are stored in the "DefuFile" file is shown in Table 4.3. These values divide the defuzzification space into two uniform lower classes and two uniform upper classes (U.S. Department of Education, n.d.). These values also can be modified via the teacher-interface.

Stereotype	W1	W2	W3	W4	W5
1	0.00	0.30	0.60	0.80	1.00
2	0.00	0.30	0.60	0.80	1.00
3	0.00	0.30	0.60	0.80	1.00

Table 4.3: The structure and initial values of the "DefuFile" file
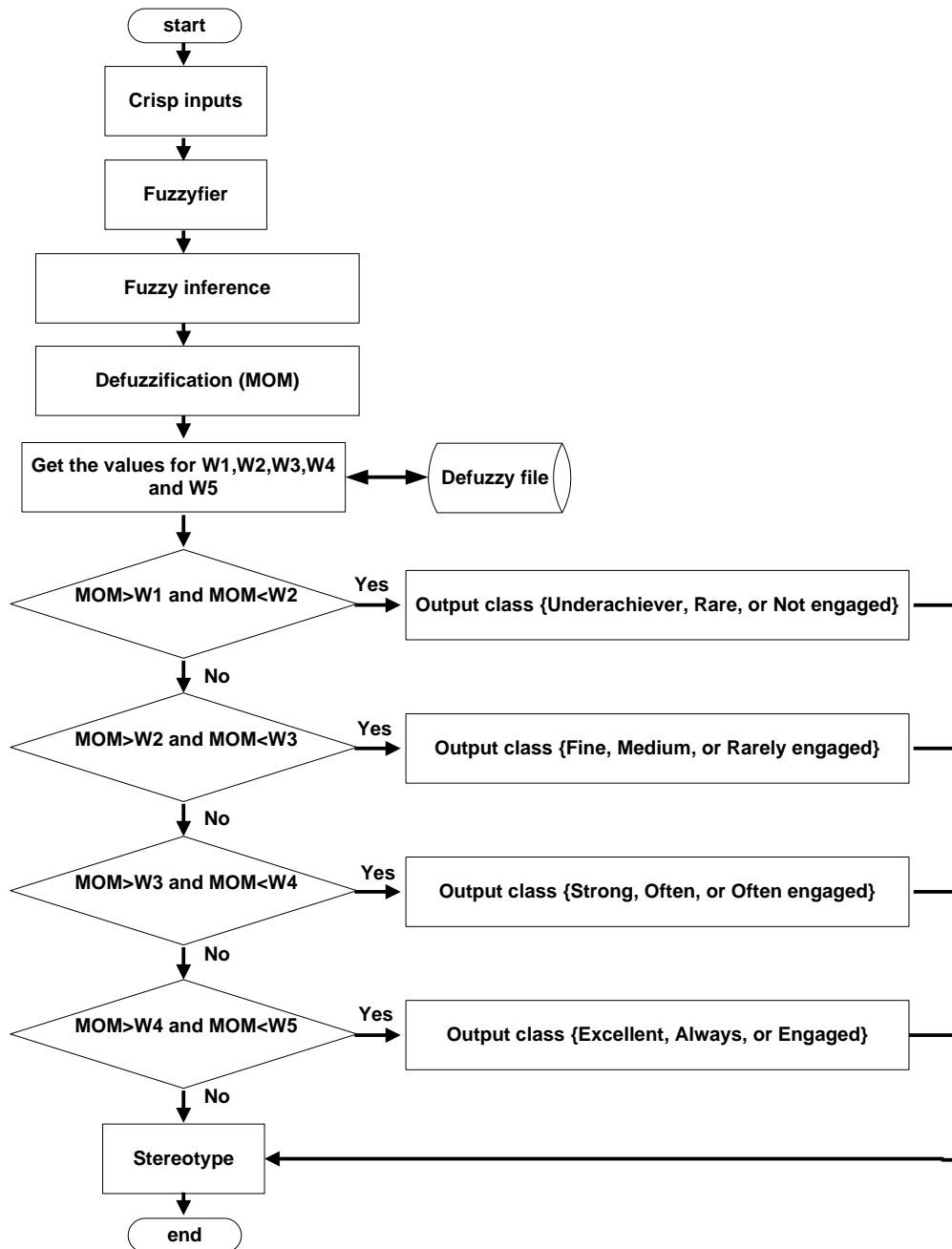
Figure 4.4: Fuzzy inference

In the next section the initial design of all the Performance, Help-seeking and Engagement fuzzy membership functions is presented and discussed. The design of all these function is based on the experience as discussed by Negnevitsky (2005). All these functions follow the general structure shown in Figure 4.2.

4.2.4   Performance

Learner's Performance is defined as the learner's involvement in his/her developmental courses, which results in an improvement of the presented course and a positive change in his/her attitude towards learning (Anderson and Elloumi, 2004; Luppicini, 2007). Average grades of both examinations and questions taken by the learner during the learning process are used as inputs to the fuzzy inference to calculate the Performance since questions and examination are well proven measurements of a learner's performance in a certain topic.

The first input is the learner's examination average grade (*EAG*). *EAG* represents the average grade value of all the learning levels' examinations taken by the learner plus the placement test result (*PG*). *EAG* can be calculated in equation 4.2 where *EX* is the number of examinations taken by the learner and *n* is the learning level.

$$EAG = \frac{\left(\sum_{n=1}^{3} Exam\ Grade_n + PG \right)}{EX} \qquad (4.2)$$

The second input is the learner's question average grade (*QAG*). *QAG* represents the average grade of all the questions answered by the learner in all the learning levels. *QAG* can be calculated in equation 4.3 *where Q* is the total number of questions taken by the learner, *n* is the learning level and *f* is the number of questions taken by the learner in each learning level.

$$QAG = \frac{\left(\sum_{n=1}^{3} \sum_{j=1}^{f} Question\ Grade_j \right)}{Q} \qquad (4.3)$$

In order to calculate the output Performance, two grading systems for both inputs: *QAG* and *EAG* are developed. These grading systems prescribe four groups of grades: "Poor", "Average", "Good" and "Excellent" which are similar to "D", "C", "B", and "A" in the academic grading system. Each group is represented by a membership function which defines how each point in the input space is mapped to a membership value or degree of membership (truth value from 0 to 1).

4.2.4.1　　Membership functions for EAG

The measurement for *EAG* grades has four separate membership functions defining grade ranges. These ranges are mapped with the functions $m_i(x)$: $m_i(x) \in \{$ Poor, Good, Very Good, Excellent$\}$.

The design of the membership function "Poor" covers the *EAG* from 0.0 to 0.40 with average grades less than 0.20 produce maximum truth value of 1. The design of the membership function "Good" covers the *EAG* from 0.20 to 0.60 with 0.40 producing a maximum truth value of 1. The design of the membership function "Very Good" covers the *EAG* from 0.40 to 0.80 with 0.60 producing a maximum truth value of 1. The design of the membership function "Excellent" covers the *EAG* from 0.60 to 1 with an average grade greater than 0.8 producing a maximum truth value of 1.

4.2.4.2　　Membership functions for QAG

The measurement for *QAG* grades has four separate membership functions defining grade ranges. These ranges are mapped with the functions $m_i(x)$: $m_i(x) \in \{$Weak, Below average, Above average, Outstanding$\}$.

The design of the membership function "Weak" covers the *QAG* from 0.0 to 0.60 with average grades less than 0.30 producing a maximum truth value of 1. The design of the membership function "Below Average" covers the *QAG* from 0.50 to 0.70 with an average grade of 0.60 producing a maximum truth value of 1. The design of the membership function "Above Average" covers the *QAG* from 0.60 to 0.80 with 0.70 producing a maximum truth value of 1. The design of the membership function "Outstanding" covers the *QAG* from 0.70 to 1 with average grades greater than 0.9 producing a maximum truth value of 1.

4.2.4.3　　Membership functions for Performance

Sub ranges of the continuous (numeric) output variable Performance are defined since Performance$\in[0,1]$. These ranges are mapped with the functions $m_i(x)$: $m_i(x) \in \{$Underachiever, Fine, Strong, Excellent$\}$. The "Excellent" class demonstrates

the learner convincing understanding of the thought language and his/her ability to complete all the learning tasks successfully. The "Strong" class demonstrates the learner understanding of thought language and his/her ability to complete most of the learning tasks successfully. The "Fine" class demonstrates the learner basic understanding of the thought language and his/her ability to complete certain learning tasks successfully. The "Underachiever" class demonstrates the learner limited ability to understand thought language and his/her inability to complete most learning tasks successfully.

The design of the membership function "Underachiever" covers inputs from 0.0 to 0.30 with inputs less than 0.10 produce maximum truth value of 1. The design of the membership function "Fine" covers inputs from 0.20 to 0.60 with the value of 0.40 producing the maximum truth value of 1. The design of the membership function "Strong" covers inputs from 0.40 to 0.80 with the value of 0.60 producing the maximum truth value of 1. The design of the membership function "Excellent" covers inputs from 0.60 to 1 with an average grade greater than 0.8 producing a maximum truth value of 1.

From the design of the previous membership functions, some similarity can be witnessed since all these functions deal with the learner's grades. However, the design of *QAG* makes it relatively harder for the learner to get high degree of membership in the classes "below average", "above average", "Outstanding". This is because learner has access to help (i.e. feedback and hints from the system) while answering the questions which is not the case when the learner takes an Examination. Therefore the functions "Good", "Very good", "Excellent" are increased in the *EAG* functions. The design of the output Performance's membership functions make it relatively easier for the learner to fit in to higher classes such as "Fine", "Strong", "Excellent". While the "Underachiever" class is reduced further to include fewer learners.

If-then rules are used to formulate the conditional statements that comprise the fuzzy inference. The input to each rule is the current values for the input variables (*EAG* and *QAG*), and the output is the entire fuzzy set which later be defuzzified, assigning one value to the output variable Performance. As a result of two fuzzy variables,

each of them having four classes, 16 different rules and its conditions are formulated. The complete lists of all the rules are presented in Table 4.4. The first two columns of the table represent the Antecedent of the membership functions (the if-part of each rule). The third column of the table represents the Consequent of the membership functions (the then-part of the rule).

			Antecedent		Consequent
	EAG			*QAG*	Performance
1	Poor			Weak	Underachiever
2	Good			Weak	Underachiever
3	Very good			Weak	fine
4	Excellent			Weak	strong
5	Poor			Below average	fine
6	Good	AND		Below average	fine
7	Very good			Below average	strong
8	Excellent			Below average	strong
9	Poor			Above average	fine
10	Good			Above average	strong
11	Very good			Above average	strong
12	Excellent			Above average	excellent
13	Poor			Outstanding	fine
14	Good			Outstanding	strong
15	Very good			Outstanding	excellent
16	Excellent			Outstanding	excellent

Table 4.4: The complete rules list of the learner Performance

4.2.5   Help-seeking

Learner's Help-seeking based on the number of hints and feedback requested by the learner is introduced as a second measurement of the learner's stereotype. The average number of hints (*ANH*) and the average number of feedback (*ANF*) requested by the learner during the learning process are used as inputs to the fuzzy inference to calculate the output Help-seeking since feedback and hints are the learner's only sources of help during the learning process.

The system provides just-in-demand two-layer of hints (H_t) to the learner for each Question. *ANH* can be calculated in equation 4.4. *P* is the number of the questions taken by the learner.

$$ANH = \frac{\sum_{t=1}^{2}(H_t)}{P*2} \qquad\qquad (4.4)$$

$$ANH = \begin{cases} ANH & if \quad ANH \leq 1 \\ \\ 1 & if \quad otherwise \end{cases}.$$

The system provides just-in-demand two-layer of feedback (F_s) for each Question. *ANF* can be calculated in equation 4.5.

$$ANF = \frac{\sum_{s=1}^{2}(F_s)}{P*2} \qquad\qquad (4.5)$$

$$ANF = \begin{cases} ANF & if \quad ANF \leq 1 \\ \\ 1 & if \quad otherwise \end{cases}.$$

4.2.5.1   *Membership functions for ANH*

The measurement for *ANH* has four separate membership functions defining different ranges where $ANH \in [0,1]$. These ranges are mapped with the functions $m_i(x)$: $m_i(x) \in$ {Few, Average, Above average, Large}.

x is the average number of hints requested by the leaner. The design of the membership function "Few" covers the learner's average number of hints from 0.0 to 0.60 with an average of less than 0.40 producing a maximum truth value of 1. The design of the membership function "Average" covers the learner's average number of hints from 0.40 to 0.70 with an average of 0.55 producing a maximum truth value of 1. The design of the membership function "Above average" covers the learner's average number of hints from 0.60 to 0.90 with an average of 0.75 producing

maximum truth value of 1. The design of the membership function "Large" covers the learner's average number of hints from 0.80 to 1 with an average greater than 0.9 producing a maximum truth value of 1. The membership function "Few" is extended while the higher membership functions {Average, Above average, Large} are slightly reduced to include less learners which helps to include more learners who received more hints. This is because hints provide little information about how to solve a certain problem.

4.2.5.2   Membership functions for ANF

The measurement for *ANF* has four separate membership functions defining different ranges where $ANF \in [0,1]$. These ranges are mapped with the functions $m_i(x)$: $m_i(x) \in$ {little, typical, frequent, numerous}.

x is the average number of feedback requested by the leaner. The design of the membership function "Little" covers the learner's average number of feedback from 0.0 to 0.40 with an average of less than 0.20 producing a maximum truth value of 1. The design of the membership function "Typical" covers the learner's average number of feedback from 0.30 to 0.70 with an average of 0.50 producing a maximum truth value of 1. The design of the membership function "Frequent" covers the learner's average number of feedback from 0.50 to 0.90 with an average of 0.70 producing a maximum truth value of 1. The design of the membership function "Numerous" covers the learner's average number of feedback from 0.70 to 1 with an average greater than 0.9 producing a maximum truth value of 1.

Unlike *ANH* membership functions, all *ANF* membership functions "typical", "Frequent", "Numerous" were extended to include more learners while the membership function "Little" was reduced to include less learners. This is because feedback provides more detailed information than hints.

4.2.5.3   Membership functions for Help-seeking

As with the last section, the output variable Help-seeking was defined since Help-seeking $\in [0,1]$. The measurement for Help-seeking has four separate membership

functions defining different ranges. These ranges are mapped with the functions $m_i(x)$: $m_i(x) \in$ {Rare, Medium, Often, Always}. The output Help-seeking determines how frequently learner relies on the tutoring system in order to get an easy answer to the presented question. Depending on the output Help-seeking value, the system chooses the proper remediation approach that suits each Lerner (as discussed in Section 2.9). The "Rare" class indicates that the learner infrequently seek help from the system. The "Medium" class indicates that the learner regularly seeks help from the system. The "Often" class indicates that the learner more frequently seeks help from the system. The "Always" class indicates that the learner is heavily seeks help from the system.

The design of the membership function "Rare" covers the learner's average number of requested feedback and Hints (Help seeking requests) from 0.0 to 0.70 with an average of less than 0.40 producing a maximum truth value of 1. The design of the membership function "Medium" covers the learner's average Help seeking requests from 0.40 to 0.70 with an average of 0.55 producing a maximum truth value of 1. The design of the membership function "Often" covers the learner's average number of Help seeking requests from 0.50 to 0.80 with an average of 0.65 producing a maximum truth value of 1. The design of the membership function "Always" covers the learner's average number of Help seeking requests from 0.70 to 1 with an average greater than 0.9 producing a maximum truth value of 1.

In Help-seeking membership functions, the membership function "Rare" is further extended to include more learners since the minority of the learners engage in harmful Help-seeking behaviour (Walonoski and Heffernan, 2006; Baker *et al.*, 2005) while the higher functions like "Medium", "Often," and "Always" are slightly reduced to include less learners. Given that the online learner commitment to the tutoring system is a personal decision and requires a strong desire to perform in order to achieve academic success.

The design of help-seeking functions encourages the system to provide more hints and feedback to learners. The intention is to create more helpful environment and avoid dissatisfaction and frustration with the tutoring system which occur in situations

when learners are blocked from receiving hints and feedback. Moreover, it has shown that learners stay online mainly when they feel looked after and supported by the system as reported by Cronje *et al.* (2006). However, Angelino *et al.* (2007) reported that the learner showed sign of confusion, anxiety, and frustration due to lack of feedback from the tutoring system.

The complete lists of all the fuzzy inference rules for Help-seeking are presented in Table 4.5.

	Antecedent				Consequent
	ANH			*ANF*	Help-seeking
1	Few			Little	Rare
2	Average			Little	Rare
3	Above average			Little	Medium
4	Large			Little	Often
5	Few			Typical	Rare
6	Average	AND		Typical	Medium
7	Above average			Typical	Medium
8	Large			Typical	Often
9	Few			Frequent	Medium
10	Average			Frequent	Often
11	Above average			Frequent	Often
12	Large			Frequent	Always
13	Few			Numerous	Medium
14	Average			Numerous	Often
15	Above average			Numerous	Always
16	Large			Numerous	Always

Table 4.5: The complete rules list of the learner Help-seeking

4.2.6 Engagement

Learner's engagement play an important role in learning and it can be defined as the substantial time devoted by the learner to a certain task or invested in meaningful educational practices (Nash, 2005; Angelino *et al.*, 2007). Different researchers suggest that learners who are more engaged in learning through ITSs are more likely to achieve success (Rishi *and* Govil, 2008).

A "time-on-task" measurement that represents how much time spent by the learner working on certain tasks (i.e. using the tutoring system) is used as the third measurement of the Stereotype in this research. Given that educational research have consistently found learner engagement to be related to positive educational outcomes and related academic learning time with learner achievement (Nash, 2005; Angelino *et al.*, 2007).

Different models of the learner's engagement developed by researchers to deal with learners interacting with computer-based tutors. Some systems use multiple choice questions in order to evaluate their teaching interventions, nevertheless, learners do not take this assessment seriously which make it difficult to determine which intervention is actually most effective (Beck, 2005). Other researchers have focused on the goal of estimating the probability of the learner disengagement with the tutoring system by tracing learner actions (Lloyd *et al.*, 2007). Some models set up a minimum threshold for time spent on the current problem. The drawbacks of these approaches that they focus on the learner current session and cannot detect changes in learner's engagement over time. Also, differences among learners and problems were ignored (Beck, 2005).

In this research Engagement is measured based on the total time spent by the learner working with the tutoring system since his/her first login to the system to the last login. The average working time (AWT) and the average number of working days (AWD) are used to represent the "time-on-task" measurements since they provide more realistic and accurate assessment of the learner engagement history with the system. These measurements are used as inputs to the fuzzy inference to produce the output Engagement. AWT represents the actual average working time per day in minutes during the learning process while AWD represents the average number of successful login (frequency of learner's login) to the system in which learner should spend at least β minutes in order for each login to be considered as Successful login (SLG) and included in the calculation process, while spending Ψ minutes produce maximum Engagement with the system.

The initial value assigned for β is 3 minutes since it is the minimum time necessary to study certain teaching unit by the learner. The initial value for Ψ is considered 120

minutes which is enough time for an underperformed learner to finish a certain lesson. These values are assigned by the teacher to suit the pedagogical needs of the tutoring system and they can be adjusted via the teacher-interface. *n* is the number of *SLG*. *AWT* and *AWD* can be calculated in equations 4.6 and 4.7 respectively.

$$AWT = \frac{\sum_{i=1}^{n}(SLG_i)}{(last\ login\ date - first\ login\ date) * \psi} \tag{4.6}$$

$$AWT = \begin{cases} AWT & if & AWT \leq 1 \\ 1 & if & otherwise \end{cases}.$$

$$AWD = \frac{No\ of\ SLG}{(last\ login\ date - first\ login\ date)} \tag{4.7}$$

$$where\ AWD = \begin{cases} AWD & if & AWD \leq 1 \\ 1 & if & otherwise \end{cases}.$$

4.2.6.1 Membership functions for AWT

The measurement for *AWT* has four separate membership functions defining different ranges where $AWT \in [0,1]$. These ranges are mapped with the functions $m_i(x)$: $m_i(x) \in \{Uninterested, Involved, Interested, Devoted\}$.

The design of the membership function "Uninterested" covers the learner's *AWT* from 0.0 to 0.30 with an average of less than 0.10 producing a maximum truth value of 1. The design of the membership function "Involved" covers the learner's *AWT* from 0.10 to 0.60 with an average of 0.35 producing a maximum truth value of 1. The design of the membership function "Interested" covers the learner's *AWT* from 0.40 to 0.90 with an average of 0.65 producing a maximum truth value of 1. The design of the membership function "Devoted" covers the learner's *AWT* from 0.70 to 1 with an average greater than 0.9 producing a maximum truth value of 1.

The design of the membership functions "Involved" and "Interested" which both are middling functions is extended to cover the learner's average working time from ≈ 12 minutes to 108 minutes given that 120 minutes is the proposed maximum working time per day. The "devoted" membership function represents learner with maximum commitment to the learning process whose average working time between ≈ 84 minutes and 120 minutes where the membership function maximum truth value of 1 produced at 120 minutes. The "Uninterested" membership function represents the least committed learner to the learning process with less than ≈ 12 minutes of working time per day produce the maximum truth value of this membership function.

4.2.6.2   Membership functions for AWD

The measurement for *AWD* has four separate membership functions defining different ranges where $AWD \in [0,1]$. These ranges are mapped with the functions $m_i(x)$: $m_i(x) \in$ {Uncommitted, Attracted, Concerned, Committed}.

The design of the membership function "Uncommitted" covers the learner's *AWD* from 0.0 to 0.30 with an average of less than 0.10 producing a maximum truth value of 1. The design of the membership function "Attracted" covers the learner's *AWD* from 0.25 to 0.65 with an average of 0.45 producing a maximum truth value of 1. The design of the membership function "Concerned" covers the learner's *AWD* from 0.50 to 0.90 with an average of 0.70 producing a maximum truth value of 1. The design of the membership function "Committed" covers the learner's *AWD* from 0.80 to 1 with an average greater than 0.9 producing a maximum truth value of 1.

Although online education is available and even extended to cover longer periods for the learners to continue their learning process, many do not take advantage of this opportunity for various reasons such as lack of Internet access and resources to enable them to gain. Since some of these reasons are external and the learner has no control over them, thus, both membership functions "Attracted" and "Concerned" which are middling functions were extended over a long range of *AWD* to include more learners. For the same reason, the membership function "Uncommitted" was reduced to include fewer learners. On the other hand, the membership function

"Committed" is designed to represent a learner who is highly committed to the system.

4.2.6.3　　Membership functions for Engagement

The output variable Engagement was defined since Engagement$\in\mathbb{R}$ and Engagement$\in$[0,1]. The measurement for Engagement has four separate membership functions defining different ranges. These ranges are mapped with the functions $m_i(x)$: $m_i(x)\in\{$ Not engaged, Rarely engaged, Often engaged, Engaged}. The output Engagement represents the learner commitment to the tutoring system in terms of the average number of days and the average number of hours per day the learner spent in the learning process. The "Not engaged" class indicates that the learner is not dedicated to the learning process in term of time and average number of days spent on the learning process. The "Rarely engaged" class indicates that the learner spent few days and little time per day dedicated to the learning process. The "Often engaged" class indicates that the learner is dedicated to the learning process by spending more time and number of days on the learning process. The "Engaged" class indicates that the learner is highly dedicated to the learning process by spending a lot of time per day on the learning process.

The membership function "Not engaged" covers the inputs from 0.0 to 0.25 with an average of less than 0.10 producing a maximum truth value of 1. The membership function "Rarely engaged" covers the inputs from 0.10 to 0.60 with input of 0.35 producing a maximum truth value of 1. The membership function "Often engaged" covers the inputs from 0.40 to 0.90 with input of 0.65 producing a maximum truth value of 1. The membership function "Engaged" covers the inputs from 0.60 to 1 with an average greater than 0.8 producing a maximum truth value of 1.

The Engagement membership functions were designed to encourage learner involvement with the system since it is characterized by maximally beneficial learning; for that reason, the membership function "Not engaged" is slightly reduced to include fewer unengaged learners while the membership functions "Rarely engaged", "Often engaged" and "Engaged" are extended to include more engaged learners.

The complete lists of all the fuzzy inference rules for Engagement are presented in Table 4.6.

	Antecedent			Consequent
	AWT		*AWD*	Engagement
1	Uninterested		Uncommitted	Not engaged
2	Involved		Uncommitted	Rarely engaged
3	Interested		Uncommitted	Rarely engaged
4	Devoted		Uncommitted	Often engaged
5	Uninterested		Attracted	Not engaged
6	Involved	AND	Attracted	Rarely engaged
7	Interested		Attracted	Often engaged
8	Devoted		Attracted	Often engaged
9	Uninterested		Concerned	Rarely engaged
10	Involved		Concerned	Often engaged
11	Interested		Concerned	Often engaged
12	Devoted		Concerned	Engaged
13	Uninterested		Committed	Rarely engaged
14	Involved		Committed	Often engaged
15	Interested		Committed	Engaged
16	Devoted		Committed	Engaged

Table 4.6: The complete rules list of the learner Engagement list

In conclusion the design of all the previous fuzzy functions and their non-numeric linguistic variables can have different reading by different teachers. This is because terms such as slightly higher or lower can have different interpretation by different teachers for different teaching context. Therefore the current design of these functions makes it easy for teachers to reconfigure these functions to suit nearly any learning experience or specific needs. These have the advantages of creating a teaching environment that encourages teachers to plan and control the learning process.

In the initial design of all these membership functions an attempt has been made to open more doors for learners not to close since it is believed that self-confidence arises not from removing hurdles such as evaluation but by making them fair and encouraging for the learners to gain self-confidence from knowing they can jump over these hurdles.

The advantage of using multiple fuzzy inferences is that it allows the system to use the outcomes of these fuzzy inferences individually or in combination with each other. The knowledge provided by the multiple fuzzy inference system is unfinished, and an exhausting process of interpretation is required by the CM in order to translate it into useful actions. The design of the instructional strategy combined with the characteristics of the learner model present one of the contributions provided in this research.

4.3 Instructional model

The Instructional model controls the behaviour of the system via collection of layers or models. These models are: *Administrator model, Instructional strategies model, Lesson model, feedback and hinting model, Pre-requisites model, Extra tutoring model, Questions model, Practice model, Explanation model, and Comments editor* (see Figure 4.5). Each model contains set of rules (productions rules) responsible for making different decisions. These rules are in the form of condition-action pairs to encode knowledge and represent a cognitive step performed by the learner or to be performed by the system (Stamper *et al.*, 2007).
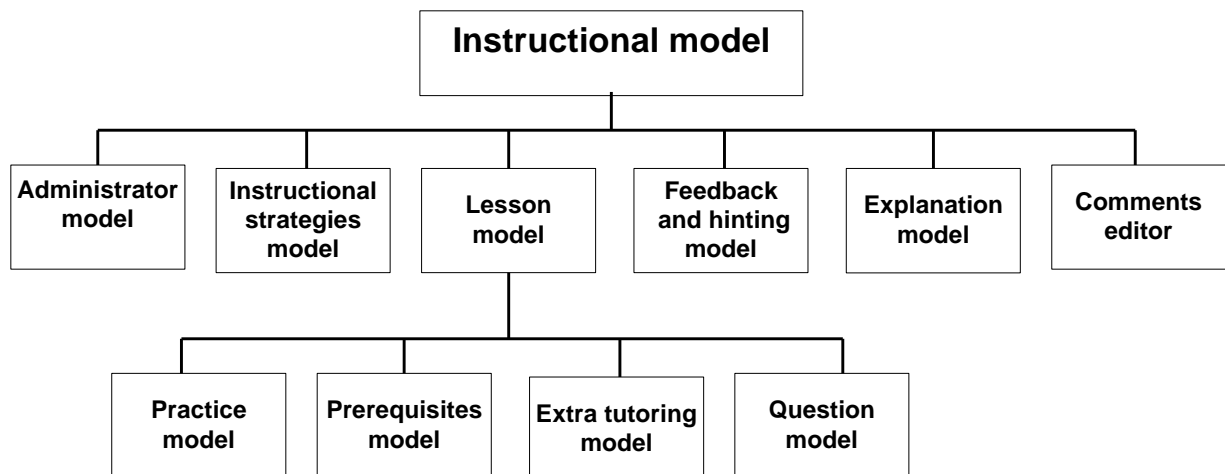
Figure 4.5: Instructional model

4.3.1 Administrator model

The administrator model controls the accountability of the learner and the teacher over who gets access to what features on the tutoring system. The administrator model also controls the pre-learning stage such as registration with the tutoring system, the presentation of the placement test examination (pre-test) and collecting the learner background information. At the registration stage the learner enters his/her personal information such as age, background of the taught language, gender and education. The placement test is designed to assess the learners' elementary knowledge of the taught language. In addition, it is aimed at identifying the learners' previous experience in the subject matter. Based on the placement test results the administrator model gives the learner the option to start in certain level. The rules responsible for the administration process are described as follows:

IF Learner is a NEW user

THEN register as a new Learner *AND* provide Learner ID

ENDIF

Save learner background information to *"LeraInfoFile"* file (see Table 4.7)

IF Learner has no Arabic language background

THEN placement test result = 0

ELSE presents Placement test examination

ENDIF

IF Learner ID is not verified

THEN presents a message

ENDIF

IF Learner ID is verified

THEN present learner-interface *AND* starts the learning process

ENDIF

Get the The start grade value of the first class (FC) and The start grade value of the second class (SC) from "PlacFile" (See Table 4.8)

IF placement test result >= FC and placement test result < SC

THEN learner can start learning from Beginner or Intermediate learning level

ENDIF

IF placement test result >= SC

THEN learner can start learning from Beginner, Intermediate, or Advanced learning level

ENDIF

IF ID entered is TeacherID *AND* ID verified

THEN present Teacher-interface

ENDIF

Fields	Description
ID	Learner ID
Name	Name
RFS	Reason for study
AB	Arabic background
FL	First language
AGE	Age
GEND	Gender
EDB	Educational background
PER	Performance
HES	Help-seeking
ENG	Engagement
LL	Learner learning level
CL	Current lesson

Table 4.7: The structure of the "LeraInfoFile" file

Fields	Description	Initial value
FC	The start grade value of the first class	80
SC	The start grade value of the second class	90

Table 4.8: The initial values of the "PlacFile" file

4.3.2   Instructional strategy model

Two approaches are developed for selecting the proper instructional strategy for certain teaching materials. The first approach is based on Artificial Neural Network (ANN) which is later replaced with a new approach that relies on flexible rules. The new approach has the advantage that it can be modified by the teacher via the teacher-interface. This provides more generality and flexibility to the selection process and allows the teacher to easily modify the instructional strategy to fit any new learning setup. In order to perform the same process using the ANN model the learner must retrain the ANN offline and come up with new parameters (e.g. weights and topology) that fit the new learning setup. This process is time consuming and requires technical skills that not every teacher necessarily possesses.

4.3.2.1 Artificial Neural Network (ANN)

ANN is used to select the proper instructional strategy (**y**) where the input attributes set (**x**) includes pedagogical parameters from learning materials such as lessons and questions. The complete list of these attributes and their values are discussed in Section 3.3. The selection process is the task to learn a target function **f** that maps each attributes' set **x** to one of the predefined labels **y**. Based on the design of the instructional strategies, each combination of these attributes is mapped into certain instructional strategy. The *ANN* has 5 inputs representing the inputs' attributes and 3 binary outputs representing each instructional strategy.

The selection process is based on Back propagation *ANN* where 72 different combinations of all attributes values were generated as inputs to the *ANN*. From this data, 80% is used for training the *ANN* and 20% randomly selected for testing (Heaton, 2008). The *ANN* role is to select each instructional strategy based on the current combination of the inputs' attributes (see Figure 4.6).

Inputs' attributes	→	*ANN*	→	Instructional strategy

Figure 4.6: The selection process using *ANN*

4.3.2.2 Reconfigurable rules

This set of rules is responsible for selecting the proper instructional strategy for certain learning materials: lessons or questions. The design of these rules is based on "reconfigurable" design in which the teacher, who has no background in the programming languages, can adjust the selection process via the teacher-interface. The teacher can also adjust the feature of each instructional strategy.

The modification process involves the following three steps:

1. Connecting a certain set of learning materials attributes to a certain instructional strategy and storing this information in a "StratFile" file. This information is used by the CM in the selection process (step 3) of the proper instructional strategy for certain learning materials attributes. In the Strategies file, each row represents an

70

instructional strategy and each column represents an input's attribute. The value of each attribute is represented as character with fixed width of 1 byte. The structure and the initial values of the "StratFile" file are shown in Table 3.2.

2. Generating a new set of patterns each with 5 inputs (attributes): learning level, difficulty, time, category and type, and one output (instructional strategy). A set of generic rules are developed to perform the "*Generation*" process based on any changes done on the contents of the "Strategies" file. These rules are used to built a variable "***Sentence***" that is embedded in the design of MySQL inquiry as follows:

Update "Strategy" Where Sentence

The complete set of rules responsible for generating the instructional strategies is presented in the next section.

IF (learning level = [Beginner *OR* Intermediate *OR* Advanced]
 OR [Beginner *AND* Intermediate]
 OR [Beginner *AND* Advanced]
 OR [Intermediate *AND* Advanced]
 OR [Beginner *AND* Intermediate *AND* Advanced])
THEN sentence=sentence+ "learning level=([Beginner *OR* Intermediate *OR* Advanced]
 OR [Beginner *AND* Intermediate]
 OR [Beginner *AND* Advanced]
 OR [Intermediate *AND* Advanced]
 OR [Beginner *AND* Intermediate *AND* Advanced])"
ENDIF

IF (Difficulty = [Easy *OR* Medium *OR* Hard]
 OR [Easy *AND* Medium]
 OR [Easy *AND* Hard]
 OR [Medium *AND* Hard]
 OR [Easy *AND* Medium *AND* Hard])
THEN sentence=sentence+ "Difficulty = ([Easy *OR* Medium *OR* Hard]
 OR [Easy *AND* Medium]
 OR [Easy *AND* Hard]
 OR [Medium *AND* Hard]
 OR [Easy *AND* Medium *AND* Hard])"
ENDIF

```
IF (Time = [Short OR Full]
    OR [Short AND Full])
THEN sentence=sentence+ "Time = ([Short OR Full]
    OR [Short AND Full])"
ENDIF


IF (Category = [Descriptive OR Procedural]
    OR [Descriptive AND Procedural])
THEN sentence=sentence+ "Category = ([Descriptive OR Procedural]
    OR [Descriptive AND Procedural])"
ENDIF
```

3. Selecting the proper instructional strategy based on the collected combination of inputs' attributes for the current learning materials (i.e. lesson or question). This process is described by the following MySQL query:

Select "instructional strategy" from "Strategies" Where "attributes"

4.3.3　　Lesson model

The lesson model uses the instructional strategies and the learner model to provide curriculum sequencing. The lesson model contains rules responsible for sequencing each lesson and its elements: such as Prerequisites, practice, extra tutoring and questions. These rules are described as follows:

```
IF start learning
THEN start learning session
ENDIF
IF learner chooses Interactive-based learning
THEN check all the lessons order
    AND retrieve the lesson with next order
    AND call "Instructional strategies" model
ENDIF
IF learner clicks "Prerequisites" button
THEN Call Pre-requisites model
ENDIF
IF learner clicks "Practices" button
THEN Call Practices model
```

ENDIF

IF learner clicks "Extra tutoring" button

THEN Call Extra tutoring model

ENDIF

IF all the requited lesson's elements are studied

THEN retrieve the lesson's Question

    AND Call "Instructional strategies" model

    AND Call Question model

ENDIF

IF all the required lesson elements are studied

    AND the lesson's Question are studied

THEN move to the next lesson order in the learning level

ENDIF

IF learner chooses discovery based learning

THEN select the discovery based learning mode

ENDIF

4.3.4 Feedback and Hinting model

The feedback and hinting model tracks the learner's work and tailors its feedback and hints based on the selected instructional strategy and the learner's stereotype. In general the feedback and hints are provided to the learner based on the equation: Rfh >= Pfh where *Rfh* are the requested feedback and hints by the learner and *Pfh* are the presented feedback and hints by the system. The feedback and hinting model provides two-layer of just-on-demand feedback and hints upon learner request. Furthermore, the model produces immediate corrective feedback when the learner submits the answer to the lesson's question. Figure 4.7 represents the feedback and hinting process.
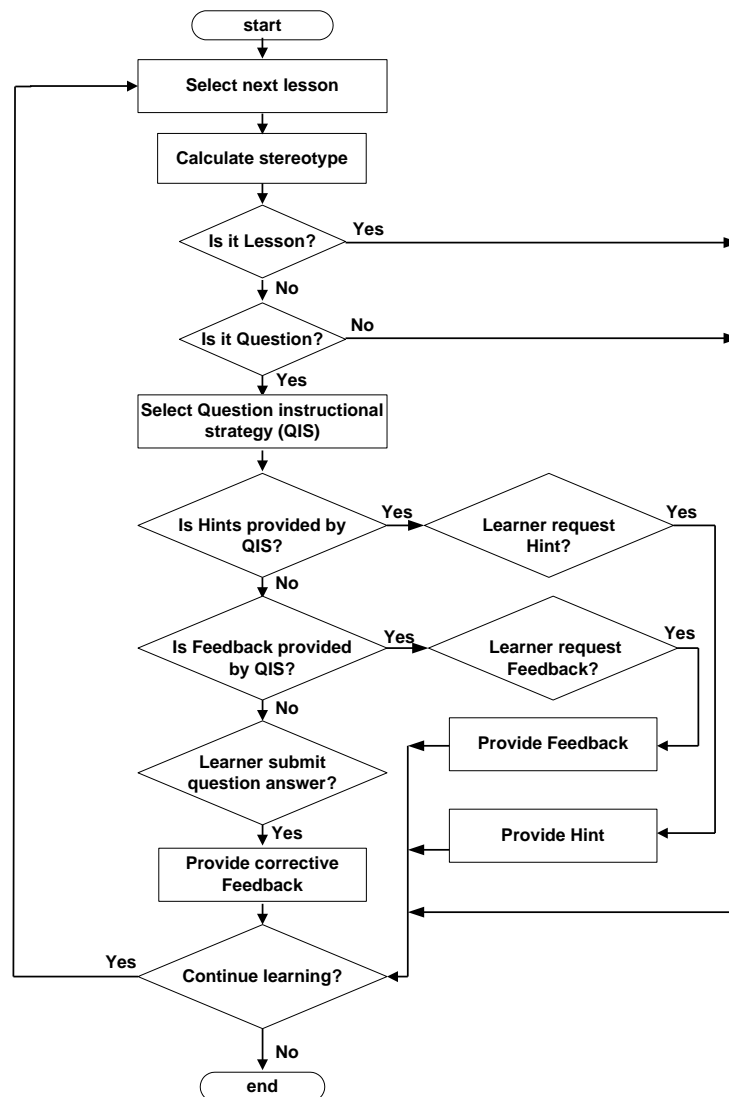
Figure 4.7: Feedback and hinting process

All the rules responsible for feedback and hinting process can be personalised by the teacher via the teacher-interface.

4.3.4.1   *Feedback*

Two types of feedback are generated by the feedback and hinting model: requested and corrective feedback. Firstly, the requested feedback is presented when learner makes a deliberate effort to get feedback by clicking a "Feedback" button. The system provides two-layer of requested feedback to the learner according to strategy carried out by set of rules which have the following general form:

IF Performance
    AND Help-seeking
    AND Engagement
THEN Feedback type
ENDIF

Table 4.9 shows a sample of three possible values that Performance (Pr) and Help-seeking (Hs) and Engagement (Eg) will take and their corresponding action when applied using the above condition. 64 different rules are developed to perform the feedback process. Each rule is accompanied by a message designed by the teacher. This message represents the current state of the learner and the number of feedback (TOF) he/she will receive.

	Pr	Hs	Eg	TOF	Message
1	1	1	1	1	Although you rarely depend on Help from the system to solve Questions nevertheless your Performance is very weak and you rarely spend adequate time in the Tutoring system.
2	1	1	2	1	Although you rarely depend on Help from the system to solve Questions nevertheless your Performance is very weak and you spend little time in the Tutoring system
3	1	1	3	2	Although you rarely depend on Help from the system to solve Questions and you spend More time in the Tutoring system nevertheless your Performance is very weak. Try to concentrate more when you solve the questions in the future.

Table 4.9: Sample of the feedback strategy

Table 4.9 Key

Stereotype	Values	Description
Performance (Pr)	1	Underachiever
	2	Fine
	3	Strong
	4	Excellent
Help-seeking (Hs)	1	Rare
	2	Medium
	3	Often
	4	Always
Engagement (Eg)	1	Not engaged
	2	Rarely engaged
	3	Often engaged
	4	Engaged
Type of feedback (TOF)	0	No feedback
	1	First feedback
	2	Second feedback

Secondly, the corrective feedback is presented immediately to the learner after submitting the lesson's questions answers. The corrective feedback can be described by the following rules:

IF Learner starts the lesson's Questions

THEN retrieve the learner's stereotype

    AND Rule1: Calculate the average time spent on discovery based learning "Discovery average time" (*DAT*)

ENDIF

IF Answers are submitted

THEN Display corrective and analysis feedback based on the learner previous "stereotype" and "*DAT*"

ENDIF

Rule1: The calculation of DAT process:

Retrieve Login and Logout time for discovery based learning from "DiscLogFile" file (see Table 4.10).

Retrieve Minimum and Maximum time designed by the teacher from "EngagFile" file (see Table 4.11).

Calculate the Time (in minutes) spent by the learner of each login in discovery based learning.

Calculate the Number of Discovery Login Days (NLD).

IF Time >= Minimum ***AND*** Time < Maximum

THEN Total time = Total time + Time

ENDIF

IF Time >= Maximum

THEN Total time = Total time + Maximum

ENDIF

Calculate *DAT* by the Equation *DAT* = (Total time) / (*NLD* * Maximum).

Retrieve the values of W1 to W5 from "DefuFile" file (see Table 4.3).

IF DAT > *W1* ***AND*** *DAT* <= *W2*

THEN DAT = "Hardly ever"

ENDIF

IF DAT > W2 ***AND*** *DAT* <= *W3*

THEN DAT = "Every now and then"

ENDIF

IF DAT > W3 ***AND*** *DAT* <= W4

THEN DAT = "Regularly"

ENDIF

IF DAT > W4 ***AND*** *DAT* <= *W5*

THEN DAT = "Constantly"

ENDIF

Fields	Description
LID	The learner ID
DATE	The learner login date to the discovery based learning
Login time	The learner login time to the discovery based learning
Logout time	The learner logout time from the discovery based learning

Table 4.10: The structure of the "DiscLogFile" file

Fields	Description	Initial value
EngMin	Engagement minimum value	3
EngMax	Engagement maximum value	120
DisMin	Discovery minimum value	0
DisMax	Discovery maximum value	15

Table 4.11: The initial values of the "EngagFile" file

4.3.4.2 Hinting

Similar to the feedback process, the hints are presented when learner clicks a "Hint" button. The system provides two-layer of requested hints to the learner according to the hinting strategy. The hinting strategy is similar to the feedback strategy discussed in Section 4.3.4.1.

	Pr	Hs	Eg	TOH	Message
1	1	1	1	1	Only the First Hint will be presented to you
2	1	1	2	1	Only the First Hint will be presented to you
3	1	1	3	2	The first and the second feedback will be presented to you

Table 4.12: Sample of the Hints strategy

Table 4.12 Key

Stereotype	Values	Description
Performance (Pr)	1	Underachiever
	2	Fine
	3	Strong
	4	Excellent
Help-seeking (Hs)	1	Rare
	2	Medium
	3	Often
	4	Always
Engagement (Eg)	1	Not engaged
	2	Rarely engaged
	3	Often engaged
	4	Engaged
Type of Hint (TOH)	1	First Hint
	2	Second Hint

Table 4.12 shows a sample of three possible values that Performance (Pr) and Help-seeking (Hs) and Engagement (Eg) will take and their corresponding action when applied using the above condition. 64 different rules are developed to perform the Hint process. Each rule accompanied by a message designed by the teacher. This message represents the current state of the learner and the type of hints (TOH) he/she will receive. These rules and their accompanied messages can be personalised by the teacher using the teacher-interface.

4.3.5 Pre-requisites model

The lesson's pre-requisites are different lessons precede the current lesson. The learner must study the lesson's pre-requisites if it is required by the selected instructional strategy otherwise he/she can skip the lesson's prerequisites. The question model can be described by the following rules:

> *IF* studying the pre-requisites is required by the selected instructional strategy
>   *AND* learner clicks "prerequisites" button
> *THEN* save time as "Initial Time"
>   *AND* presents the pre-requisites window
>   *AND Rule1*: EXIT process
> *ENDIF*

> *Rule1: EXIT process:*
> *IF* learner clicks "EXIT" button
> *THEN* Calculate time as "Final Time"
> *ENDIF*
> *IF* (Final time-Initial time)< 3 minutes
> *THEN* GO back to the pre-requisites window
> *ENDIF*
> *IF* (Final time-Initial time)>= 3 minutes
> *THEN* GO back to the Lesson model
> *ENDIF*

4.3.6 Extra tutoring model

Extra tutoring provides extra learning materials to the learner. The learner must study the lesson's Extra tutoring if it is required by the selected instructional strategy

otherwise he/she can skip this tutoring. The Extra tutoring model can be described by the following rules:

> ***IF*** studying the Extra tutoring is required by the selected instructional strategy
>> ***AND*** learner clicks "Extra tutoring" button
>
> ***THEN*** save time as "Initial Time"
>> ***AND*** presents the Extra tutoring window
>> ***AND Rule1***: EXIT process
>
> ***ENDIF***
>
> ***Rule1: EXIT process:***
> ***IF*** learner clicks "EXIT" button
> ***THEN*** Calculate time as "Final Time"
> **ENDIF**
> ***IF*** (Final time-Initial time)< 3 minutes
> ***THEN*** GO back to the Extra tutoring window
> **ENDIF**
> ***IF*** (Final time-Initial time)>= 3 minutes
> ***THEN*** GO back to the Lesson model
> ***ENDIF***

4.3.7 Practice model

The practice provides interactive teaching materials and some questions relates to the current lesson. After answering a question the Practice model presents personalised analysis of the learner answer based on the current and previous interaction with the practice model and his/her stereotype. The learner interaction with the practice model is stored in the file "PractFeed" as described by Table 4.13.

Fields	Description
LID	Learner ID
PID	Practice ID
LL	Learning level
Grammar	Grammar results
Listening	Listening results
Reading	Reading results

Table 4.13: The structure of the file "PractFeed"

The Practice model can be described by the following rules:

IF Learner starts the lesson's Practice

THEN Select the lesson instructional strategy

   AND retrieve the learner's stereotype

   AND retrieve the learner's previous practice interactions

   AND presents the lesson's Practice

ENDIF

IF Learner is required to study the lesson's Practice by the lesson instructional strategy

THEN save time as "Initial Time"

ENDIF

IF Learner answer Practice's Questions

THEN Display Questions' answers

   AND corrective feedback

ENDIF

IF Learner is NOT required to study the lesson's Practice

   AND learner press "EXIT" button

THEN GO back to the lesson

ENDIF

IF Learner is required to study the lesson's Practice

   AND learner press "EXIT" button

   AND (the Current Time – Initial Time) <3 minutes

THEN GO back to the Practice

ELSE GO back to the lesson

ENDIF

4.3.8 Question model

As previously discussed in Section 3.2.1., the learner must answer the lesson's questions after completing all the required teaching units (i.e. prerequisites, practice and extra tutoring) and before moving to the next lesson. The question model responsible for providing questions and presenting personalised corrective feedback and analysis to the learner's answers. The answers of the questions affect the learner's performance. The question model can be described by the following rules:

IF Learner starts the lesson's Questions

THEN Select the Question instructional strategy

   AND presents the Question

ENDIF

IF Learner request feedback

THEN provide feedback according to the selected instructional strategy and the feedback and hinting model

ENDIF

IF Learner request Hint

THEN provide Hint according to the selected instructional strategy and the feedback and hinting model

ENDIF

IF Learner has unlimited time to answer questions

    AND submit answers

THEN calculate Question Average Grade (QA)

ENDIF

IF Learner must answer questions within time

    AND submit answers within time

THEN calculate Question Average Grade (QA)

ENDIF

IF Learner must answer questions within time

    AND did not submit answers within time

THEN $QA=0$

ENDIF

IF Question Q_i is correct

THEN display Q_i is correct

ELSE display Q_i is incorrect

ENDIF

IF Answers are submitted

THEN Display QA **AND** Display corrective and analysis feedback

ENDIF

4.3.9 Explanation model

The explanations model provides overview and guidance to the learner regarding his current and previous interaction with the tutoring system. The explanation model has two modes: lesson and question mode. Firstly the lesson mode provides an explanation to the learner during the lesson session. This mode can be described by the next rule:

IF Learner starts the Lesson's Explanation

THEN Retrieve the current lesson information

    AND Retrieve the learner current information

    AND Select the lesson instructional strategy

AND Retrieve the learner's stereotype

AND Calculate the *DAT*

AND presents Explanation according to the learner's current information, stereotype, selected instructional strategy, and *DAT*

ENDIF

Secondly, the question mode provides explanation to the learner during the question session. This mode can be described by the next rule:

IF Learner starts the Question's Explanation

THEN Retrieve the current Question information

AND Retrieve the learner current information

AND Select the question instructional strategy

AND Retrieve the learner's stereotype

AND Calculate *DAT*

AND presents Explanation according to the learner's current information, stereotype, selected instructional strategy, and *DAT*

ENDIF

4.3.10   Comments editor

The learner-interface provides "Comments editor" in which the learner can use to write any comments during the learning process. These comments can be retrieved or deleted at any time by the learner. Each comment can be given a title and important level (normal and important). The learner is reminded by the system to check his comments by a moving arrow according to the following rules:

IF there is a comment in the "ComE" file

THEN retrieve the last date are checked by the learner

AND retrieve the current date

ENDIF

IF the date difference between the current date and the last date greater than 7 days

THEN activate the arrow reminder

ENDIF

The data for this process is stored in "ComE" file. The structure of the "ComE" file is shown in Table 4.14.

Fields	Description
IND	Index
LID	Learner ID
TIT	Title of the Comments
REM	Comments contents
IMP	Importance of the Comments

Table 4.14: The structure of the "ComE" file

4.4   Summary

This chapter has presented the overall design of the CM and its components which include: the learner model and the Instructional model. The learner model is based on stereotype learner model. The stereotypes represent three different learner's characteristics: these are Performance, Help-seeking and Engagement. These characteristics are implemented as multiple-fuzzy inference. The design of the learner model can be reconfigured by the teacher via the teacher-interface at runtime. This has the advantage of allowing the teacher to readjust the systems parameters to fit various learning setup. The instruction model controls the behaviour of the system through collection of layers or models: these models are Administrator model, Instructional strategies model, Lesson model, feedback and hinting model, Pre-requisites model, Extra tutoring model, Questions model, Practice model, Explanation model, and Comments editor. The description of each model is presented through set of IF-THEN rules and tables. The design of the learner model presented in this chapter makes one of the contributions of this research. The learner model implemented as a reconfigurable multiple-fuzzy inference based on a general fuzzy membership function and flexible fuzzy rules. This has the advantage of providing multiple reasoning outcomes that can be used separately or in combination with each other. The following chapter discusses the system's knowledge and its main components.

CHAPTER 5: SYSTEM'S KNOWLEDGE DESIGN

This chapter introduces the system's knowledge and its main components which are the learning database and the domain knowledge.

5.1   System's Knowledge

The system's knowledge contains the overall knowledge of the system presented using a set of system templates such as lesson, questions, practices and Examinations. It contains two major components: the Learning database and the Domain knowledge. These components are thoroughly discussed in the subsequent sections.

5.2   Learning database

The learning database (LDB) records information about the learner's knowledge state (i.e. Performance) and personality (i.e. Help-seeking and Engagement). The LDB contains the learner's information (i.e. personal or academic). This information is used by the learner model and the instructional model, and it is vital for the learning process. Moreover, the learning database contains the templates of the system (e.g. questions, lessons and practices) while the actual data is stored in the domain knowledge (DK).

A relational database with a single repository stored in a MySQL database is used to represent the LDB. The design of the LDB is such that it must be able to provide all the information needed by the Course manager (CM) during the learning process in order to fit the construction of the new language teaching materials.

The main entities in the system are Learner, Lesson, Question, Examination, and Practice, Login and Prerequisites. The entity relation diagram (ERD) of the learning database is shown in Figure 5.1.
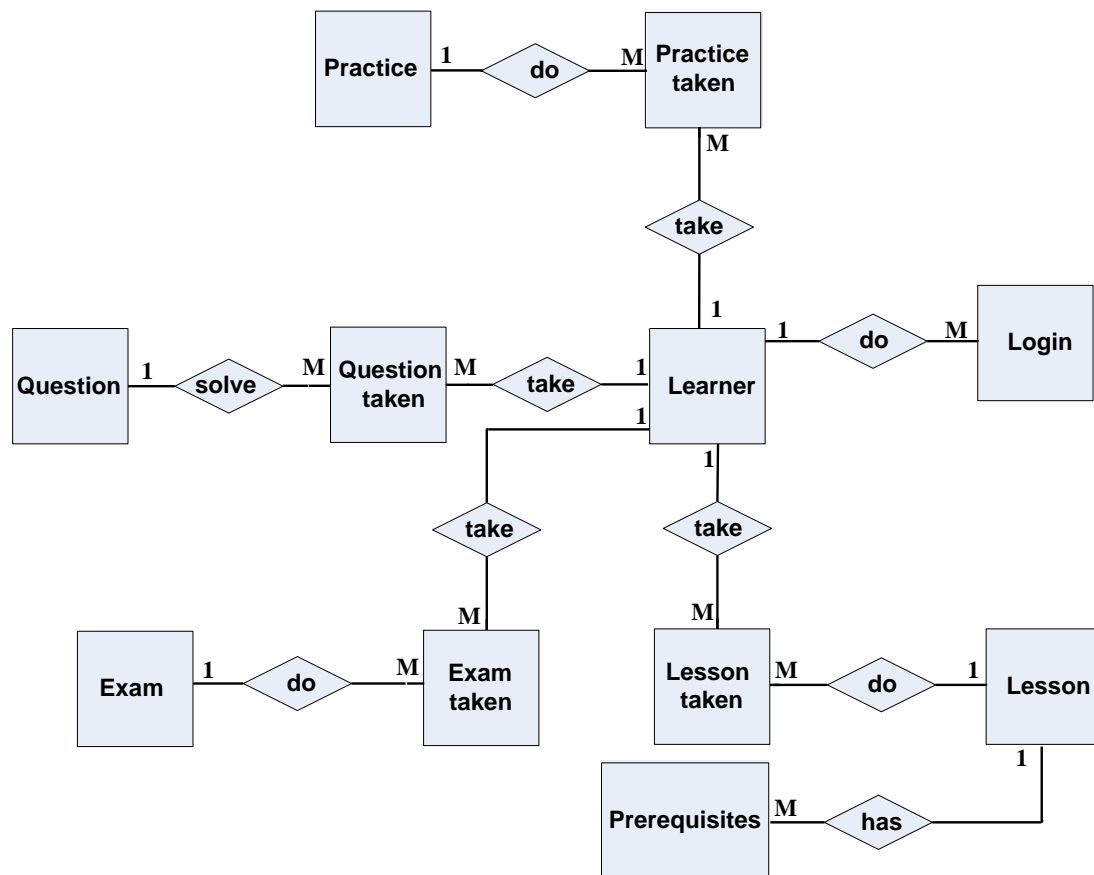
Figure 5.1: Entity relationship diagram (ERD) of the learning database

5.3   Domain knowledge

The DK is designed to summarise the important background information about foreign languages teaching that suit a casual learner of foreign languages such as a businessman or a tourist. The Domain Knowledge (DK) contains knowledge about the taught language (i.e. Arabic) represented as a set of files. The design of these files is based on a general design that can be utilised in the teaching of different languages (e.g. Arabic or French).

Each file (if applicable) in the DK represents all the language's narrative modes (i.e. First, Second and Third persons), different forms of singular, dual (i.e. Arabic has this form), plurals, and the different forms of masculine and feminine. The general structure of the DK files is shown in Table 5.1. Sample of the general structure of the DK files is shown in Table 5.4.

	Narrative mode	Singular/dual/plural form	Gender
1	First person	Singular	Muscular
			Feminine
		Dual	Muscular
			Feminine
		Plural	Muscular
			Feminine
2	Second person	Singular	Muscular
			Feminine
		Dual	Muscular
			Feminine
		Plural	Muscular
			Feminine
3	Third person	Singular	Muscular
			Feminine
		Dual	Muscular
			Feminine
		Plural	Muscular
			Feminine

Table 5.1: The general structure of the DK files

The DK contains the knowledge of the taught language stored in a set of files. Each file represents some components of the course structure such as Adverbs and Nouns. The contents of each file can be used by the system templates during the learning process.

The following clarifications should be considered when reading the next sections:

- Some files such as "characters file "CharFile" and number file "NumFile" contain permanent information. In other files such as noun file "NounFile" and verb file "VerbFile", information can be added/deleted by the teacher, therefore; only a sample of the data is shown.
- System audios are of the type "wave" (*.wav). The images are of the type "Joint Photographic Experts Group" (*.jpg) and "Graphics Interchange Format" (*.gif). The accompanied videos multimedia are of the type "Shockwave Flash" (*.swf).

86

5.3.1 Components of the DK

The structure and sample of the contents of the Arabic characters' file "CharFile" is shown in Table 5.2. Various components of the DK are used by the tutoring system. These components are Nouns, Verbs, Pronouns, Negation forms, Demonstratives, Prepositions, and Adjectives distinguish between singular, Dual and Plural as well as it is more precise about gender: masculine and feminine. These components are summarised in Table 5.3.

IND	NL	Title	Unicode	DESC	PIC	AUD	VED
1	أ	Alef	أ	The description of the character Alef.	-	-	-
2	ب	Baa	ب	The description of the character Baa.	-	-	-
3	ت	Taa	ت	The description of the character Taa.	-	-	-
4	ث	Thaa	ث	The description of the character Thaa.	-	-	-
5	ج	Jeem	ج	The description of the character Jeem.	-	-	-
6	ح	H'aa	ح	The description of the character H'aa.	-	-	-
:	:	:	:	:	:	:	:
28	ي	Yaa	ى	The description of the character Yaa.	-	-	-

Table 5.2: The structure of the "CharFile" file and sample content

Components	Description
Nouns	Arabic Nouns "NounFile" (and their modifying adjectives) are either definite or indefinite (there is an article for the definite state only). A noun is definite (in Arabic: معرفة marifa) if it has the definite article prefix (al-) (In Arabic: ال)
Verbs	Arabic verbs "VerbFile" can take different forms such as present "VeprenTF" (see Table 5.4), past "VerPastFile", future "VerbfFile", passive "VerPasFile", imperative "VerbImpFile", past participle "PastPtFile", and continuous "ContFile"
Auxiliary verbs	An Auxiliary verb can take three different forms: present "AuxPrFile", past "AuxPaFile", or future "AuxPfFile".
Pronouns	There are three types of Arabic pronouns: Subjective "SubjProFile", Objective "ObjProFile"and Possessive pronouns "PossProFile"
Negation	There are many words that are used to form negation "NegaFile" in the Arabic language such as "Laysa" (In Arabic: ليس).
Demonstratives	Demonstratives "DemoFile" (in Arabic: pointing nouns اسماء الاشارة) are words that indicate which entities a speaker refers to (e.g. this – hatha; In Arabic: هذا)
Prepositions	Arabic Prepositions "PrePosFile" come before the noun to introduce a prepositional phrase.
Adverbs	Arabic Adverbs "AdeVrFile" (in Arabic: Haal ظرف . حال) are used to modify a verb, an adjective, or another adverb
Question form	Arabic language question form "QuesFoFile" contains various question words used in the question formation such as Where (In Arabic: aena أين) and How (In Arabic: keafa كيف)
Adjective	Arabic adjective " AdjecFile" is a word used to describe or qualify a noun or pronoun
Comparison	Comparison or equality in Arabic are expressed by adding the word 'similar'/'like' (In Arabic: methel مثل) or the prefix "ka")In Arabic: ka ك) to the object or person compared with. Comparison takes three forms: first "CompFfoFile", second "CompSfoFile" and similarity "CompSmfoFile" forms.

Table 5.3: The language components of the DK

5.3.2 Additional components of the DK

This section presents other components used by the tutoring system to store additional teaching materials such as reading passages and expressions. These components are summarised in Table 5.5.

	Form	VS	VS1	Form	VS	VS1	Form	VS	VS1
1	**IND**	**1**	**2**	**spsf**	تلعبي	تجري	**AUD**		
2	**Verb**	**2**	**1**	**PIC**			**tpdf**	تلعبان	تجريان
3	**fpsm**	العب	اجري	**AUD**			**PIC**		
4	**PIC**	**-**	**-**	**spdm**	تلعبان	تجريان	**AUD**		
5	**AUD**			**PIC**			**tppm**	يلعبون	يجرون
6	**fpsf**	العب	اجري	**AUD**			**PIC**		
7	**PIC**			**spdf**	تلعبان	تجريان	**AUD**		
8	**AUD**			**PIC**			**tppf**	يلعبن	يجرن
9	**fpdm**	نلعب	نجري	**AUD**			**PIC**		

Table 5.4: The structure of the "VeprenTF" file and sample content

Components	Description
Reading	The "Reading" file "ReadFile" stores the reading materials provided by the tutoring system and their resources: such as pictures, audios and videos if available (see Table 5.6).
Reading Question	The reading material's questions and their answers are stored in the "ReQuFile" file (see Table 5.7).
Expressions	Arab's daily expressions are closely related to religion, from simple greetings to future planning. The expressions are stored in the "ExprFile" file
General	The "General" file "GenerlFile" contains general introductory learning materials about lessons such as the Arabic verbs, adjectives, and the articles (see Table 5.8).

Table 5.5: The additional components of the DK

IND	RP	ENG	NL	PICT	AUD	VID
1	1	The camel is a friendly, hard working, and intelligent animal.	الجمل حيوان لطيف ويعمل بجد وذكي			
1	2	The camel has ability to go days without water.	الجمل لديه القدرة أن يعيش عدة ايام بدون ماء			
1	3	The Arab use camel to travel across the desert.	العرب يستخدمون الجمل في التنقل عبر الصحراء			

Table 5.6: The structure of "ReadFile" file

IND	RP	RQ	RQFA	RQSA	RQTA	RQRA
1	1	The Arab use the camel for?	Travelling	Ploughing	Racing	hunting
1	2	The camel has ability to go days without?	Sitting	Resting	Sleeping	Drinking

Table 5.7: The structure of "ReQuFile" file

IND	Title	ENG	NL	NL1	NL2	Text	PIC	AUD	VID
1	Arabic numbers	-	-	-	-	Arabic numbers is also called Hindu-Arabic numbers. This number system was used by Arab mathematicians since the 9th century AD.	-	-	-
2	The article	the	ال	-	-	In Arabic language any Nouns preceded by the definite article (ال) are definite.	-	-	-

Table 5.8: The structure of "GenerlFile" file

5.4   Summary

This chapter discussed the design of the system knowledge and its components such as the learning database and the domain knowledge. The learning database (LDB) stores information about the learner's knowledge state and personality. This information is used by the learner model and the instructional model during the learning process. A relational database with a single repository stored in a MySQL database is used to represent the LDB. The domain knowledge (DK) contains knowledge about the taught language (i.e. Arabic language) represented as a set of files. The design of the DK represents a new feature that lays the foundation for the pluggable domain concept. A general design the DK is implemented based on natural language's various narrative modes, gender, etc. This design can be utilised by the pluggable domain concept in the teaching of different languages.   The following chapter introduces the concept of the pluggable domain which allows for the teaching of a new language using the same system's templates.

CHAPTER 6: Pluggable Domain

This chapter introduces the concept of the pluggable domain which allows the system's templates to be used in the teaching of a new language. The French language is used as an example in the pluggable domain concept. Also the advantages and disadvantages of this concept are highlighted.

6.1   The Pluggable Domain

The system provides a pluggable domain based on the concept of a general design of the domain knowledge (DK) and a high level representation of that knowledge by the control of the system represented by the Course manager (CM). This is done through the separation of the CM and the DK. The challenge is to build a DK design that guarantees the availability of certain teaching units (*TUs*) such as characters, words, or audio from the newly plugged language.

The concept of the pluggable domain combined with the design of the DK presents one of the contributions of this research since there is no authoring tools yet developed for authoring the human languages. This is because authoring natural languages is more difficult due to the differences between the languages structures and grammars. As a result the concept of the pluggable domain is developed to allow teachers to create a new language learning environments from an existing one. This has the advantage of making such a process cheaper, faster, and easier.
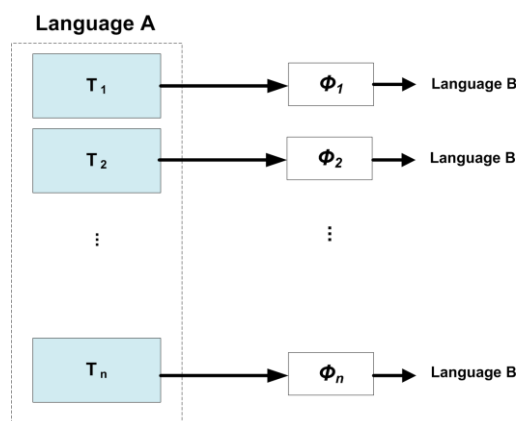
Figure 6.1: The conversion processes

The tutoring system contains templates with different formats for Lessons, Practices, Extra tutoring, and Questions. Each template can have text, audio, picture, and video in its contents. Given a language A (e.g. Arabic) and a language B (e.g. French), a template T, and a conversion processes Φi which converts the teaching materials of the template T from language A to language B based on the contents and format of T. This facilitates the introduction of the new language materials into the same template T (see Figure 6.1).

6.2   System templates

The system templates are basically web pages that are connected to the DK via PHP/MySQL queries. Various templates are developed for this system that includes: lessons (L), practices (P), extra tutoring (ET), questions (Q), and examinations (E) templates. Each template may contain various TUs such as texts, audios, pictures, or videos. These templates are stored in the learning database while their knowledge is retrieved from the DK.

For the instance presenting the lesson template (e.g. teaching Arabic characters) before loading the knowledge form the DK produces an empty web page. Whereas after loading the knowledge form the DK the web page presents a template with different TUs that include text, picture, audio, and video (see Figure 6.2).
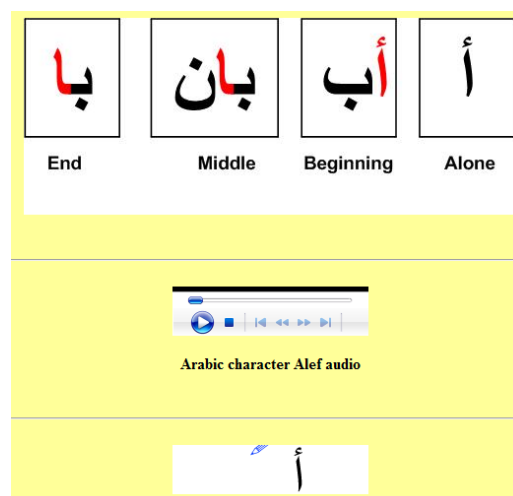
Figure 6.2: Lesson template after loading the knowledge from the KD

6.3   The conversion process

The conversion process (Φi) is the process of mapping certain TUs of a source language (e.g. Arabic) into the target language (e.g. French). The conversion process is a content-based process which means changing the contents of each template T requires the modification of Φi of that template. The Φi includes the task of deciding which knowledge of the target language is the most appropriate equivalent of the source language *TU.* In other words Φi represents the differences between the source language and the target language*.* In addition, the target language may specify additional equivalents that differ mainly in their usages (e.g. the Arabic language article "ال" pronounced '*al'*, and "the" in the English language, have two equivalents in the French language "*el*" and "*al*"). Therefore the design of the DK must support all these equivalents.

The mapping process can be one-to-one in which a *TU* in the source language is replaced by the same *TU* in the target language without any change to the content of the template (i.e. no Φi needed). In other words after replacing the source language DK with the target language DK the *TU* can still convey proper information. Examples of the one-to-one mappings of different *TUs* are presented in Figure 6.3.
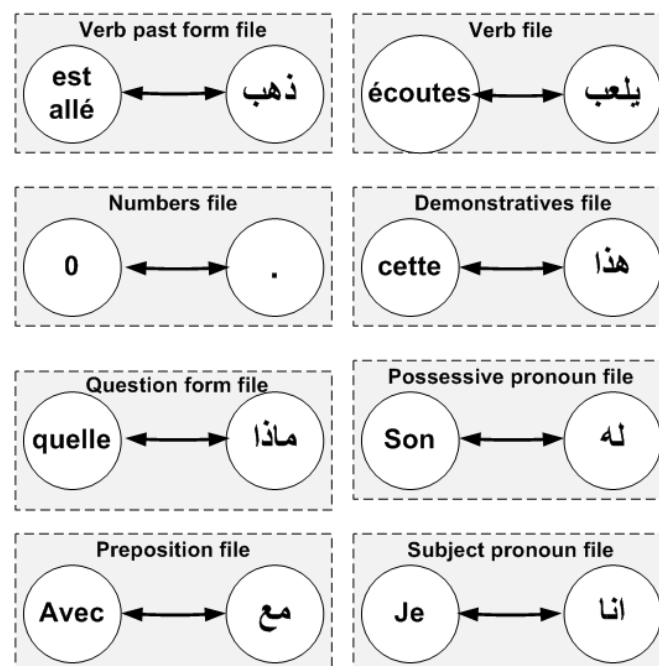
Figure 6.3: Examples of one-to-one mappings between Arabic and French

92

Pictures and videos can also have one-to-one mappings between the source and target languages. Figure 6.4 presents an example of two pictures (ولد: in English Boy; and مفتاح : in English Key).
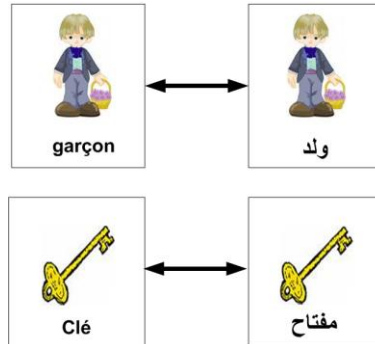
Figure 6.4: Examples of one-to-one mappings of pictures between Arabic and French

Each language has its own set of characters and some of these characters in Arabic do not exist in other languages and vice versa. Consequently there is no one-to-one mapping between the source and the target language for characters. Throughout the teaching of the characters' lessons various nouns and verbs are used in the learning process. Therefore, two index files are designed to link these nouns and verbs with certain characters. The file "charNounFile" is designed to link each character with certain nouns (see Table 6.1). The file "charVerbFile" is designed to link each character with certain verbs (see Table 6.2). These files are used to link each character with certain verbs and nouns that contain that character. These files are used by the system's templates to explain characters through various nouns and verbs contain those characters.

Field	Description
IND	Index
CI	Character index
NI	Noun index

Table 6.1: The description of "charNounFile" file

Field	Description
IND	Index
CI	Character index
VI	verb index

Table 6.2: The description of "charVerbFile" file

Arabic language characters are written in "Unicode" format. The list of all Arabic characters and their Unicode format is shown in Table B.1 (see Appendix B).

For a more complicated language construction like sentence building, the conversion processes *Φi* are needed to be included in the design of the template. The conversion process includes tasks like changing TUs order or adding an auxiliary verb. For example, consider the following Arabic sentence "الولد يلعب بالكرة", which corresponds to the English sentence: "The boy plays with the ball" (see Figure 6.5). It can be seen in the source Arabic sentence, the article "الـ" (in English: the) is combined with the nouns "ولد" (in English: boy) and "كرة" (in English: ball). The preposition "ب" (In English: with) is also combined with both the article "الـ" and with the noun "كرة". The Arabic words are principally written by linking or connecting the characters together (Al-Neaimi, *et al.*, 2009). Unlike the Arabic language, the French language has various articles' formats (e.g. the article "le" for masculine things/persons and the article "la" for feminine objects/persons) and it is not connected to the noun. Therefore the conversion process *Φi* of the above sentence into a French language equivalent ought to perform the following tasks:

1. Choose the proper article for the sentence.
2. Change the orientation of the sentence for left-to-right into right-to-left.
3. Dismantle the articles from the nouns.
4. Dismantle the preposition from the nouns.
5. Place the proper article for each noun.
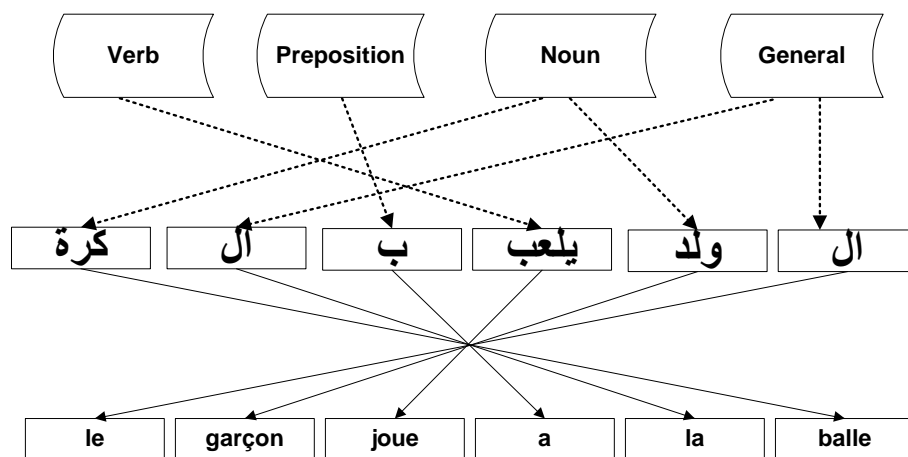
Figure 6.5: The conversion process of the sentence "The boy plays with the ball" form Arabic to French

Another example is the Arabic sentence "الحداد ذهب الى الجبل", which corresponds to the English sentence: The blacksmith went to the mountain. The conversion of this sentence is shown in Figure 6.6.
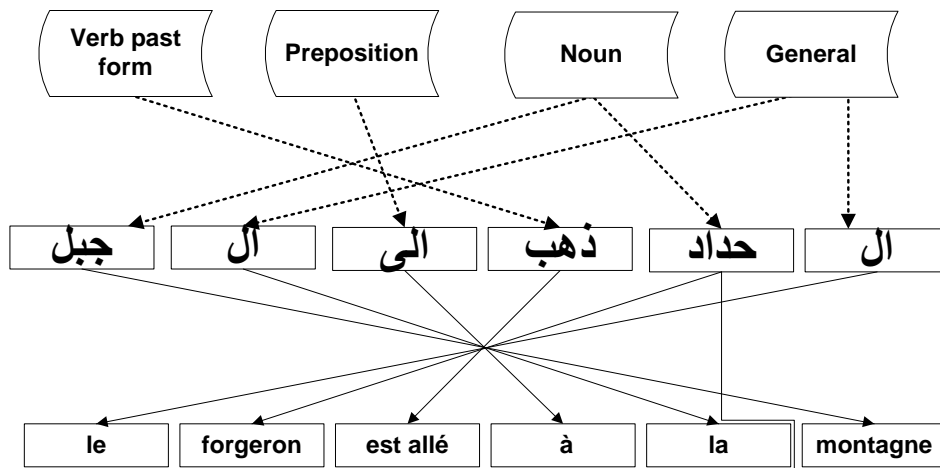
Figure 6.6: The conversion process of the sentence "the blacksmith went to the mountain" from Arabic to French

The conversion process of the above sentence into a French language equivalent ought to perform the following tasks:

1. Choose the proper article for the sentence.
2. Change the orientation of the sentence for left-to-right into right-to-left.
3. Dismantle the articles from the nouns and the verb past.
4. Place the proper article for each noun.

Finally the Arabic sentence " السماء صافية", which corresponds to the English sentence: "The sky is clear". The conversion process of this sentence is shown in Figure 6.7.
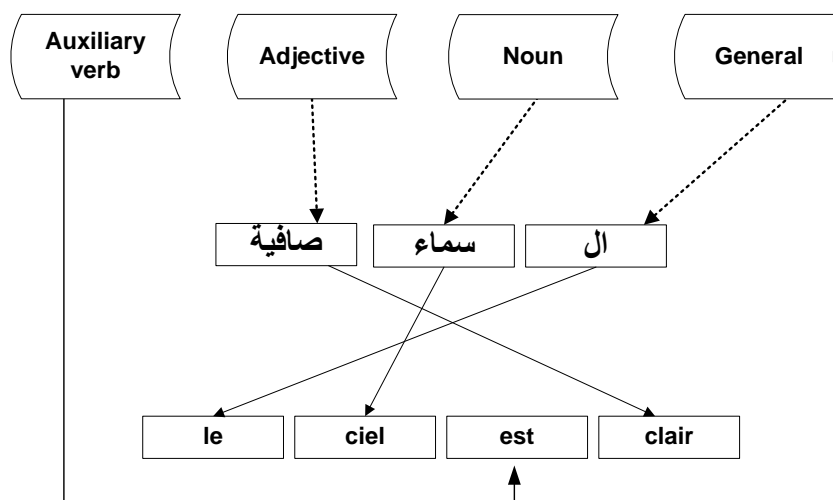
Figure 6.7: The conversion process of the sentence "the sky is clear" from Arabic to French

The conversion process of the above sentence into a French language equivalent ought to perform the following tasks (see Figure 6.7):

1. Choose the proper article for the sentence.
2. Change the orientation of the sentence for left-to-right into right-to-left.
3. Dismantle the articles from the nouns.
4. Place the proper article for the noun.
5. Add auxiliary verb.

6.4   Reading materials

The proposed system provides short reading materials (i.e. passages) to the learner to help him/her in reading and vocabulary learning. These passages stores about one to four lines in length, covering a wide variety of topics related to the taught language culture (e.g. The Arabic world). The "ReadFile" file contains all the reading passages and the "ReQuFile" file stores all the passages' questions and answers. The structure and sample of the contents of the files "ReadFile" and "ReQuFile" are shown in Tables 5.6 and 5.7 respectively.

6.5   Advantages and disadvantages of the pluggable domain

The design of the pluggable domain can provide the following advantages:

- The system's resources (texts, pictures, audio, and videos) are stored once and used many times, i.e. save storage space.
- Very easy to modify or replace certain teaching materials since the knowledge (e.g. Boy) is stored in the domain knowledge (DK) while the system templates only provide links to the DK. Changing the contents of the DK (e.g. Garçon) affect the knowledge presented in all the system templates linked to those contents.
- Reduce the time and effort required to build a new system by using the same templates for teaching different languages. This is done by changing the conversion process Φi for the newly introduced language.

- Based on simple concept compared to other techniques such as NLP and machine translation. The teaching of colloquial languages such as Egyptian Arabic or Syrian Arabic can be implemented using this concept.

The disadvantages of this approach can be summarised as follows:

- This approach is restricted to the system templates only (template-based). Therefore any new templates must have an associated conversion process written and linked to the DK files.
- Changing the contents of each template T require the modification of the conversion process of that template since this process is a content-based process.

6.6   Summary

This chapter has presented the concept of the pluggable domain which is developed with the intention to allow teachers to create a new language learning environments from an existing one by reusing the system's templates. Various templates are developed for this system that includes: lessons, practices, extra tutoring, questions, and examinations templates. Each template may contain various texts, audios, pictures, or videos. These templates are stored in the learning database while their knowledge is retrieved from the DK. Each template contains a conversion processes that converts the teaching materials of certain template from one language to another. Finally the advantages and downsides of the pluggable domain is presented and discussed. The concept of the pluggable domain is another central contribution of this work. This concept is based on a general design of the domain knowledge (DK) and a high level representation of that knowledge by the system control represented by the Course Manager (CM). The concept is based on the reusability of the system's templates and a conversion process that provides mapping between the source and target languages. The system is implemented using PHP scripts, and MySQL database running on an Apache server. The next chapter discusses the implementation of various components of the tutoring system.

CHAPTER 7: IMPLEMENTATION AND TESTING

This chapter presents the implementation of the tutoring system and its components such as the learner model and the instructional strategies. Data-flow diagrams are used to provide graphical representations of the flow of data.

7.1   System Implementation

The implementation of the system components such as the learner model, instructional strategies, user-interface and modes of interaction with the tutor system is presented in this chapter. The system is implemented using PHP scripts, and MySQL database running on an Apache server. The communication between the user-interface and the server is based on exchanged HTTP requests and responses. The design provides support for multiple users on an arbitrary number of concurrent sessions. Each individual session is established when learner login to the system.

7.2   Fuzzy inference

The fuzzy inference (i.e. Learner model) is implemented as the function "Fuzzy" which takes two crisp inputs and produce one fuzzy class that represents certain learner's stereotype. The function Fuzzy calls three functions: Fuzzyify, Defuzzy and Classify. The function Fuzzyify performs the fuzzification process of the crisp inputs. The function Defuzzy performs the defuzzification process. The function Classify produces the final class (i.e. Stereotype). The function Fuzzy is shown in Figure 7.1.

```
Function Fuzzy (First input, Second input)
{
Collect information from the file FuzzyInfo.
Collect information from the file FuzzyRule.
Fuzzify().
Defuzzy();
Classify();
}
```

Figure 7.1: The structure of the function Fuzzy

The fuzzy membership functions and their rules can be reconfigured by the teacher via the teacher-interface. The initial values for each fuzzy inference function and its rules are stored in the files "FuzInfFile" and "FuzzRulFile" respectively. The structure of these files is shown in Tables 4.1 and 4.2 respectively.

As discussed in Section 4.2.3 the classification variables (i.e. W1 to W5) are retrieved from the "DefuFile" file and used in the fuzzy inference. The initial values of the "DefuFile" file are shown in Table 4.3. These values can be reconfigured by the teacher via the teacher-interface.

7.3   Selecting instructional strategy

The process of modifying and selecting the proper instructional strategy is based on three steps. These steps are implemented as follows:

1. Connecting a certain set of the learning materials' input attributes to a certain instructional strategy. The attributes of each strategy are stored in the "StratFile" file. The structure and the initial values of the "StratFile" file are shown in Table 3.2.

2. Generating a new set of patterns each with 5 inputs (attributes): learning level, difficulty, time, category and type, and one output (instructional strategy) as described in Section 4.3.2. All the generated inputs and outputs are stored in the "InstStratFile" file. These processes are carried out by the teacher via teacher-interface.

3. Selecting the proper instructional strategy (InstStr) is based on the collected combination of attributes for the current learning materials (i.e. Lesson or Question). The variable xi represents certain input attributes (i.e. learning level, difficulty, time, category and type). This process is implemented by the MySQL query shown in Figure 7.2.

$query="SELECT InstStr *from InstStratFile WHERE* `learning level`=x1 *and* `difficulty`=. *and* `time`=x3 *and* `category`=x4 *and* ` type`=x5"; $result = mysql_query ($query,$conn) die(mysql_error());

Figure 7.2: MySQL query for selecting the proper instructional strategy

Moreover the features comprising each instructional strategy are stored in the "FeatFile" file (see Table 3.5). These values can be reconfigured via the teacher-interface.

7.4   User-interface

The user-interface plays a major role in the presentation and interaction between the learners and the system. A graphical user interface (GUI) is designed to be user-friendly for the teachers and learners alike. However the GUI is a prototype and the Human-Computer Interaction principles were not essential as the research is fundamentally looking at the design features. The tutoring system has two user-interfaces: *the learner-interface* and *the teacher-interface*. The GUI is also used to display pictures, audio, and video. Moreover the GUI is used as a browser for teaching units information.

Hello Dawod

| Browse previous Lessons | Transcript | Exit to the Main window |
| Start learning |
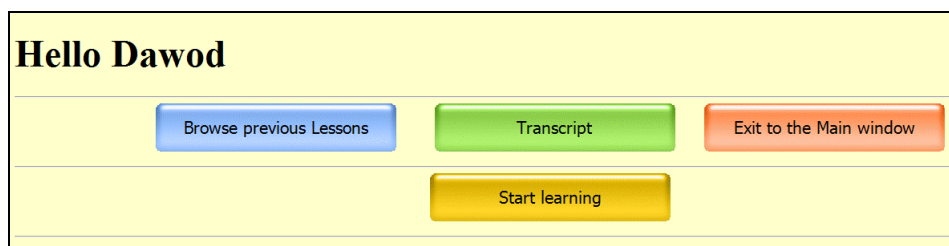
Figure 7.3: A screen of the main window of the learner-interface

The learner-interface (LI) provides various teaching materials, feedback, and explanations to the learner. A screen of the main window of the learner-interface is shown in Figure 7.3. The teacher-interface (TI) allows the teacher to reconfigure various components of the system such as the learner model and the instructional strategies. Moreover, the teacher can customise (i.e. add, delete and/or modify) the contents of the system's templates and the domain knowledge. The teacher logs to the TI using a special teacher ID.

7.5   Learner interaction with the system

The LI provides interactive components that help learners to study the teaching materials during the learning process. The LI facilitates two modes of learning: interactive and discovery based learning. The LI can also be used to display the learner's transcript any time during the learning process.

7.5.1   Interactive-based learning

In this mode the system presents the current lesson and its components such as Prerequisites, Practice, Extra tutoring, and Questions according to the selected instructional strategy. The learner controls when to access each of these components. The implementation of the lesson and its components are discussed in the next sections. The data-flow diagram (DFD) of the Learner interaction with the Interactive-based learning is shown in Figure 7.4.
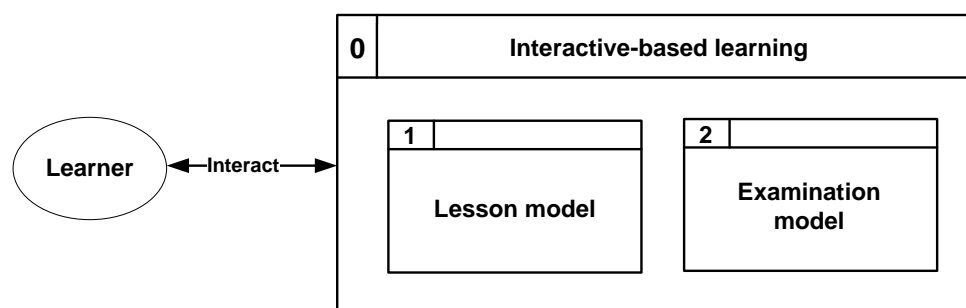
Figure 7.4: The DFD of the Learner interaction with the Interactive-based learning

7.5.1.1   Lesson model

The Lesson model is responsible for presenting the current lesson and its components. The DFD of the Lesson model and its components are shown in Figure 7.4. A screen of the Lesson model is shown in Figure 7.6. The lesson-interface presents the contents of the current lesson and the knowledge of the selected instructional strategy in an understandable format aimed at guiding the learner during the learning process. It also provides a set of control buttons that allow the learner to study the lesson's components as well as provide explanations and system's help. Various progress indicator icons such as the learner's learning level, progress in the current learning level, and performance are provided by the lesson-interface. In addition the lesson-interface provides a Comments editor icon through which the learner can write any comments during the learning process.
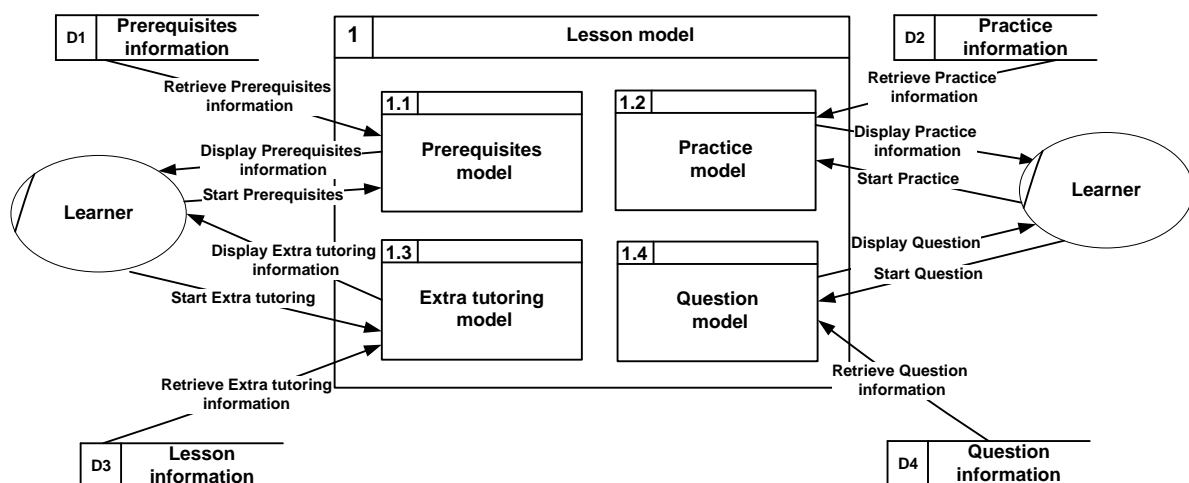
Figure 7.5: The lesson model DFD

102

LLID=13

No							No						
4	ث	ثثث	ـث	ـثـ	ثـ	Thaa	20	ف	ففف	ـف	ـفـ	فـ	Faa
5	ج	ججج	ـج	ـجـ	جـ	Jeem	21	ق	ققق	ـق	ـقـ	قـ	Qaaf
6	ح	ححح	ـح	ـحـ	حـ	Ha	22	ك	ككك	ـك	ـكـ	كـ	Kaaf
7	خ	خخخ	ـخ	ـخـ	خـ	Kha	23	ل	للل	ـل	ـلـ	لـ	Laam
8	د	ددد	ـد	ـد	د	Daal	24	م	ممم	ـم	ـمـ	مـ	Meem
9	ذ	ذذذ	ـذ	ـذ	ذ	Dhaal	25	ن	ننن	ـن	ـنـ	نـ	Noon
10	ر	ررر	ـر	ـر	ر	Raa	26	ه	ههه	ـه	ـهـ	هـ	Haa
11	ز	ززز	ـز	ـز	ز	Zaa	27	و	ووو	ـو	ـو	و	Waaw
12	س	سسس	ـس	ـسـ	سـ	Seen	28	ي	ييي	ـي	ـيـ	يـ	Yaa
13	ش	ششش	ـش	ـشـ	شـ	Sheen							
14	ص	صصص	ـص	ـصـ	صـ	Saad							
15	ض	ضضض	ـض	ـضـ	ضـ	Daad							
16													

It is mendatory to study the Practice

It is mendatory to study the Prerequisites

STUDYING the extra tutoring is Optinal for you

You must study the Questions (after the other required teaching materials)

| Prerequsites | Practice | Question | Extra tutoring |

| Continue learning | Exit |

Performance   Beginner Level   Progress 10 %

Figure 7.6: A screen of the Lesson model

7.5.1.2   Practice model

The Practice model is responsible for presenting the lesson's practice. The DFD of the Practice model is shown in Figure 7.5. The practice interface (PI) provides a short overview of the lesson's materials, a practical interaction with the system using questions and analysis of the learner's answers along with corrective feedback. The analysis is based on the learner's previous and current interactions with the Practice model. This section is activated only when the learner clicks on the "feedback" button.

7.5.1.3   Extra tutoring model

The Extra tutoring model provides extra teaching materials related to the current lesson. The DFD of the Extra tutoring model is shown in Figure 7.5. The pathname

of the lesson's extra tutoring is stored in the "Lesson" file. Based on the previous interactions with the system, some learners are required to go through extra tutoring to enhance their understanding of the taught materials (i.e. as a remedy to the weak and below average learners).

7.5.1.4 Pre-requisites model

The Pre-requisites model presents different lessons which precede the current lesson. The DFD of the Pre-requisites model is shown in Figure 7.5.

7.5.1.5 Question model

The Question model provides a set of questions to test the learner's understanding of the presented lesson. The DFD of the Question model is shown in Figure 7.5. A screen of the Question model is shown in Figure 7.7. The learner must answer these questions before moving to the next lesson. When the learner clicks on the "submit" button the system presents answers, corrective feedback and analysis of the answers. The question model provides three control buttons: feedback, hint and explanation. The "Feedback" button displays help information about the current questions. The "Hint" button provides hints about the current questions. The "Explanations" button displays explanations to the learner. A pop-up window is used to present the contents of the feedback, hints, and Explanations so the learner can simultaneously work with the original window and the additional one.

(2) Which of the following is nominal Sentence ?

ذهب الطالب المدينة الى

الطالب ذهب الى ال مدينة

المدينة الطالب ذهب الى

المدينة الى الطالب ذهب

(3) Choose the verb from the following sentence ?

الطالب ذهب الى المدينة

الطالب

ذهب

Figure 7.7: A screen of the Question model interface

7.5.1.6   *Examination model*

The Examination model provides a level examination after the completion of each learning level in addition to the placement test examination. The learning level examinations have the purpose of evaluating the learner knowledge of each learning level. The system consequently presents the learning level examination after the learner completes each learning level. The DFD of the Examination model is shown in Figure 7.8. The placement test aims at assessing the approximate level of a learner's knowledge of the taught language before he/she starts the course. The examination information is stored in the "ExamFile" file. The structure of this file is shown in Table 7.1.

Fields	Description
ExamID	Exam ID
Title	Exam title
Time	Exam time (min)

Table 7.1: The structure of the "ExamFile" file
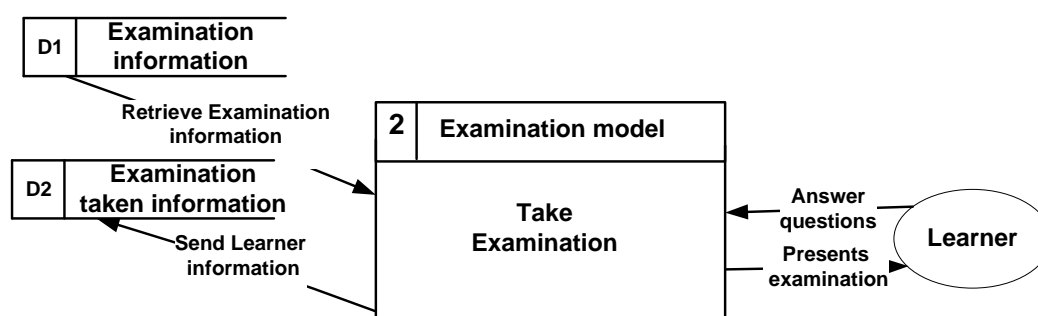
Figure 7.8: The DFD of the Examination model

7.5.2   **Discovery based learning**

This mode provides unconstraint environment to the learner to study the previous lessons. The discovery based learning interface provides navigation buttons (i.e. "Previous lesson" and "Next lesson") that can be used by the learner to go through the lessons and their contents. The learner can navigate as far back as the first lesson and as forward as the current less. All of the learner's logins with this mode are stored in the file "DiscLogFile" (see Table 4.8). This file is used in the calculation process of the discovery average time (DAT) as discussed in Section 4.3.4.

7.5.3　　　　　Learners' Questionnaire

The learner is required to fill a learners' questionnaire after the completion of his/her final examination. The questionnaire contains 42 questions as discussed later on in this thesis. The learners' responses to the questionnaire's questions are stored in the "LeaQuesFile" file. The structure of the "LeaQuesFile" file is shown in Table 7.2. The full list of the learners' questionnaire's questions is shown in Table A.1.

Fields	Description
LID	Learner ID
A1	The learner's response question number 1
A2	The learner's response question number 2
A3	The learner's response question number 3
A4	The learner's response question number 4
:	:
A41	The learner's response question number 41
A42	The learner's response question number 42

Table 7.2: The structure of the "LeaQuesFile" file

7.6　Teacher-interface

The teacher-interface provides various controls which help the teacher to manage the learning, presents comprehensive statistics, reconfigure the courseware, and view the system's listener logs containing the contents of each learner's sessions. The DFD of the teacher-interface controls is shown in Figure 7.9. These controls are discussed in the next sections. The general structure of the teacher-interface is shown in Figure 7.10.
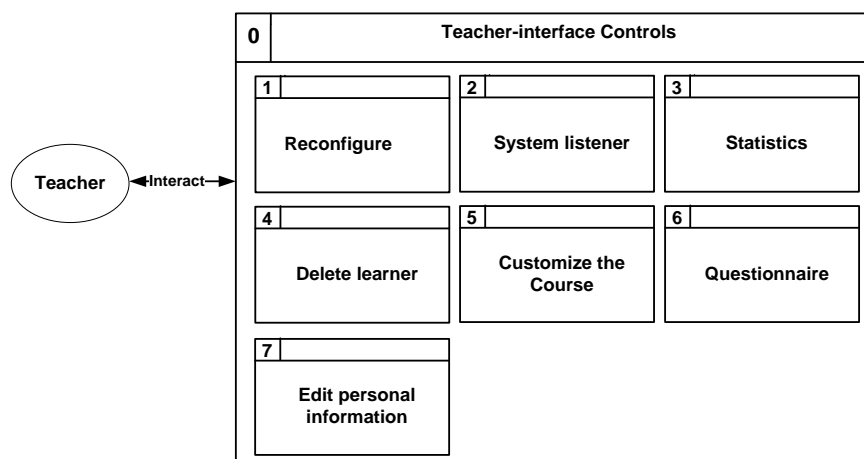
Figure 7.9: The DFD of the teacher-interface controls

106
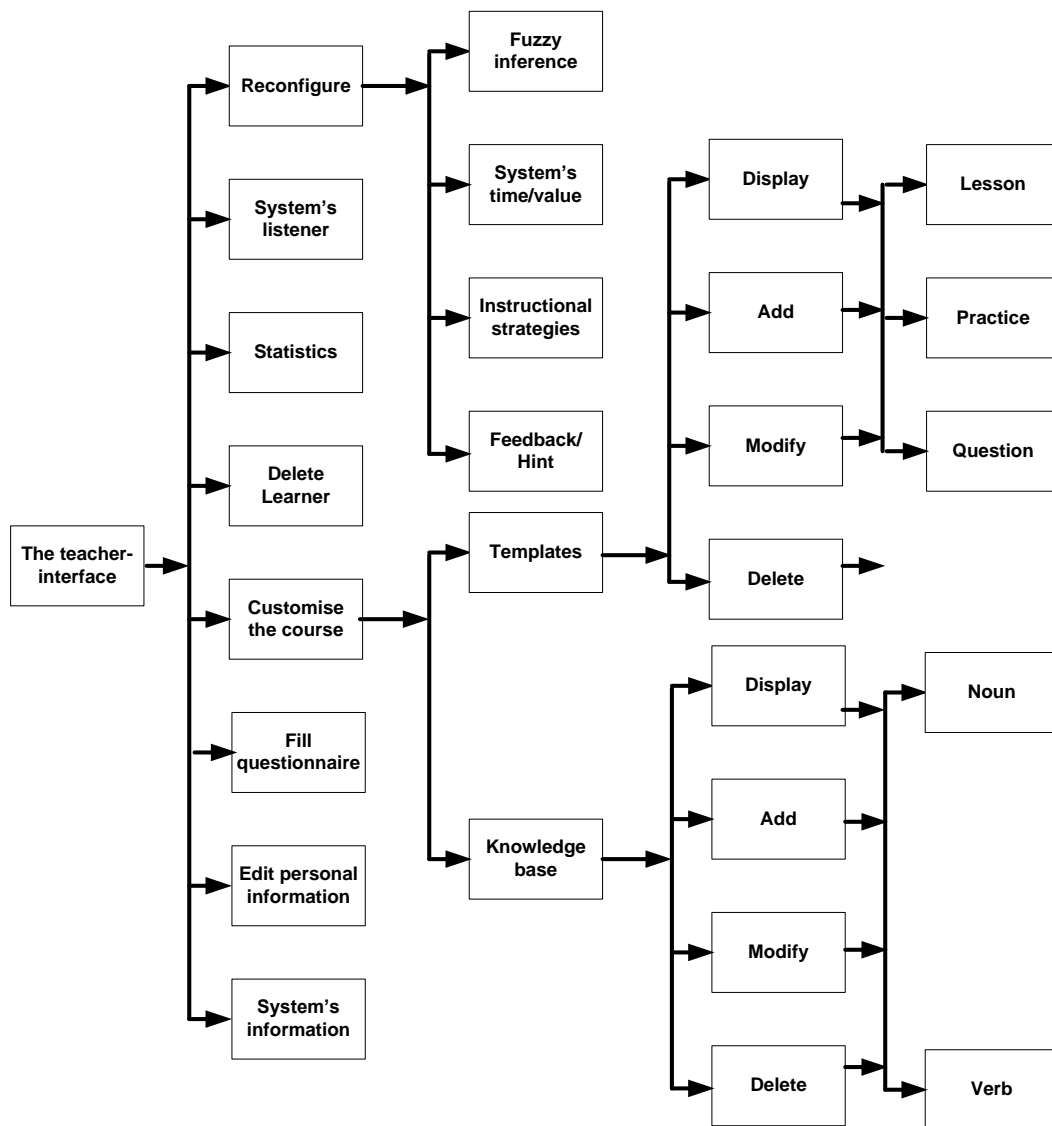
Figure 7.10: The general structure of the teacher-interface

7.6.1 Reconfiguration control process

The reconfigure control contains several functions that allow the teacher to reconfigure various components of the system. The Reconfigure page provides four controls: "Fuzzy inference", "Instructional strategy", "system time/values", and "feedback and hinting". The DFD of the reconfigure control is shown in Figure 7.11.
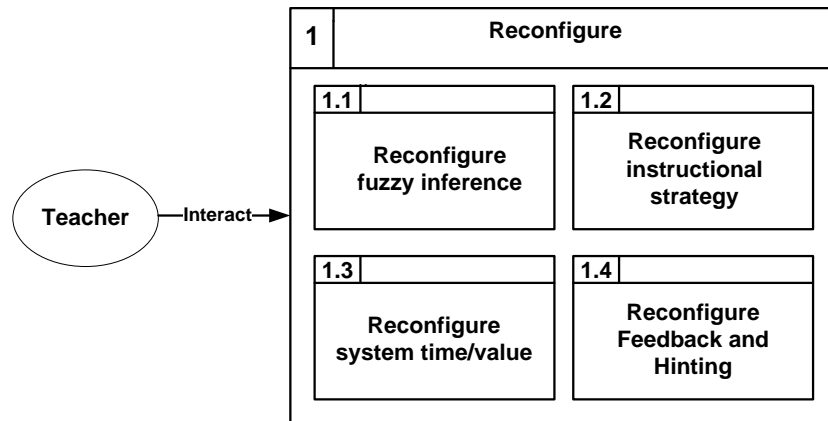
Figure 7.11: The teacher-interface adjust/modify control DFD

The "Fuzzy inference" control allows the teacher to modify the fuzzy inference engine and its components. The DFD of the "Fuzzy inference" control is shown in Figure 7.12. The "Fuzzy inference" web page contains three controls: "Membership functions", "Rules", and "Defuzzification".
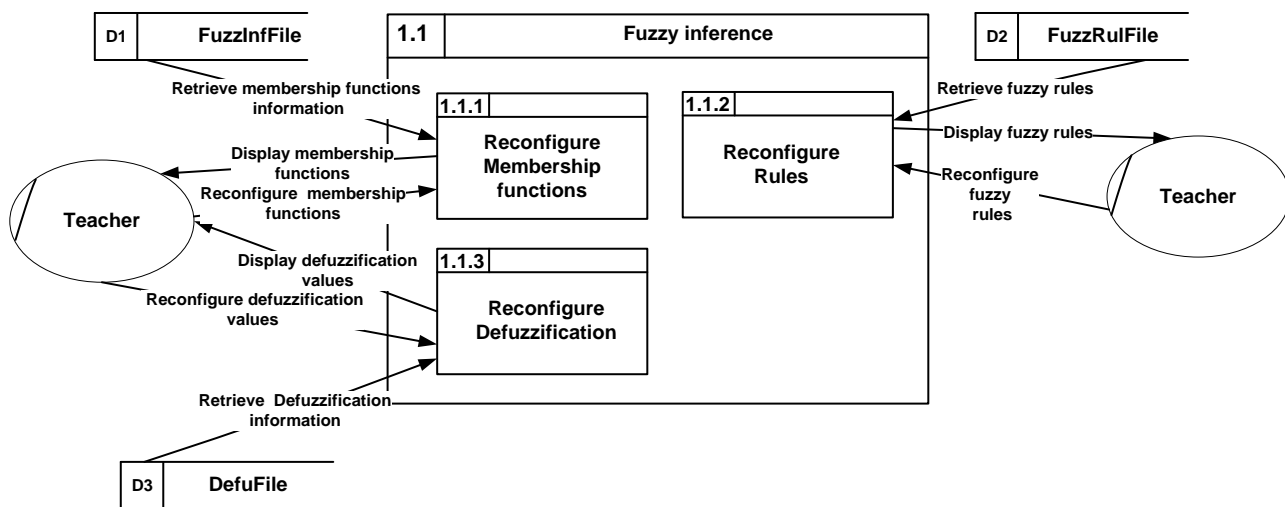
Figure 7.12: The fuzzy inference control DFD

The "Instructional strategy" control allows the teacher to reconfigure the Instructional strategies (IS) and their components. The DFD of the "Instructional strategy" control is shown in Figure 7.13. This control allows the teacher to link each instructional strategy with certain inputs, generates an associated instructional strategy, and modifies the features of each instructional strategy.
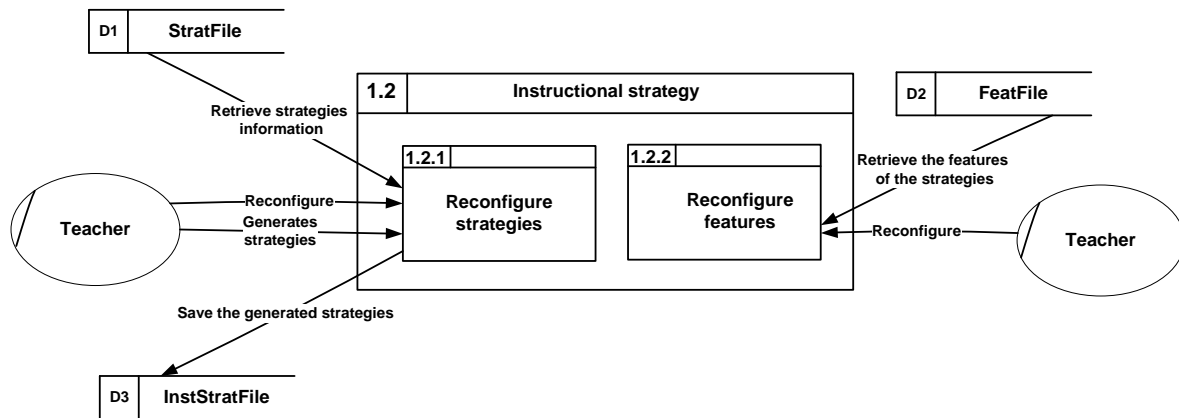
Figure 7.13: The Instructional strategy control DFD

The "System time/values" control allows the teacher to modify the minimum and maximum values that are used in the calculation of the learner's engagement as described in Section 4.2.6.3. In addition this control allows the teacher to modify the minimum and maximum discovery time that are used in the calculation of discovery average time (DAT) as described in Section 4.3.4. These values are stored in the "EngagFile" file. This web page also allows the teacher to modify the start grade of the first and second classes that are used by the Administrator model as described in Section 4.3.1. These values are stored in the "PlacFile" file. The DFD of the "System time/values" control is shown in Figure 7.14.
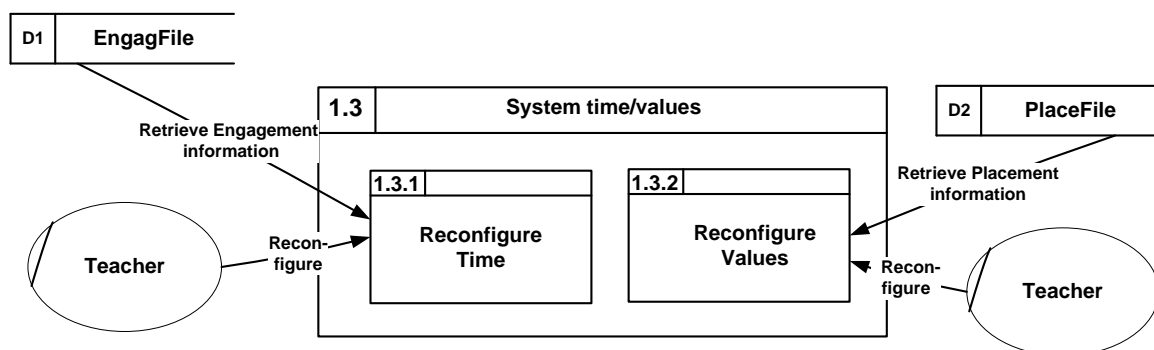
Figure 7.14: The "System time/values" control DFD

The "Feedback and Hinting" link allows the teacher to modify the feedback and hinting strategies as described in Section 4.2.3. The DFD of the "Feedback and Hinting" control is shown in Figure 7.15.

109
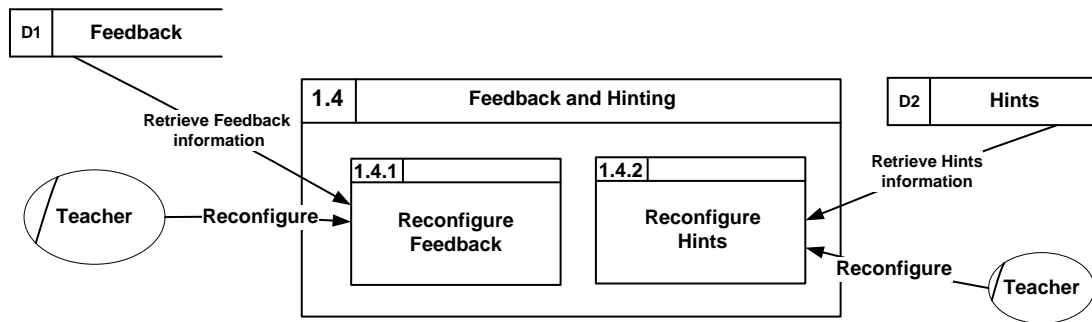
Figure 7.15: The "Feedback and Hinting" control DFD

7.6.2 System listener

The "System listener" control displays the contents of the learner's log file. The log file is a text file that records all the learners' actions and the system's responses. The system decodes all these actions and presents them in a format understandable by the teacher. A screen of the system listener is shown in Figure 7.16. In order to increase the readability of the log file each row i of the file is numbered as "Line i".

Figure 7.16: A screen of the system's listener

7.6.3　System's statistics

This web page contains two statistics: "General and Personalised statistics". The General statistics provides General statistics about the learners. The Personalised statistics allows the teacher to adjust his/her own profile by selecting different attributes about the learners (e.g. gender and learning level).

7.6.4　Delete learner

The "Delete learner" control allows the teacher to delete any learner and all their previous interactions from the learning database. This process is shown in Figure 7.17.
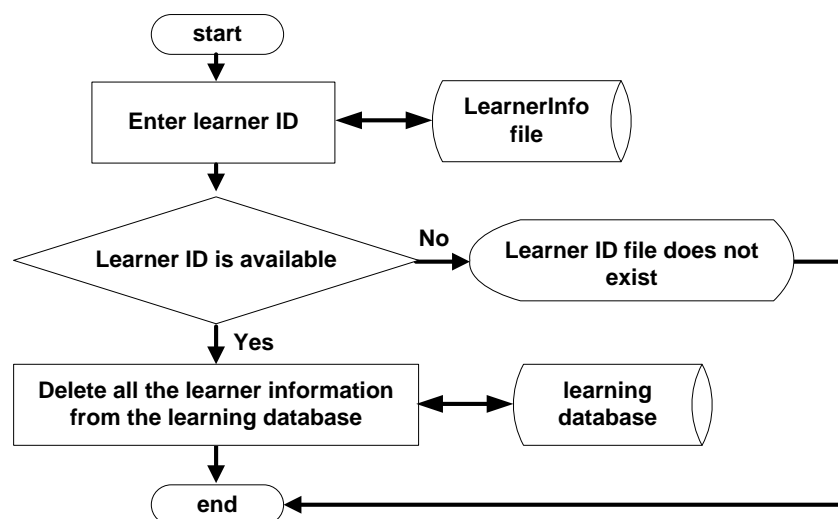
Figure 7.17: The "Delete the learner" process

7.6.5　Customisation of the course

The customisation function allows the teacher to add or modify system's templates or the contents of the domain knowledge. The DFD of the "Customise the course" control is shown in Figure 7.18.
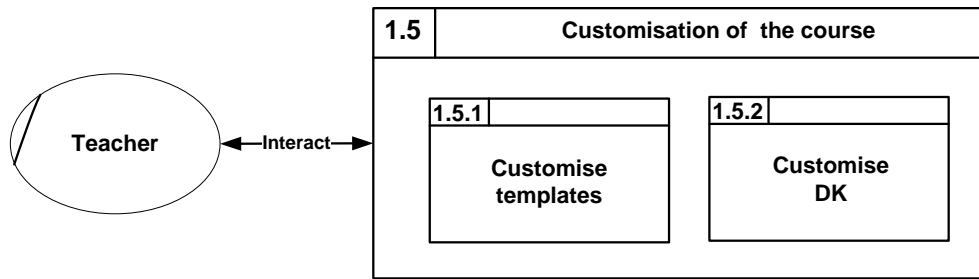
Figure 7.18: The Customise the course control DFD

7.6.6 Teacher' questionnaires

The teacher can fill a questionnaire by clicking the button "questionnaire". The questionnaire contains 44 questions. The teacher's responses to the questionnaire's questions are stored in the "TeachQuest" file. The structure of the "TeachQuest" file is shown in Table 7.3. The full list of the teachers' questionnaire's questions is shown in Table A.2.

Fields	Description
TID	Teacher ID
A1	The teacher's response question number 1
A2	The teacher's response question number 2
A3	The teacher's response question number 3
A4	The teacher's response question number 4
A5	The teacher's response question number 5
:	:
A44	The teacher's response question number 44

Table 7.3: The structure of the "TeachQuest" file

7.7 Pluggable domain tools

Various tools are developed in this approach for handling different inputs and outputs to facilitate the transformation process. These tools are discussed in the following sections.

7.7.1   Text orientation handling

The orientation of the Arabic language text is right-to-left. Therefore an HTML direction attribute "DIR" is embedded in the structure of the PHP script to specify the direction of the template's text when needed, or sections of the text: left to right (ltr) or right to left (rtl). This process is shown in Figure 7.19.

```
              ┌─────────┐
              │  start  │
              └─────────┘
                   │
                   ▼
          ◇─────────────────◇    Yes    ┌──────────────────────────────────┐
          Taught language is Arabic ──────▶│ echo"<p align='right' dir='rtl'>"; │
          ◇─────────────────◇            └──────────────────────────────────┘
                   │ No                                       │
                   ▼                                          │
          ◇─────────────────◇    Yes    ┌──────────────────────────────────┐
          Taught language is French ─────▶│ echo"<p align='left' dir='ltr'>";  │
          ◇─────────────────◇            └──────────────────────────────────┘
                   │ No                                       │
                   ▼                                          │
              ┌─────────┐◀──────────────────────────────────┘
              │   end   │
              └─────────┘
```
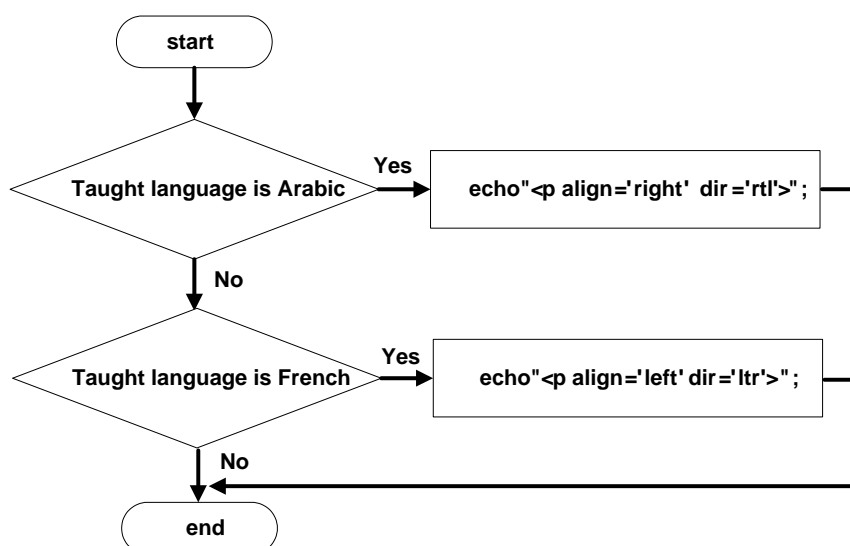
Figure 7.19: the process of changing the orientation of the text RTL or LTR.

7.7.2   Questions handling

The system provides two forms of questions: multiple-choice and short free-text response. The system handles the short free-text responses based on text matching. The learner answer is divided into elements by using the space character as a delimiter. These elements are then compared to the question answers elements that are stored in the file "QueFtextFile". The file structure is shown in Table 7.4. The text matching process of the answers is shown in Figure 7.20.

Fields	Description
QID	Question ID
Element	Identifies certain element in the question's answer
Answer	The answer for certain element

Table 7.4: The structure of "QueFtextFile" file
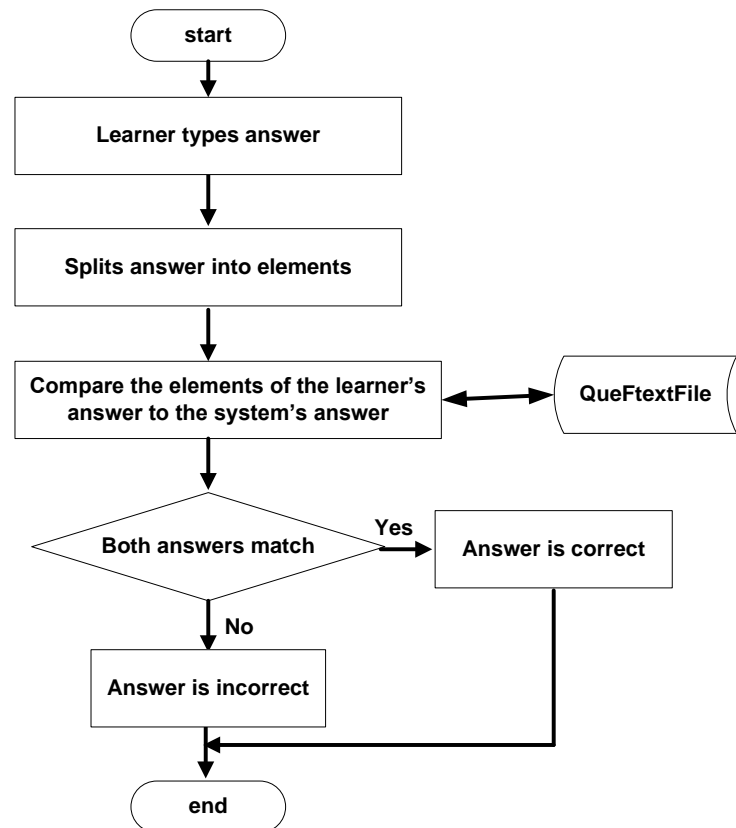
Figure 7.20: The process of handling the answers of free text question

7.7.3 Unicode character processing

This process is responsible for splitting any word into its characters (e.g. the Arabic word "student" "الطالب" becomes "s t u d e n t" "ب ل ا ط ل ا"). Since Arabic language is coded using Unicode format. For this, the function *"Characters"* is developed using PHP to split any word ($word) into its characters. The PHP script for the function *"Characters"* is shown in Figure 7.21.

```
function characters($word)
{
$character_array= array();
$character_array=explode(";",$word);
foreach ($character_array as $value) {echo " $value ";}
}
```

Figure 7.21: The PHP script of the function "Characters"

114

7.7.4   Loading audio/video files

The EMBED tag "<embed>" is used for loading audio/video to the system templates. In order to make the sentence dynamic the variable $AudVid is used in the "<embed>" tag. Based on the current template the variable $AudVid gets its information from the DK.

7.7.5   Loading image file

The image tag "<img>" is used for loading an image onto the system templates. In order to make the sentence dynamic the variable $PicT is used in the "<img>" tag. This variable gets its information from the DK.

7.7.6   Arabic keyboard

The system provides a virtual Arabic keyboard for Arabic letters. The learner writes text in Arabic language then "cut" and "paste" the text into the appropriate area.

7.7.7   Handling sentence audio

The system provides two ways of handling sentence audios. Firstly by collecting the audio components of each sentence and combine them together as one audio after adding 2 seconds gap between these audios. This process is shown in Figure 7.22.
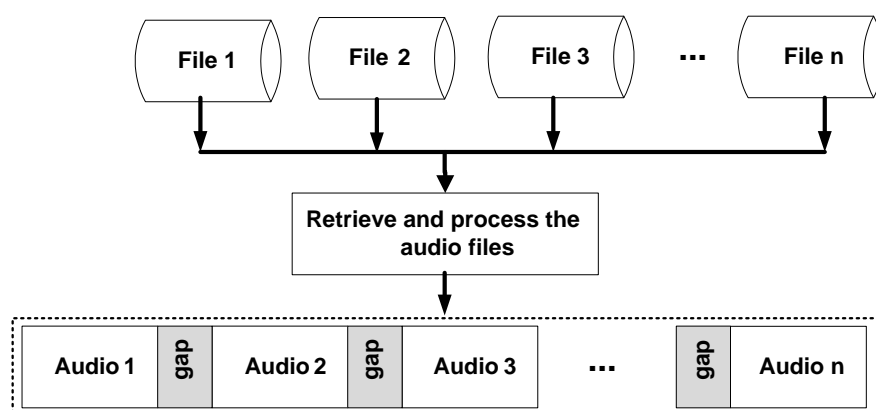
Figure 7.22: Working with sentence audio

For a more complicated sentence structure the audio can be stored and retrieved from the "QueAudioFile" file. The structure of the file is shown in Table 7.5.

Fields	Description
AuID	Audio ID
AuID	Audio file
Desc	Description of the audio file

Table 7.5: The structure of "QueAudioFile" file

7.8   Testing the functionality of the system

The functionality of the system internal components such as the learner model, the instructional strategies and instructional model are tested to evaluate the outputs of each component. The outputs of the learner model (i.e. fuzzy inference) are compared to the outputs of MATLAB and both outputs were identical. Also the outputs of the instructional strategies and the instructional model are inspected by analysing the contents of the learners' log files and the learning database. The analysis showed that these components complied with their specified requirements as discussed in chapter 3.

Input validation of the tutoring system is made by testing the system with 'typical' inputs and incorrect or illogical inputs (e.g. -1) and the system's response to these inputs is observed. The learner-interface has two types of inputs: selection type and supply type. The supply type which represents the learner's answers to certain questions is based on text matching. However any input other than the correct answer is regard as incorrect answer. The inputs of the teacher-interface are tested with 'typical' inputs and incorrect or illogical inputs. The system presented proper validation message such as 'input entered is not correct' and 'Field is required' when illogical input is entered or field is left empty.

7.9   Testing the Pluggable domain

The data for the domain knowledge (i.e. Arabic language) of the tutoring system was replace with new domain (i.e. French language) in order to test the feature of the pluggable domain discussed in Chapter 6. The intention is to see the effect of the new knowledge on the learning process.

The French language components has been supplied and verified by a French teacher. The evaluation of the testing process showed that the presentation of the teaching materials, the interaction of the instructional strategies and the learner model were not affected by the introduction of new DK. Moreover the conversion process of the Arabic language knowledge to French knowledge was successful which verify the pluggable domain feature of the tutoring system. However the DK for teaching complete French language course needed to be plugged and the learning process of this knowledge need to be evaluated via full version of the tutoring system and test group in order to provide more insight into this process.

7.10   Summary

This chapter discussed the implementation of the components of the tutoring system. The tutoring system is based on client/server architecture. Central to the design is the fuzzy inference (i.e. Learner model) is implemented as a PHP function "Fuzzy" which takes two crisp inputs and produce one fuzzy class that represents certain learner's stereotype. The function Fuzzy performs the fuzzification, defuzzification and classification process of the fuzzy inference. The implementation of the process of modifying and selecting the proper instructional strategy is based on generic rules which can be reconfigured by the teacher to adapt to different learning setup. The tutoring system has two user-interfaces: the learner-interface and the teacher-interface. The learner-interface provides various teaching materials, feedbacks, and explanations to the learner. The learner interaction with the tutoring system has two modes: the interactive-based learning and discovery based learning. The learner interaction with the system is presented through various DFDs. The teacher-interface

allows the teacher to reconfigure various components of the system such as the learner model and the instructional strategies. Moreover, the teacher can customise (i.e. add and/or modify) the contents of the system's templates and the domain knowledge. The teacher interaction with the system is presented through various DFDs. Finally the implementation of various tools used by the pluggable domain are presented and discussed. The next chapter discusses the evaluation of the tutoring system and analysis of the results.

CHAPTER 8: SYSTEM EVALUATION

This chapter presents the evaluation of the tutoring system and analyses the results of the evaluation. The results will be utilised as the basis for recommendations for change in the design methodology of the proposed tutoring system. In addition the results of training and testing the ANN model are also included.

8.1   Background

The evaluation process is an essential part of the development of any system since it shows the educational impact of an ITS on learners. To test the effectiveness and reliability of the ITS in a real environment with typical learners, a series of evaluations were conducted. The learners who participated in the evaluation process were mature English-speaking foreigners who work in Libya and have the desire to learn Arabic, as well as, Libyan English-speaking language teachers who have the need to assess the tutoring system. A pre-test and post-test analysis combined with data collected from the learners' log files are used to evaluate the system (Ainsworth, 2008). The goal is to assess the extent the system had on the learning process and to measure the learners' knowledge and understanding of the taught language (Jeremic, *et al.*, 2009; Kong, *et al.,* 2009).

In addition, a learners' questionnaire was used to collect the learners' perception of the system at the end of the learning process. Moreover, human teachers were asked to complete a teachers' questionnaire, regarding the teaching strategies and friendliness of the system. The complete text of both questionnaires (i.e. learners and teachers) is shown in Appendix A. Moreover log files that capture the learners' actions and the system output are investigated to see how effective the instructional strategies and the learner model in the actual learning setup.

8.2   Learners' evaluations

In order to evaluate the effectiveness of the system and its contribution to language learning, two experiments were conducted. The experiments involved two groups of learners studying Arabic language. The first group, *experimental group*, used a fully functional system and named ITS version. The second group, *control group*, used a cut-down version of the system, named CALL version. The CALL version contained a set of lessons, practices, and questions similar to the ITS version, however, all of the AI components (i.e. learner model, instructional strategies, etc.) were removed and the learners can interact with the CALL version in an open environment. Comparison between the ITS and the CALL versions of the tutoring system is shown in Table 8.1.

Components	ITS	CALL
Interactive-based learning	Included	Not Included
Discovery-based learning	Included	Included
Learner model	Included	Not Included
Feedback, Hint and Explanation	Included	Not Included
Lessons	Included	Included
Practices	Included	Included
Questions	Included	Included
Examinations	Included	Included

Table 8.1: Comparison between the ITS and the CALL
versions of the tutoring system

Forty participants were recruited for this study. Two experiments were conducted with twenty participants in each experiment. The participants were randomly divided into two equal-sized groups each with twenty participants. Each group was allocated a set of machines running the appropriate version of the software (i.e. ITS or CALL). The difference was in the version of the tutoring system they interacted with. The evaluation process is shown in Figure 8.1.
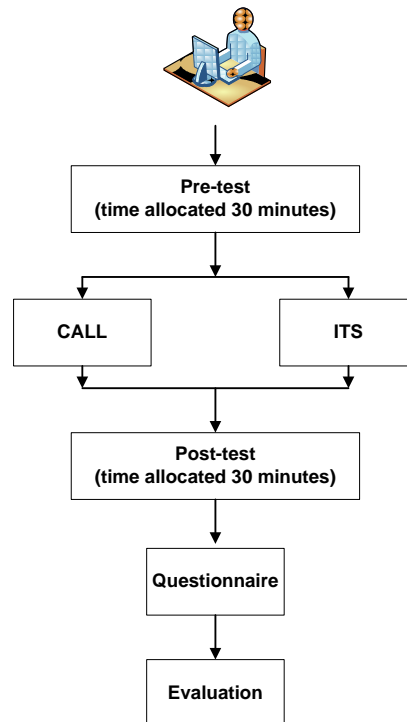
Figure 8.1: The evaluation process

The study was conducted in the Software Engineering Department, of the Faculty of Information Technology, Garyounis University, Libya. Each evaluation session was carried on in three distinct phases:

- Pre-test / Post-test
- Interaction with the system (ITS version or CALL version)
- Learners' questionnaire

The three phases of the experiment study are explained in the next sections. The learning gains are measured using questions at the end of each lesson and Examinations at the end of each learning level (beginner, intermediate, or advanced). Participants from Experimental group are required to answer all the questions and Examinations during the learning process.

8.2.1 Pre-test / Post-test

The pre- and post-tests are included to evaluate the learners' knowledge of the Arabic language before and after the interaction with the two versions of the tutoring

system. The pre-test (i.e. the placement test) and the post-test (i.e. final examination) consist of a multiple choice and free form questions. The placement test is designed to test the learners' elementary knowledge of the Arabic language. Additionally, this pre-test is aimed at identifying the learners' expectations from the course as well as their previous experience in the subject matter. The final examination is designed to test the learners' knowledge and what have been learned after completing the course using the tutoring system (Fossati, 2008). The learning of each group during the evaluation session is computed by comparing the results of both tests. Scoring higher in the post-test conclude that learners acquired knowledge by interacting with the system (ITS version and CALL version). The score results of the two groups can be used to compute the efficiency of the system. In other words, if the Experimental group shows a higher improvement in the post-test compared to the control group, it implies that learners learn more with the ITS version. To avoid the effect of bias from a particular test, the pre- and post-tests were identical for both groups. The tests were designed to be completed in less than 30 minutes.

A paired t-test is used to indicate the significant differences between the pre-test and the post-test achievements for both groups: ITS version and CALL version (Chien *et al.*, 2008). The paired t-test is used to compare two population means in which you have two samples of observations are paired together (e.g. learners' test results before and after a particular course). The aim of the test is to find out if the tutoring system leads to improvements in learners' knowledge/skills (i.e. test results). The analysis of the paired t-test is carried out using SPSS (PASW statistics, release 2009). The description of the paired t-test is provided in Appendix B.

To make sure all the learners work on the pre-test before starting the learning process, all the participants from the control group were asked to work on the pre-test under the supervision of a human supervisor. The pre-test is presented to the participants from the Experimental group based on a strategy controlled by the tutoring system. After completing the pre-test, participants from each group started working with one version of the tutoring system (i.e. ITS or CALL). After completing the learning process, all the participants from control group are asked to work on the post-test by a human supervisor.

8.2.2 Interaction with the system

All participants interacted with both versions of the tutoring system: the ITS and CALL versions. The teaching materials (i.e. lessons, practices, and questions) for both versions were identical. The participants of the Experimental group using the ITS version were required to complete the current lesson before moving on to the next one, based on the requirements of the tutoring system which guides them throughout the learning process, as discussed in Chapter 3. Those who proceeded with the CALL version, the control group, were free to skip any lesson, practice, or question as they pleased.

However, due to their important nature in the evaluation process, in the CALL version access to the pre- and post-tests were restricted. The access to the placement test and the final test can be done only under the supervision of a human supervisor. This is to make sure these tests are answered by the learners before and after the learning process.

In the CALL version experiments, the learners' personal information and the information for Practices and Questions are stored in the system. The CALL version shares the same information for lessons and Examinations with the ITS version as shown in Figure 8.2.
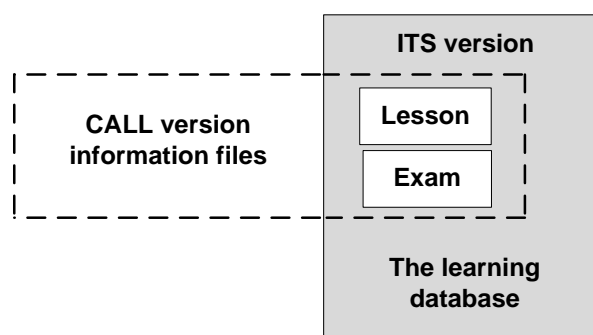
Figure 8.2: Information shared between CALL and ITS versions

8.2.3 Learners' questionnaire

The learners' questionnaire is used to assess the tutoring system and records the participants' perception of the system. The questionnaire contains a total of 42

questions ranging from the participant's background knowledge to their learning experience with the system. Five question responses and free-form responses are included in all the questionnaires. The responses' scale system range from strongly agree (1) to strongly disagree (5) or little (1) to excellent (5). The participants are also encouraged to give free-form responses to summarise their experience with the tutoring system. The complete questionnaire is included in Appendix B.

8.3   System assessment by human teachers

The assessment of the tutoring system by human teachers is based on teachers' questionnaire. The teachers participated in this assessment with the purpose of evaluating their perception of the tutoring system. The teachers' questionnaire contained a total of 44 questions which ask the teacher to rank their perception on various aspects of using the tutoring system with the focus on the contents covered and the ease of using the system (i.e. clarity). The same as on the learners' questionnaire five questions and free-form responses are included in all the questionnaires. The questions are based on five responses scale system ranging from strongly agree (1) to strongly disagree (5) or little (1) to excellent (5), and free-form responses are also allowed to record any suggestions that may enhance the tutoring system. The questionnaire is included in Appendix B. In order to conduct the assessment experiment, 4 language teachers are recruited (3 males and 1 female). The teachers were given the chance to use the tutoring system and observe other learners using the tutoring system during the learning process. The duration of the assessment of the tutoring system by human teachers was 4 weeks.

8.4   New language domain (French)

In order to verify the concept of the pluggable domain, the Arabic domain knowledge is replaced with the French domain knowledge. The system should be able to teach the new language properly without changing the tutoring system design or its production rules. The aim is to test the usefulness of this approach in foreign language teaching. A new very limited version of the tutoring system was developed

for the teaching of the French language. This version contains 6 lessons with two lessons in each learning level. The results of this process were tested to the knowledge of French language teacher.

8.5   Results and discussion

The following section discusses the results obtained from the experiment conducted in the work of this thesis.

8.5.1   The ANN model experimentation and results

The results of the design and the training errors for 3-layer ANN with different number of hidden units are presented in this section. The design of the final structure of the ANN was based on the lowest training error values. In the training mode, the learning cycles were fixed to 50,000 while the number of units in the hidden layer was varied from 4 to 16. The average error last cycle in the training set is presented in Figure 8.3. The ANN showed excellent training results and the network converged to a minimum error with 10 units in the hidden layer. In the next stage, the network structure was fixed to 10 units in the hidden layer and the learning cycles were varied from 20,000 to 80,000. The average error last cycle in the training set is presented in Figure 8.4.
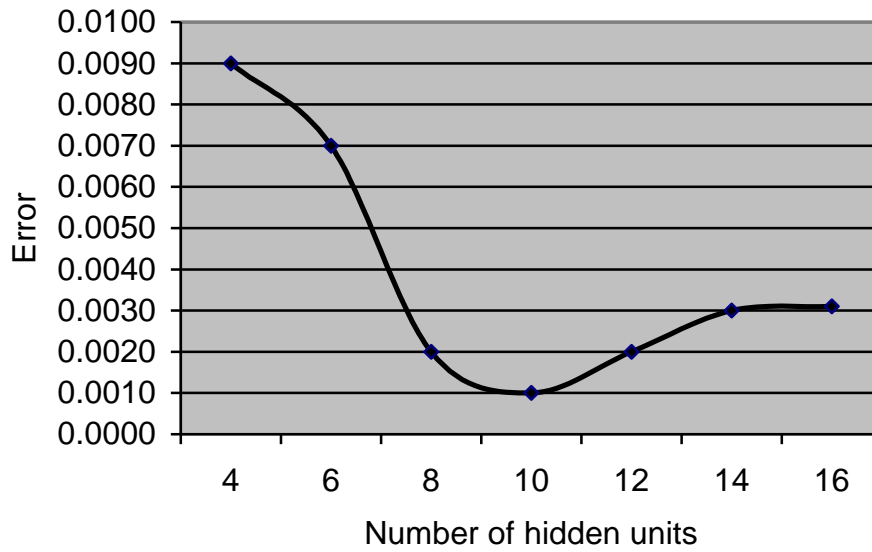
Figure 8.3: The learning errors results from training with different number of hidden units
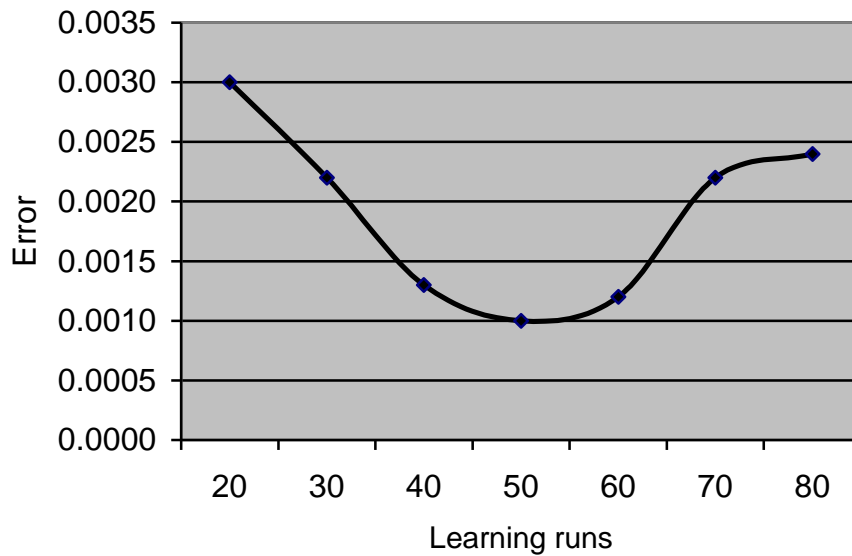
Figure 8.4: The learning errors results from training with 10 hidden units and different number of learning runs

In test mode, the ANN went through one learning cycle in all testing patterns. The generated outputs were drawn and the average error between the ANN outputs and the real output were calculated. The average error in the prediction of each output was less than 0.001%. The correlation results showed the strength and direction of a linear relationship between the two outputs. The correlation the ANN outputs and the real output were over 0.99.

8.5.2 Evaluating the tutoring system

Two experiments with the same learning setup were conducted in this study. In each experiment the ITS was compared to a CALL version of the tutoring system in tutoring learners on the topic of Arabic language with the intention of assessing the affect of the intelligent components of the ITS such as the learner model and feedback and hints. The aims of these experiments are to evaluate the effectiveness of the tutoring system and its contribution to language learning. In addition it records the understanding of the Arabic language by learners after completing the learning process. In each experiment the learners are randomly divided in to two groups each with 20 learners. In both groups, all the participants worked individually at their own paces. There were 22 lessons in total presented in three learning levels. All events such as logging in and submitting and answers were recorded in a log file specific to each learner. Results from both experiments are discussed in the next two subsections.

8.5.2.1 Evaluation experiment set-up

The evaluation experiment involved 40 participants randomly allocated to the experimental group or the control group. This number of participants is considered appropriate within the time and resources available for the research. General information about the sex, motivation for studying the course and the age of the learners is presented in Table 8.2.

Feature	Number of learners	Experimental group	Control group
Sex	Male	18	18
	Female	2	2
Motivation	Religious purposes	4	5
	To do business effectively	4	5
	Tourism	3	4
	Others	9	6
Age	between 15 and 25	4	4
	between 25 and 35	5	4
	were between 35 and 45	5	7
	Over 45	6	5

Table 8.2: General information about the learners of the Experiment

8.5.2.2 *Evaluation experiment findings*

The finding from the evaluation experiment showed that the learners of the control group spent more time interacting with the tutoring system than the experimental group. Table 8.3 shows the time (minutes) spent by learners from both groups (i.e. experimental and control group) working with the touring system. The total learners' interaction with the system was 200 hours for the experimental group compared to 222 hours for the control group. The average interaction per learner form the experimental group was 101 minutes compared to 113 minutes from the control group. As expected the learners spent long time to finish the course. This is probably due to the fact that the tutoring system contains a large number of lessons (i.e. 22 lessons).

	Experimental group	Control group
The total learners' interaction with the system (hours)	200	222
The average learners interaction per session (minutes)	101	113

Table 8.3: Time spent by learners from the experimental and control group

A comparison of the learners' average grades of the pre-test and post-test from the ITS experiment were used to evaluate the progress of the learners' Grammar, Listening, and Reading skills. The results showed that Grammar skills of the learners of experimental group increased (learning Gain) by 30%. Learners' Listening and

Reading skills were also strengthened with learning gain increased by 48% and 40% for Listening and Reading skills respectively (see Figure 8.5).
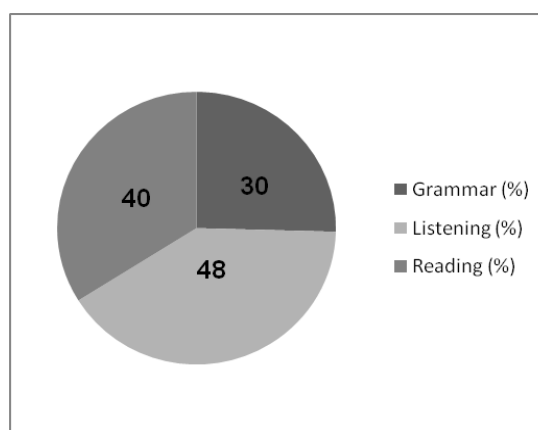
Figure 8.5: The learning gain of the learners' Grammar, Listening, and Reading skills of the experimental group

This is compared to the results of the control group in which Grammar, Listening, and Reading skills also strengthened with learning gain increased by 17%, 34% and 27% respectively (see Figure 8.6). Both groups achieved better results in the post-test as compared to the pre-test. However the experimental group had shown a greater increase in scores between the post-test and the pre-test (i.e. learning gain). As expected the learning gain of the learners' Grammar skills was lower than the learners' Listening and Reading skills since the grammar skills contains elements of the Listening and Reading skills.
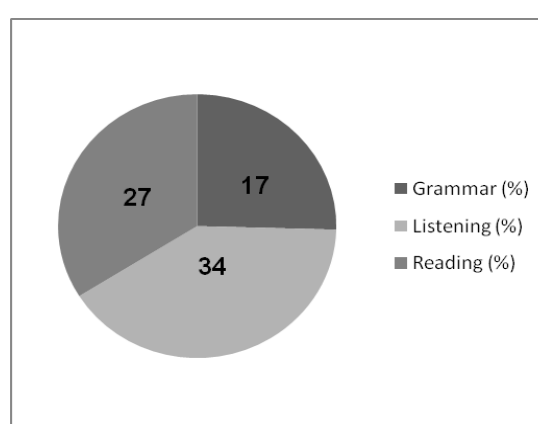
Figure 8.6: The learning gain of the learners' Grammar, Listening, and Reading skills of the control group

The total number of questions answered by the learners was 440 for the ITS version compared to 463 from the CALL version. The evaluation results showed that the

learners of the control group spent more time (i.e. Average of 4 minutes) working with the questions compared to the learners of the experimental group (i.e. Average of 2 minutes). This is partially because the learners in the CALL version can answer the same questions more than once which is not the case in the ITS version. However some learners from the control group skipped some questions while others answered the same questions more than once.

In the ITS version, the instructional strategy uses the learner model to generate feedback and hints to the learners. The learners from the experimental group made 470 requests for feedback but they received only 407. Moreover the learners made 257 requests for hints but they received 223 only. The total number of the requested and received feedback and hints were comparable with the feedback and hinting strategy discussed in Chapter 4. Thus it can be concluded that the feedback and hints were effective in assisting the learners during the questions sessions giving that the final Question average grade form the experimental group was 66% compare to 59% from the control group. The results showed that the learners relied more on the feedback than hints with the average number of feedback received per question is 0.92 compared to 0.51 for hints. This is because feedback provides more descriptive information than hints. The results showed that the learners received about 1 feedback per Question which is relatively good given that the maximum number of feedback learner can receive is 2.

Features	ITS	CALL
Average number of practices	200	276
Average number of practices for the first learning level	60	94
Average number of practices for the second learning level	81	92
Average number of practices for the third learning level	59	90

Table 8.4: The average number of Practices interactions

The average number of practice interactions was 200 for the experimental group compared to 276 for control group (see Table 8.4). This is probably due to the fact that there is no minimum time constrains on studying the practices on the CALL version. The analysis of the log files showed that most of the learners from the experimental group skipped the practice when it was not required by the instructional strategy while learners from the control group did not interact with all the tutoring system practices. This is implying that although the discovery based learning (i.e.

130

unconstrained environment) was successful in encouraging the learners to spend more time with the tutoring system nevertheless it was not effective in improving the learning gain of the learners. In addition there is no linear model of discovery learning since learner process information in different orders. Therefore the system may fail to detect problems and misconceptions of learner using this mode of learning. Thus there should be a mechanism for forcing the learners to study certain teaching materials (e.g. practice and prerequisites) based on certain measurements of learners (e.g. performance). Moreover an alternative option for using discovery based learning should be provided by the tutoring system when the learner needs to do that. In the proposed system this option is provided to the learner as discussed in Chapter 3.

8.5.2.3   Pre- and Post-test performance

Table 8.5 shows data for the two groups (i.e. ITS and CALL) of participants. The mean score for pre-test of the experimental group (ITS group) was 6.50, while the post-test for this group was 41.50. The pre- and post test are discussed in Section 8.2.1. Paired-sample t-test among participants in ITS group for pre-test and post-test ($t = 10.345$, $p < 0.001$) indicated that there was significant differences between pre-test and post-test accomplishment. The mean learning gain score for the experimental group of participants was 35 out of 100. In the control group (CALL group), the mean score of the pre-test for the participants was 11.50 and the post-test was as 36.40. A paired sample t-test ($t = 7.46$, $p < 0.001$) indicated that there was a significant difference between pre-test and post-test achievements. The mean gain score for CALL only group was 25 out of 100.

Group	n	Pre-test	Post-test	Mean gain	t
ITS	20	6.50	41.50	35	10.34
CALL	20	11.50	36.40	25	7.46

Table 8.5: Pre-test and Post-test for ITS and CALL approach

Thus, both groups (i.e. experimental and control group) achieved better results in the post-test as compared to the pre-test results. Nevertheless the ITS experimental group had shown a more increase in scores between the post-test and the pre-test. Therefore, we can conclude from these results that learners who used ITS version of the tutoring system learnt more than learners who used the CALL version.

8.5.2.4   Analysis of learners' questionnaire

The analysis of the learners' questionnaire responses from both groups (i.e. control and experimental) are analogous with our expectations due to the differences in both systems. Learners who used the ITS version had a better perception of the system as a whole. This was shown in their responses to whether they would recommend the tutoring system to anyone who wants to learn a new language, where approximately 90% of the experimental group learners indicated that they would with 55% agree and 35% strongly agree (see Table 8.6). Conversely the percentage of the control group learners who had the same opinion was lower, approximately 70%. Also 80% of learners from the experimental group agreeing that learning a second language using this tutoring system are useful. On the other hand 70% of learners from the control group agreeing that learning a second language using this tutoring system are useful.

	Strongly agree	Agree	Neutral	Disagree	Strongly disagree
I recommend this tutoring system to anyone who wants to learn a new language	7	11	2	0	0
I found learning a second language using this tutoring system is useful	4	12	4	0	0

Table 8.6: Questionnaire responses of the learners' perception of the system

The overall effect of the tutoring system's supports on the learning process is evaluated based on the questionnaire responses of the experimental group (see Table 8.7). The analysis results showed that 90% of the learners are satisfied with the system's support with 65% "agree" and 25% "strongly agree". Moreover 75% of the learners agree with 20% strongly agree that the system's support is clear. Finally 80% agree with 10% strongly agree that the system's support help them to become productive quickly.

	Strongly agree	Agree	Neutral	Disagree	Strongly disagree
Overall, I am satisfied with the system's support	5	13	2	0	0
The system's support (e.g. feedback, hints, explanations and thoughts) is clear	4	11	2	2	1
The system's support helped me to become productive quickly	2	14	4	0	0

Table 8.7: Questionnaire responses of the learners' perception of the system's support

The effect of each feature of the system's support is collected and presented in Table 8.8. Most participants were relatively happy with the system's feedback with 40% think it is "Average" and 40% think it is "Above average" while 45% think the usefulness of the system hints are "Below average" and 45% think it is "Average". Moreover 35% of the participants think the usefulness of the system explanation is "Above average" and 35% think it is "Average".

Lastly more than 40% of the learners think the usefulness of the Comments editor is "Below average" while 35% of the participants think the usefulness of the Comments editor is "Little". On the other hand although the learner-interface of the tutoring system provides this feature nevertheless it is not part of the learner model and the instructional strategies nor does it interfere with instructional strategies.

	Little	Below average	Average	Above average	Excellent
Was the system feedback helpful	0	3	8	8	1
Were the system Hints helpful	0	9	9	2	0
Were the system explanations helpful	0	3	7	7	3
Was the system comments editor control helpful	7	8	5	0	0

Table 8.8: Questionnaire responses of the learners' perception of each feature of the system's support system's support

The usefulness of the Lesson's components such as Practice, Prerequisites, Extra tutoring, and Questions in the learning process were evaluated and presented in Table 8.9. The results showed that 75% of the learners from the experimental group agree that the Practice was useful in the learning process compared to 80% from control group. Moreover 65% of the learners form The experimental group agree that the Extra tutoring was useful in the learning process while 85% agree that the Pre-requisites was useful. Finally 90% of the learners from the experimental group think the Questions and Examinations set by the tutoring system were realistic compared to 85% from control group. It is clear that most of the learners from the experimental group found that the Practice model is very useful since it provides unlimited interaction with teaching materials and the questions of the current lesson plus it provides analysis of their interaction with Practice model. On the other hand these interactions are not provided by the CALL version of the tutoring system.

133

	Strongly agree	Agree	Neutral	Disagree	Strongly disagree
Studying the Practices was very useful in the learning process	3	12	3	1	1
Studying the Extra tutoring was very useful in the learning process	1	12	5	2	0
Studying the Pre-requisites was very useful in the learning process	2	15	3	0	0
The Questions and Exams set by the tutoring system were realistic	8	10	1	1	0

Table 8.9: Questionnaire responses of the learners' perception of the Lesson's components

The learners from the experimental group responses to the Questionnaire's free-form were retrieved and analysed; however not all learners provided free-form remarks. Most of these remarks were general and related to the improvement of the system while some negative remarks regarding the clarity of the system's audios, feedback/hints and the time duration of the evaluation process.

Overall the evaluation of the tutoring system by the learners showed that 30% of the learners think the system's Explanation is the feature of the system they prefer the most. This is probably because this feature provides comprehensive information about the learner at any time (i.e. on-demand explanation) during the learning process compared to the feedback and hints which provide restricted help. Moreover 30% of the learners are "Not sure" about which feature of the system they prefer while 25% of the learners prefer all the features of the system.

From the evaluation questionnaires and discussions with participants, we also identified that some learners had difficulties in understanding the functionality of some buttons' name such as "discovery based learning" and "continue learning". As a result these buttons' names were replaced to "browse previous lessons" and "next lesson" respectively. Moreover the feedback/hints messages presented by tutoring system were not clear to some of the learners. This is because the feedback/hints do not provide explicit answers to the questions. Hence the feedback/hints messages of the tutoring system were short and less descriptive.

8.5.2.5   Analysis of the logs file

A log file that captures the learner's actions and the system outputs was produced for each learner. The aim is to track the learners' interaction with the tutoring system and analyse the role of the instructional strategies in the learning process. Sample results from log files of two learners' learning sessions are presented in Table C.1 (see Appendix C).

By comparing the contents of the two log files it is apparent that the first learner starts the learning process immediately. This is because he/she chose his/her Previous Arabic language Proficiency as a "None" and as a consequence of that the tutoring system skipped the "Placement test" process and mark the learner a score of 0 for the Placement test (see line 1 in Table C.1). On the other hand the second learner is forced by the tutoring system to take the placement test before starting the learning process since he/she selected his/her previous Arabic language Proficiency as a "Beginner". The second learner's Placement test result was 20% (see line 1 in Table C.1).

The learner's stereotypes were computed before starting the learning process. The learner's stereotypes were similar for both learners (i.e. default stereotyping). The Performance results were "Underachiever" while the Help-seeking stereotype was "Rare" for both learners. Also the learner's Engagement was "Not engaged" for both learners (see line 2 in Table C.1).

The lesson "Introduction to Verbs" was chosen to compare the progresses of both learners after eight lessons. According to the selected instructional, strategy 2, the first learner must study all the lesson's practices, pre-requisites and Extra tutoring while the second learner must study only the lesson's practices and Extra tutoring, and it was optional for him/her to study the pre-requisites (see line 3 in Table C.1). This is comparable with the instructional strategy design (as discussed in Chapter 3) since at that stage the first learner Performance was "Underachiever" while the second learner Performance was "Excellent".

135

The interaction with the Practice model is presented in line 4 in Table C.1. The comparison between the two log files showed different responses to the learners' answers for the Practice questions. Moreover the Practice model provided analysis and corrective feedback to the learners based on their current and previous interaction with Practice model as explained in Chapter 3.

The interaction with the Question model is presented in line 5 of Table C.1. The comparison showed different responses to different learners. The Question model provided different corrective feedback and analysis of the results based on the current learner's answers and the previous learners' stereotype and interaction with the discovery based learning as described in Chapter 4.

The final stereotypes varied for both learners where they were (*Performance=Underachiever, Help-seeking=Rare, Engagement=Often Engaged*) for the first learner and (*Performance=Strong, Help-seeking=Medium, Engagement= Often Engaged*) for the second learner (see line 6 in Table C.1). The final Performance for the first learner was "Underachiever" because his previous question average was 47% and Examination average was 30% while the final Performance for the second learner was "Strong" since his/her previous question average was 89% and Examination average was 48%. The final Help-seeking for the first learner was "Rare" given that he requested 16 feedback and 6 hints while the final Help-seeking for the second learner was "Medium" since he requested 35 feedback and 15 hints during the learning process. The Engagement was "Often Engaged" for both learners since both learners spent almost similar amount of time per session given that the first learner spent 73 minutes per session compared to the second learner who spent 77 minutes per session.

The log files collected during the evaluation experiments provided insight into the operation of the tutoring system, both at initialisation and during the learner's interaction with the tutoring system. Comparing two log files from the evaluation experiment (see Table C.1), it can be concluded that the tutoring system was adapting to the learners activities during the learning process which prevented the tutoring system from presenting the same teaching materials and control information repeatedly. Moreover a learner model with multiple-stereotype enabled several

different forms of background data about the learner to be taken into account, and provided a profile of the learner in terms of pedagogically useful attributes. The system was able to select for the learner the instructional strategy appropriate for certain learning material.

8.5.3 The tutoring system assessment by human teachers

The assessment of the tutoring system by human teachers was based on a system-based assessment questionnaire. The opinions of the teachers participated in this assessment were collected with the purpose of evaluating their perception of the tutoring system and for future improvement of the tutoring system.

From the analysis of the teachers' questionnaires it can be concluded that teachers think that the theoretical concepts are adequately complemented by examples, exercises and problems however they think that the course was not entirely covered. This is possibly because the tutoring system is developed as a prototype based on limited teaching material of the Arabic language.

All the teachers acknowledge that the teaching materials were organised in logical sequence. However, they think that the audio-visual presented by the system were ineffective. The reason for that may be due to the fact that only one video was provided by the tutoring system for the lesson "Introduction to Arabic characters". Furthermore, the quality of some of the audios was below average or not consistent in terms of volume. This is because all the audios were recorded by the researcher using the simple Windows sound recorder. For the same reasons perhaps they think the language pronunciations were not easy to understand. On the other hand they acknowledge that reading text on the tutoring system's pages was very easy.

Regarding the presentation of the tutoring system they agreed that the organisation of the teaching materials on the tutoring system's pages and the messages presented by the tutoring system were very clear. Also all the teachers would like to use the tutoring system frequently. Moreover they acknowledge the usefulness of reconfiguring the tutoring system via the teacher's interface.

Finally the teachers have acknowledged that they would recommend the tutoring system to be used for teaching foreign languages. The tutoring system's Practices were considered as the most useful in the learning process compared to the Extra tutoring and Pre-requisites. In addition all the teachers confirmed that the Questions and Examinations set by the system were realistic. However they preferred to have human assistance to explain to them how to use the teacher's interface.

In general the overall evaluation of the tutoring system by teachers was below the expectation in some aspects of the tutoring system such as the limitations of the Arabic language teaching materials and the quality of some of the multimedia components such as audios and videos. However some good perception is recorded concerning reconfiguring the tutoring system, organisation of the teaching materials and the recommendation for the tutoring system to be used for teaching foreign languages.

8.6   Summary

This chapter has presented and discussed the evaluation studies of the tutoring system by learners and teachers. Two different evaluation studies (i.e. learners and teachers evaluation) were conducted where each of these studies had a specific focus. The learners' evaluation evaluates the effectiveness of the tutoring system and its contribution to language learning. Two experiments were conducted in this process with two versions of the tutoring system: the ITS version and a cut down version named CALL version. The results of these experiments are based on a pre-test/post-test results and questionnaire analysis. The assessment of the tutoring system by teachers is based on system assessment questionnaire with the purpose of evaluating the teachers' perception of the tutoring system. Moreover, in order to verify the concept of the pluggable domain, the Arabic language domain knowledge is replaced with a limited subset of French language domain. Finally, the evaluation finding, learners' questionnaire analysis and comparison of the log files of two learners are presented and discussed. A paired t-test is used to indicate the significant differences between the learning achievements for both groups: ITS version and CALL version. In addition an analysis of the teachers' questionnaire is

presented and discussed. Finally the pluggable domain feature of the tutoring system was successfully verified through sample of French knowledge. The next chapter presents and discusses the main contributions of this research.

CHAPTER 9: RESEARCH CONTRIBUTIONS AND FUTURE WORK

This chapter discusses the main contributions of this research and proposes some future work that can be based on it.

9.1 Research Contributions

The research contributions can be summarised in the next sections.

9.1.1   A framework for differentiated instruction

This research introduces the design of a framework for selecting the appropriate instructional strategies for teaching a certain lesson or question based on its attributes. These attributes includes pedagogical parameters from learning materials such as difficulty and learning level as discussed in chapter 3.

Based on the design of the instructional strategies, each combination of these attributes is mapped into a certain instructional strategy. Two groups of different instructional strategies are developed. Each group contains different strategies and each strategy is pre-defined based on the combination of the input's attributes. The first group is concerned with providing the proper instructional strategies for lessons. The second group is concerned with providing the proper instructional strategies for questions. The instructional strategies can also include various learners' characteristics in its design. Three features (activities) which control the actions of each instructional strategy are embedded in the design of each group. For lessons' strategies the features are practice, pre-requisites and extra tutoring. For questions' strategies the features are hint, feedback and time.

The selection process of the proper instructional strategy for certain learning materials: lessons or questions are based on a set of rules. The design of these rules is based on "reconfigurable" design in which the teacher, who has no background in computer programming, can adjust the selection process via the

teacher-interface. In addition the teacher can adjust the features of each instructional strategy.

The advantages of the presented framework over the existing techniques are that it provides *meta-strategies* that select the appropriate tutoring strategy for a given situation. The meta-strategies are based on both the teaching materials' attributes and various learners' characteristics. This has the advantages of allowing for the reconfiguration of all aspects of the instructional strategy such as the selection process, the teaching materials' attributes and the features of each teaching materials. In addition, the framework introduces the concept of extra tutoring as a remedy for weak learners. Furthermore, the system's support is based on all the learners' characteristics. This has the advantages over previous research of providing more realistic support for learners during the learning process.

9.1.2 Multiple fuzzy inferences (Stereotypes) for representing the learner's learning characteristics.

This research introduces the design of a framework for representing the learner model based on three different characteristics: *performance*, *help-seeking* and *engagement*. These characteristics represent important factors in indicating the online learner's attitude towards the learning process as discussed in chapter 4.

The learner model is represented using multiple stereotypes implemented as multiple-fuzzy inferences. A general fuzzy membership function was developed to represent the linguistic sets of each fuzzy inference. Fuzzy inferences are adapted since most of the human reasoning is associated with the use of fuzzy rules. Fuzzy logic can be used to model the humans' decision making and common sense. This allows the overlapping of vague concepts and overcomes limitations such as lack of information. The advantage of using multiple-fuzzy inferences is that it allows the system to use the outcomes of these fuzzy inferences individually or in combination with each other. Furthermore, the proposed approach has distinctive reconfigurable structure so that it allows the teacher to alter various parameters of the instructional

141

strategies and the learner model according to the requirements of the learning process (i.e. no manual editing of program code is required).

Various stereotype learner models have been developed based on an abstract classification of frequently occurring characteristics of learners. However, most of these systems move the learner from one stereotype to another, but they do not change the stereotypes themselves. In this research the design of the tutoring system allows for the modification of its learner model via user-friendly authoring interfaces designed for use by the teachers. Since there is no definitive answer to what the system design should optimally look like, it seems rational to give the teacher some choices in the system design matter

An existing fuzzy technology was used to develop a new approach for learner modelling in ITSs based on multiple stereotype learners' modelling technique. The learner model was based on multiple-fuzzy inferences which have the advantages of allowing the outcome of these inferences to be used separately or in combination with each other in the reasoning process of the system. A new general membership function was developed to represent various linguistic variables. This approach has the advantage over the existing techniques by allowing the reconfiguration of various parts of the learner model by the teacher. Finally the design of the learner model is language-independent that can be applied to any languages other than Arabic that have similar linguistic features.

9.1.3 A framework for the pluggable-domain (i.e. independent domain)

This research introduces a new concept of a pluggable domain that can be used for teaching a new language. This concept was met by the separation between the control of the system represented by the course manager and the domain knowledge. This approach implemented a general design of the domain knowledge and a template-based conversion process from a source language (e.g. Arabic) to a target one (e.g. French) as discussed in chapter 6.

Various templates are developed for this system that includes: lessons, practices, extra tutoring, questions, and examinations templates. Each template may contain various teaching content such as texts, audios, pictures, or videos. These templates are stored in the learning database while their knowledge is retrieved from the domain knowledge.

The conversion process is the process of mapping certain teaching units of a source language into the target language. It includes the task of deciding which knowledge of the target language is the most appropriate equivalent of the source language teaching unit. In other words it represents the differences between the source language and the target one. In addition, the target language may specify additional components (see section 6.3) that differ mainly in their usage hence requiring the design of the domain knowledge to support all these equivalents.

This pluggable domain concept has the advantages of allowing an ITS designed to be adapted to the teaching of a new language using the same system's templates given that there are no tools yet developed for authoring natural languages. This is because authoring natural languages is more difficult due to the differences between the languages' structures and grammars. This approach is easy to implement compared to other techniques such as NLP and machine translation. Another advantage of this design which is especially important in language tutoring systems is the ability to teach colloquial languages such as Egyptian or Syrian Arabic that can be implemented using this concept.

9.2   Conclusions

In this research a framework for an adaptive tutoring system for teaching Modern Standard Arabic has been developed. The research, involved the design, development, and evaluation of the intelligent tutoring system, was interdisciplinary in scope, requiring an understanding of aspects of artificial intelligence (specifically, Intelligent Tutoring Systems, Fuzzy logic, and Artificial Neural Networks), education, and linguistics as part of the domain language study (Arabic). Table 9.1 illustrates

the proposed system against previous Arabic systems such as Arabic ICALL (Shaalan, 2005; Farghaly and Shaalan, 2009) and TLTS (Johnson and Valente, 2008).

	The proposed system	Arabic ICALL (Shaalan, 2005; Farghaly and Shaalan, 2009)	TLTS (Johnson and Valente, 2008)
Techniques	ILTS	ICALL	ILTS
Intelligent components	Fuzzy inference / Rule-based	NLP / Rule based	Speech recognition/ Pedagogical agent
Availability	*Web based*	Stand alone	Stand alone
Learner model	Stereotype model	Not provided	Diagnosis model
Teaching strategy (instruction model)	Sequencing based / guided discovery	Curriculum sequencing	Task-based approach / scaffolding
Feedback	On-demand / Corrective /Analysis	Error feedback	Corrective feedback
Hints	On-demand	Not provided	Not provided
Explanation	On-demand	Not provided	Not provided
Language skills	R G L	G	G S
Type of interaction with learner	Supply-type / selection-type.	Supply-type / selection-type.	Supply-type

Language skills key: R: reading; G: grammar; L: listening; S: speech

Table 9.1: Features of some of Arabic ILTS/ICALL teaching tools

The current researches in Arabic ILTS and ICALLs focus on the area of speech recognition and NLP. The proposed system offers multiple instructional strategies and multiple-stereotype learner model. The selection of each instructional strategy is based on general rules that provide means to a reconfigurable prediction. The learner model has a reconfigurable design that can be altered by the teacher at runtime via a teacher-interface. A pluggable-domain model design has been developed for teaching other languages. This approach allows the system to adapt to the teaching of a different language with minimal changes required.

The learning process of the tutoring system is enhanced by various types of feedback, hints and explanations as shown in table 10.1. Since online learners have differences in their languages, prior education level, skills level and dedication to the learning process. Thus, the focus was in developing a reconfigurable design that can be easily altered through user-interface at runtime.

In order to evaluate the effectiveness of the system and its contribution to language learning, two experiments were conducted. The experiments involved two groups of learners studying Arabic language. The first group, *experimental group*, used a fully

functional ITS version. The second group, *control group*, used a cut-down version of the system, referred to as CALL version. The CALL version contained a set of lessons, practices, and questions similar to the ITS version, however, all of the AI components (i.e. learner model, instructional strategies, etc.) were removed and the learners can interact with the CALL version in an open environment.

A paired t-test was used to indicate the significant differences between the pre-test and the post-test achievements for both groups: ITS version and CALL version. The evaluation experiment involved 40 participants randomly allocated to the experimental group or the control group. Both groups achieved better results in the post-test as compared to the pre-test results. The mean learning gain score for the experimental group of participants was 35 out of 100 compared to the mean learning gain of 25 out of 100 from the CALL group. Therefore, it can be concluded from these results that learners who used ITS version of the tutoring system learnt more than learners who used the CALL version.

The discovery based learning (i.e. unconstrained environment) provided by the tutoring system was successful in encouraging the learners to spend more time with the tutoring system nevertheless it was not effective in improving the learning gain of the learners. Thus there should be a mechanism for forcing the learners to study certain teaching materials (e.g. practice and prerequisites) based on certain measurements of learners (e.g. performance). Moreover an alternative option for using discovery based learning should be provided by the tutoring system when the learner needs to do that.

The evaluation results and analysis of the questionnaires (i.e. learners and teachers) demonstrated that the proposed design has potential in foreign language learning via computer applications. The fact that learners achieved better results in the ITS approach indicates that the system is viable instructional option. The multiple-fuzzy stereotyping of learner model worked well in categorizing the learners according to three different measures (i.e. performance, help-seeking and engagement). The tutoring system was able to adapt its instructional strategies according to the needs of the individual learner. The observations are very positive, despite being aware that

there are additional works to improve the tutoring system and in particular make better multimedia presentations.

9.3   Further work

Further work can be summarised by the following ideas:

- The implementation of the pluggable domain and the design of the domain knowledge were based on the conversion of teaching materials from Arabic to French. However more foreign languages should be investigated in order to verify the general design of the domain knowledge. Moreover the design of the pluggable domain only works with languages that have features similar to the Arabic language in terms of their general grammar structure (e.g. Nouns, Verbs, Adjectives, etc) such as most European languages (e.g. French, German, etc.). Other languages such as oriental languages need to be investigated given their unique features.
- Voice recognition feature can be added to the tutoring system to allow learners practice their pronunciation of words, etc.
- A game-based learning platform can be added to the tutoring system to help with the learning and assessment of students in situ.

REFERENCES

Abdel Monem, A., Shaalan, K., Rafea, A. & Baraka, H., (2008), "Generating Arabic Text in Multilingual Speech-to-Speech Machine Translation Framework", vol 22, pp205–258, *Machine Translation*, Springer, Netherlands.

Abu Naser, S., (2008), "JEE-Tutor: An intelligent tutoring system For Java Expression Evaluation", *Information Technology Journal*, vol 7, No. 3, pp528-532.

Ainsworth, S, (2007), "Using a Single Authoring Environment across the Lifespan of Learning", *Educational Technology & Society*, vol 10, pp22-31.

Ainsworth, S., (2008), "How should we evaluate multimedia learning environments", *Understanding Multimedia Comprehension*, pp249-265, Springer.

Ainsworth, S. & Fleming, P., (2005), "Evaluating a mixed-initiative authoring environment: Is Redeem for real", *Proceedings of the 12th International Conference on Artificial Intelligence in Education*, pp9–16, IOS Press, Amsterdam.

Aleven, V., Sewall, J., McLaren, B., & Koedinger, K., (2006a), "Rapid authoring of intelligent tutors for real-world and experimental use", *Proceedings of the 6th IEEE International Conference on Advanced Learning Technologies (ICALT 2006)*, pp847-851, Los alamitos, CA: IEEE Computer Society.

Aleven, V., McLaren, B., Roll, I. & Koedinger, K., (2006b), "Toward Meta-cognitive Tutoring: A Model of Help-Seeking with a Cognitive Tutor", *International Journal of Artificial Intelligence in Education*, pp101-130, IOS Press.

Alves, P., Amaral, L. & Pires, J., (2008), "Case-Based Reasoning Approach to Adaptive Web-Based Educational Systems", *Eighth IEEE International Conference on Advanced Learning Technologies*, pp.260-261.

Amaral, L., (2007), "Designing Intelligent Language Tutoring Systems: integrating Natural Language Processing technology into foreign language teaching", Ph.D. thesis, the Ohio State University.

Anderson, T. & Elloumi, F., (2004), "Theory and Practice of Online Learning", Athabasca University press, pp271-367.

Anderson, K., (2007), "Tips for teaching: Differentiating instruction to include all students", Preventing School Failure, vol 51, pp49-54, Heldref Publications.

Angelino, L., Williams, F. & Natvig, D., (2007), "Strategies to Engage Online Students and Reduce Attrition Rates", *the Journal of Educators Online*, vol 4, No. 2.

Anohina, A., (2007), "Advances in Intelligent Tutoring Systems: Problem-solving Modes and Model of Hints", *International Journal of Computers, Communications & Control*, vol 2, pp. 48-55.

Al-Neaimi, S., Harada, N., Sasaki, H. & Mizuno, K., (2009), "Development of an Arabic Language Learning Support System", *Proceedings of the 17th International Conference on Computers in Education*, Hong Kong: Asia-Pacific Society for Computers in Education, pp486-489.

Baker, R., (2007), "Modelling and Understanding Students' Off-Task Behaviour in Intelligent Tutoring Systems", *Proceedings of ACM CHI 2007: Computer-Human Interaction,* pp1059-1068.

Baker, R., Corbett, A. & Koedinger, K., (2004), "Detecting Student Misuse of Intelligent Tutoring Systems", *Lecture Notes in Computer Science: Intelligent Tutoring Systems*, pp531-540, Springer.

Baker, R., Corbett, A., Koedinger, K. & Roll, I., (2005), "Detecting when students game the system, across tutor subjects and classroom cohorts", *User Modelling*, pp220-224, Springer.

Baker, R., Koedinger, K., Evenson, S., Roll, I., Wagner, A., Naim, M., Raspat, J., Baker, D. & Beck, J., (2006), "Adapting to When Students Game an Intelligent Tutoring System", *Intelligent Tutoring Systems*, pp392-401.

Balasubramanian, A., Shankar, N., Subbaraman, S. & Rengaraj*, R., (*2008*),* "A Novel Fuzzy Logic Based Controller to Adjust the Brightness of the Television Screen with Respect to Surrounding Light", *World Academy of Science, Engineering and Technology*, vol 39, pp188-192.

Beck*, J., (2005), "*Engagement tracing: using response times to model student disengagement", *Artificial Intelligence in Education* - Supporting Learning through Intelligent and Socially Informed Technology*, vol 125, pp88-95, IOS Press.

Blessing, S., (2003), "A Programming by Demonstration Authoring Tool for Model-Tracing Tutors", *Authoring Tools for Advanced Technology Learning Environments: Toward Cost-Effective Adaptive, Interactive and Intelligent Educational Software*, pp93-119, Boston, MA: Kluwer Academic Publishers.

Blessing, S. & Gilbert, S., (2008), "Evaluating an authoring tool for model-tracing intelligent tutoring systems", *Proceedings of the ITS Conference,* pp204-215, Montreal, Springer.

Brian, Y., Anind, K. & Avrahami, D., (2009), "Why and why not explanations improve the intelligibility of context-aware intelligent systems", *Conference on Human Factors in Computing Systems, Proceedings of the 27th international conference on Human factors in computing systems*, pp2119-2128.

Brusilovsky, P. & Tasso, C., (2004), "User modelling for Web information retrieval", *Preface to special issue of User Modelling and User Adapted Interaction*, vol 14, pp147-157.

Bush, M., & Browne, J., (2004), "Teaching Arabic with Technology at BYU: Learning from the Past to Bridge to the Future", *CALICO Journal,* vol 21, pp497-522.

Butz, C., Hua, S., and Maguire& R., (2008), "Web-based Basyan Intelligent Tutoring Systems", in "Evaluation of the Web in Artificial intelligence environment", Nayak, R., Ichalkaranji, N.& Jain, C., (Eds.), pp221-222, Springer.

Chalfoun, P. & Frasson, C., (2008), "Subliminal priming enhances learning in a distant virtual 3D Intelligent Tutoring System", *IEEE Multidisciplinary Engineering Education Magazine*, vol 3, No. 4, pp125-130.

Chang, Y., Morales-Arroyo, M., Chavez, M. & Jimenez-Guzman, J., (2008), "Social Interaction with a Conversational Agent: An Exploratory Study", *Journal of Information Technology Research*, vol 1, pp14-26, IGI publishing.

Carmona, C., Castillo, G. & Millan, E., (2007), "Discovering Student Preferences in E-Learning", *Proceedings of the International Workshop on Applying Data Mining in e-Learning*, pp33-43.

Chen, C., Hong, C., Chen, S. & Liu, C., (2006), "Mining Formative Evaluation Rules Using Web-based Learning Portfolios for Web-based Learning Systems", *Educational Technology & Society*, vol 9, pp69-87.

Chew, L., (2008), "SCORM 2004 3rd Edition Specification and Intelligent Tutoring System - the differences and similarities", *Synthesis journal*, pp79-86.

Chien, T., Yunus, A., Wan Zah, W. & Bakar, R., (2008), "The effect of intelligent tutoring system (ITS) on student achievement in Algebraic expression", *International Journal of Instruction*, vol1, No. 2, pp25-38.

Chrysafiadi, K. & Virvou, M., (2008), "Personalized Teaching of a Programming language over the web: Stereotypes and rule-based mechanisms", JCKBSE, pp484-492.

Contreras, W., Galindo, E., Caballero, E. & Caballero, G., (2006), "An Intelligent Tutoring System for a Virtual E-learning Centre", *Current Developments in Technology-Assisted Education*, pp768-772.

Cristea, A., (2005), "Authoring of Adaptive Hypermedia", *Educational Technology & Society*, vol 8, pp6-8.

Cronje, J., Adendorff, D., Meyer, S. & Van Ryneveld, L., (2006), "Surviving the shipwreck: What makes online students stay online and learn?", *Educational Technology & Society,* vol 9, pp.185-193.

Crowley, R. & Medvedeva, O., (2006), "An Intelligent Tutoring System for Visual Classification Problem Solving", *Artificial Intelligence in Medicine*, vol 36, pp85-117.

Cushion, S. & Hemard, D., (2003), "Designing a CALL package for Arabic while learning the language ab initio", *Computer Assisted Language Learning*, *vol 16*, pp259-266.

Cutshall, S., (2007), "Riding the Wave: the surge in interest in Arabic language learning", *Language Educator*, vol 2, pp32-39.

Denis, B., Watland, P., Pirotte, S. & Verday, N., (2004), "Roles and Competencies of the e-Tutor", *Network learning conference*, pp150-157.

Dickinson, M. & Herring, J., (2008), "Developing Online ICALL Exercises for Russian", *3rd Workshop on Innovative use of NLP for Building Educational Applications*, pp1-9.

Dubois, D., Gaha, M., Nkambou, R. & Pierre Poirier, P., (2008), "Cognitive Tutoring System with Consciousness", *Lecture Notes in Computer Science*, pp803-806, Springer.

Dunn, R. & Honigsfeld, A., (2009), "Differentiating Instruction for At-risk Students: What to do and how to do It", pp7-23, Rowman & Littlefield publishing Group, USA.

El-Barouki, F., (2008), "Developing a Learner-Centred Curriculum: The New Arabic Basic Course", *Dialogue on Language Instruction*, vol 19, pp11-22.

Esposito, F., Licchelli, O. & Semeraro, G., (2004), "Discovering Student Models in e-learning Systems", *Journal of Universal Computer Science*, vol 10, pp47-57.

Farghaly, A. & Shaalan, K., (2009), "Arabic natural language processing: Challenges and Solutions", *ACM transactions on Asian language information processing (TALIP)*, vol 8, No.14, pp1-22, USA.

Feng, M., Beck, J., Heffernan, N. & Koedinger, K., (2008), "Can an Intelligent Tutoring System Predict Math Proficiency as Well as a Standardized Test", *Proceedings of the 1st International Conference on Education Data Mining*, pp107-116, Montreal.

Fossati, D., (2008), "The role of positive feedback in Intelligent Tutoring Systems", *Proceedings of the ACL-08: HLT Student Research Workshop*, pp31–36, Columbus, OH.

Galeev, I., Mustaphin, R. & Ardil, G., (2006), "A New Method of Adaptation in Integrated Learning Environment", *World academy of science, Engineering and Technology*, vol 14, pp27-33.

Gamper, J., Knapp, J., (2002), "A Review of Intelligent CALL Systems", *Computer Assisted Language Learning*, vol 15, pp329-342, Routledge, part of the Taylor & Francis Group.

Godwin-Jones, R., (2007), "Emerging technologies: tools and trends in self-paced language instruction", *Language Learning & Technology*, vol 11, pp10-17.

Godwin-Jones, R., (2009), "Emerging Technologies focusing on form: tools and strategies"*, Language Learning & Technology,* vol 13, pp5-12.

Gregory, G., & Chapman, C. (2007), *"*Differentiated instructional strategies. One size doesn't fit all", pp47-133, Corwin Press, Thousand Oaks, California.

Guo, Y., Beck, J. & Heffernan' N., (2008), "Trying to Reduce Bottom-out hinting: Will telling student how many hints they have left help", *Lecture Notes in Computer Science*, pp774-778, Springer.

Haake, M. & Gulz, A., (2008), "Visual Stereotypes and Virtual Pedagogical Agents", *Educational Technology & Society*, vol *11*, pp1–15.

Hartley, J. & Sleeman, D., (1973), "Toward more intelligent teaching systems", *International Journal of Man-Machine studies*, pp215-236.

Heaton, J., (2008), "Introduction to neural networks with Java", pp143-173, Heaton Research.

Heift, T., (2004), "Corrective feedback and learner uptake in CALL", *ReCALL*, vol 16, pp416-431.

Hrastinski, S., (2008), "Asynchronous and Synchronous E-learning", EDUCAUSE quarterly magazine, vol 31, pp51-55.

Jeremic, Z., Jovanovic, J., & Gasevic, D., (2009), "Evaluating an Intelligent Tutoring System for Design Patterns: the DEPTHS Experience", *Educational Technology & Society*, vol *12*, pp111–130.

Jia, J., (2004), "CSIEC (computer simulator in educational communication): A virtual context-adaptive chatting partner for foreign language learners", *Proceedings of the IEEE conference on advanced learning technologies*, pp690–692.

Ji-lin, C., Yuan-long, H., Zong-yi, X., Li-min, J. & Zhong-zhi, T., (2006), "A Multi-objective Genetic-based Method for Design Fuzzy Classification Systems", *IJCSNS International Journal of Computer Science and Network Security*, vol 6, No. 8, pp110-117.

Johnson, L. & Valente, A., (2008), "Tactical Language and Culture Training Systems: Using Artificial Intelligence to Teach Foreign Languages and Cultures", *Proceedings*

of the 20th national conference on Innovative applications of artificial intelligence, vol 3, pp1632-1639*.*

Kazi, H., Haddawy, P. & Suebnukarn, S., (2007), "Towards Human-Like Robustness in an Intelligent Tutoring System", *Proceedings of 8th International Conference on Cognitive Modeling,* pp247-252*.*

Khribi, M., Jemni, M., & Nasraoui, O., (2009), "Automatic Recommendations for E-Learning Personalization Based on Web Usage Mining Techniques and Information Retrieval", *Educational Technology & Society*, vol *12*, pp30–42.

Kirschner, P., Sweller, J. & Clark, R., (2006), "Why minimal guidance during instruction does not work: an analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching", *Educational Psychologist,* vol. 41, pp75–86.

Koedinger, K., Aleven, V., Heffernan, N., McLaren, B. & Hockenberry, M., (2004), "Opening the Door to Non-Programmers: Authoring Intelligent Tutor Behavior by Demonstration"*,* In *Proceedings of 7th Annual Intelligent Tutoring Systems Conference,* Maceio, Brazil*,* pp162-173.

Kong, S., Ogata, H., Arnseth, H., Chan, C., Hirashima, T., Klett, F., Lee, J., Liu, C., Looi, C., Milrad, M., Mitrovic, A., Nakabayashi, K., Wong, S. & Yang, S., (2009), "A Concept Based Digital Study Guide and Annotation System on Paper Textbook", *Proceedings of the 17th International Conference on Computers in Education,* pp399-406*,* Hong Kong: Asia-Pacific Society for Computers in Education.

Levy, M. & Stockwell, G., (2006), "CALL Dimensions: Options and Issues in Computer-Assisted language learning", pp84-226, Lawrence Erlbaum Associates Inc., Mahwah, New Jersey.

Lloyd, N., Heffernan, N. & Ruiz, C., (2007), "Predicting student engagement in intelligent tutoring systems using teacher expert knowledge", *Educational*

DataMining Workshop held at the 13th Conference on Artificial Intelligence in Education, pp40-49.

Luppicini, R., (2007), "Online Learning Communities", pp3-169, Information Age Publishing.

Makatchev, M., VanLehn, K., Jordan, P. & Pappuswamy, U., (2006), "Representation and reasoning for deeper natural language understanding in a physics tutoring system", *Proceedings of the 19th International FLAIRS conference*, pp682-687, Menlo Park, CA: AAAI Press.

Mamadani, E. & Assilian, S., (1975),"An experiment in linguistic synthesis with a fuzzy logic controller", *International Journal of Man-Machine studies*, vol 7, pp1-13.

Matsuda, N. & VanLehn, K., (2003), "Modelling Hinting Strategies for Geometry Theorem Proving", *Proceedings of 9th International Conference on User Modelling*, pp373-377, Johnstown, PA, USA.

Mavrikis, M., (2008), "Data-driven modelling of students' interactions in an ILE", *1st International Conference on Educational Data Mining (EDM08)*, pp87-96, Montreal, Canada.

Mills, C. & Dalgarno, B., (2007), "A Conceptual Model for Game-Based Intelligent Tutoring System in ICT: Providing choices for learners and learning*", Proceedings of ASCILITE conference,* pp692-702*,* Singapore.

Mitrovic, A., Martin, B., Suraweera, P., (2007), "Intelligent tutors for all: Constraint-based modeling methodology, systems and authoring", IEEE Intelligent Systems, special issue on Intelligent Educational Systems, vol 22, No. 4, pp38-45.

Mitrovic, A., Suraweera, P., Martin, B. & Weerasinghe, A., (2004), "DB-suite: Experiences with Three Intelligent, Web-based Database Tutors", *Interactive Learning Research*, vol 15, No. 4, pp409-432.

Murray, T., Ainsworth, S. & Blessing, S., (2003), "Authoring Tools for Advanced Technology Learning Environments", Kluwer Academic Publishers, Netherland, pp493-546.

Murray, R. & VanLehn, K., (2005), "Effects of dissuading unnecessary help requests while providing proactive help", *Proceedings of the 12th International Conference on Artificial Intelligence In Education*, pp887–889, Amsterdam.

Murphy, P. (2007), "Reading comprehension exercises online: The effects of feedback, proficiency and interaction", *Language Learning & Technology*, vol *11,* No. 3, pp107-129.

Nash, R., (2005), "Course completion rates among distance learners: Identifying possible methods to improve retention", *Journal of Distance Learning Administration*, vol 8, no. 4.

Nedic, Z., Nedic & V., Machotka, J., (2002), "Intelligent Tutoring System for teaching 1st year engineering", *World Transactions on Engineering and Technology Education*, vol1, No. 2, pp241-243.

Negnevitsky, M., (2005), "Artificial intelligence a guide to intelligent systems", Second edition, pp87-125, Addison-Wesley.

Negoita, M. & Pritchard, D., (2004), "Using a virtual student model for testing intelligent tutoring systems", *Interactive Technology and Smart Education*, vol 1, pp195–203, Troubador Publishing Ltd.

Nejdl, W., Kay, J., Pu, P. & Herder, E., (2008), "Adaptive Hypermedia and Adaptive Web-Based Systems", *Lecture Notes in Computer Science*, pp 275-278, Springer.

Nielsen, H. & Carlsen, M., (2003), "Interactive Arabic Grammar on the Internet: Problems and Solutions", *Computer Assisted Language Learning*, vol 16, pp95-112.

Qu, L. & Johnson, W., (2005), "Detecting the learner's motivational states in an interactive learning environment", *Artificial intelligence in education: Supporting learning through intelligent and socially informed technology*, pp547-554, IOS Press.

Razzaq, L., Heffernan, N., (2009), "To Tutor or Not to Tutor: That is the Question", *Proceedings of the Conference on Artificial Intelligence in Education*, pp457-464.

Rishi, O., Govil, R., (2008), "DCBITS: Distributed Case Based Intelligent Tutoring System", *Proceedings of World Congress on Engineering and Computer Science*, vol 1007, pp162-176.

Roselli, R., Gilbert, S., Howard, L., Blessing, S., Raut, A. & Pandian, P., (2008), "Integration of an Intelligent Tutoring System with a Web-based Authoring System to Develop Online Homework Assignments with Formative Feedback", *Proceedings of the American Society for Engineering Education Annual Conference*, pp4603-4608, USA.

Samuelis, L., (2007), "Notes on the Components for Intelligent Tutoring Systems", *Acta Polytechnica Hungarica,* vol 2, pp77-85.

Sanchez-Torrubia, M., Torres-Blanc, C., Krishnankutty, S., (2008), "Mamdani's fuzzy inference eMathTeacher: a tutorial for active learning", *WSEAS Transactions on Computers*, vol 7, pp363-374, WSEAS.

SCORM 2004 4th Edition Documentation Suite Public Draft, version 1.1, (n.d), available from: www.adlnet.gov/Technologies/scorm [Accessed 6th April 2010].

Shaalan, K., (2005), "An Intelligent Computer Assisted Language Learning System for Arabic Learners", *Computer Assisted Language Learning*, vol 18, pp81–109, Rutledge, part of the Taylor & Francis Group.

Shin, N. & Chan J., (2004), "Direct and indirect effects of online learning on distance education", *British Journal of Educational Technology*, vol 35, No. 3, pp275-288, Blackwell Publishing.

Simpson, O., (2002), "Supporting Students in Open and Distance Learning", pp9-50, Taylor & Francis Ltd, London: Kogan.

Sivanandam, S., Sumathi, S. & Deepa, S., (2007), "Introduction to fuzzy logic using MATLAB", pp96-120, Springer.

Stathacopoulou, R., Grigoriadou, M., Samarakou, M. & Denis Mitropoulos, D., (2007), "Monitoring students' actions and using teachers' expertise in implementing and evaluating the neural network-based fuzzy diagnostic model", *Expert Syst. Appl.*, vol 32, No. 4, pp955-975.

Stathacopoulou, R., Magoulas, G., Grigoriadou, M. & Samarakou, M., (2005), "Neuro-fuzzy knowledge processing in intelligent learning environments for improved student diagnosis", *Information Sciences*, pp273-307.

Stathacopoulou, R., Samarakou, M., Grigoriadou, M., George, D. & Magoulas, G., (2004), "A Neuro-Fuzzy Approach to Detect Student's Motivation", *Fourth IEEE International Conference on Advanced Learning Technologies, IEEE Computer Society*, pp71-75.

Stamper, J., Barnes, T. & Croy, M., (2007), "Extracting Student Models for Intelligent Tutoring Systems", *Proceedings of the 22nd national conference on Artificial intelligence*, vol 2, pp1900-1901, Vancouver, British Columbia, Canada.

Wheeler, S., Yeomans, P. & Wheeler, D., (2008), "The Good, the Bad and the Wiki: Evaluating Student-Generated Content for Collaborative Learning", *British Journal of Educational Technology*, vol 39, pp987-995.

Tsiriga, V. & Virvou, M., (2002), "Initializing the Student Model using Stereotypes and Machine Learning", *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, vol 2, pp410-415.

Turksen, B., (2005), "An Ontological and Epistemological Perspective of Fuzzy Set Theory", pp145-213, Elsevier Science, Netherland.

Turner, T., Macasek, M., Nuzzo-Jones, G. & Heffernan, N., (2005), "The assessment builder: A rapid development tool for ITS", *Proceedings of the 12th Annual Conference on Artificial Intelligence in Education*, pp929–931.

U.S. Department of Education, (n.d.), available from: www.ed.gov/index.jhtml [Accessed 28th Feb 2009].

Ushida, E., (2006), "Robo-Sensei Personal Japanese Tutor", *CALICO Software Review*, vol 24, No. 2, pp408-420.

Van de Sande, C. & Leinhardt, G., (2007), "Help! Active Student Learning and Error Remediation in an Online Calculus e-Help Community", *Electronic Journal of e-Learning*, vol 5, pp227-238, Academic Conferences Ltd.

VanLehn, K., Lynch, C., K. Schulze, K., Shapiro, J., Shelby, R., Taylor, L., Treacy, D., Weinstein, A., & Wintersgill, M., (2005), "The Andes Physics Tutoring System: Lessons Learned", *International Journal of Artificial Intelligence in Education*, vol 15, No. 3, pp147-204.

Veermans, K., De Jong, T. & Van Joolingen, W., (2006), "Using heuristics to facilitate discovery learning in a simulation learning environment for a physics domain", *International Journal of Science Education*, vol 21, pp341-361.

Virvou, M., Katsionis, G. & Manos, K., (2005), "Combining Software Games with Education: Evaluation of its Educational Effectiveness", *Educational Technology and Society*, vol 8, pp54-65.

Virvou, M. & Tsiriga, V., (2004), "A Framework for the Initialization of Student Models in Web based Intelligent Tutoring Systems", *User Modelling and User-Adapted Interaction*, vol 14, pp289 -316.

Wagner, N., Hassanein, K. & Head, M., (2008), "Who is responsible for E-learning success in Higher Education? A Stakeholder Analysis", *Educational Technology & Society*, vol 11, pp26-36.

Wang, H. & Chen, S., (2007), "Artificial Intelligence Approach to Evaluate Students' Answerscripts Based on the Similarity Measure between Vague Sets", *Educational Technology & Society*, vol 10, pp224-241.

Walonoski, J. & Heffernan, N., (2006), "Prevention of Off-Task Gaming Behaviour in intelligent tutoring Systems", *Lecture Notes in Computer Science*, pp722-724, Springer.

Weerasinghe, A., Mitrovic, A., (2006), "Facilitating deep learning through self-explanation in an open-ended domain", *KES Journal*, vol 10, pp3-19.

APPENDICES

APPENDIX A.   The learners and teachers questionnaires

APPENDIX B.   The paired t-test

APPENDIX C.   Sample of log files for two learners

APPENDIX A. The learners and teachers questionnaires

A.1. The learners' questionnaires

	Questions
1	It was easy to learn to use this system?
2	The information provided by the system is easy to understand?
3	It is easy to find the information I needed?
4	The interface of the system is clear and pleasing?
5	It was easy to learn to use this system?
6	The information provided by the system is easy to understand?
7	It is easy to find the information I needed?
8	The interface of the system is clear and pleasing
9	The system's support (e.g. feedback, hints, explanations and thoughts) is clear
10	The system's support is effective in helping me complete the tasks and lessons
11	The system's support helped me to become productive quickly
12	The system gives messages that clearly tell me how to fix problems in my knowledge
13	Overall, I am satisfied with the system's support
14	I am comfortable using the Internet for various applications
15	I am able to access the Internet as needed
16	I have access to updated computer and fast Internet
17	I usually use the Internet for educational purposes
18	I have capabilities to view multimedia contents on my computer
19	I am able to manage my study time on the Internet more effectively
20	I possess sufficient computer skills for doing online work
21	I feel comfortable communicating online in English
22	I believe that learning on the Internet outside of class is more motivating than a regular course
23	I believe a complete course can be given through the Internet without difficulty.
24	I could pass a course on the Internet without any teacher assistance
25	I believe an Internet course is possible but for learning language it would be difficult.
26	Studying the Practices was very useful in the learning process
27	Studying the Extra tutoring was very useful in the learning process
28	Studying the Pre-requisites was very useful in the learning process
29	The Questions and Examinations set by the tutoring system were realistic
30	I recommend this tutoring system to anyone who wants to learn a new language
31	I believe what I have learned will benefit me in the future
32	I found learning a language using this system is useful
33	This system has all the functions and capabilities I expect it to have
34	The discovery learning mode was very helpful?
35	What are your computer skills?
36	What is your level of satisfaction with the tutoring system?
37	Did you obtain the assistance you required during the learning process?
38	Was the Learning received adequate to answer the Questions and Examination?
39	Did the tutoring system respond positively to your needs?
40	Was the system feedback helpful?
41	Were the system Hints helpful?
42	Was the system comments editor control helpful?

Table A.1: The full list of the learners' questionnaire's questions

A.2. The teachers' questionnaires

	Questions
1	The system contributes to making the teaching interesting
2	Theoretical concepts are adequately complemented by examples, exercises, problems, etc
3	The course was entirely covered
4	The materials were organized in logical sequence
5	The system explained the topics effectively
6	The topics were simplified for better understanding?
7	The audio-visual presented by the system were effective
8	The voice was clearly audible
9	The language pronunciations were easy to understand
10	Reading text on the tutoring system's pages was very easy
11	Organization of the materials on the tutoring system's pages was very clear
12	Prompts for input were very clear
13	The tutoring system speed was slow
14	The messages presented by the tutoring system were very clear
15	I think I would like to use the tutoring system frequently
16	I thought the tutoring system was easy to use
17	I think I need a human assistance to help me use the tutoring system
18	I found various functions in the tutoring system are well integrated
19	I felt very comfortable using the tutoring system
20	It is necessary to provide an introduction session by human about how to use the tutoring system
21	I need to learn a lot about the tutoring system before I could effectively use it
22	I recommend the system to be used for teaching foreign languages
23	I usually use the Internet for educational purposes
24	I possess sufficient computer desktop skills
25	I feel comfortable communicating online in English
26	I believe a complete language teaching course can be given online
27	The system Practices were very useful in the learning process
28	The system Extra tutoring was very useful in the learning process
29	The system Pre-requisites were very useful in the learning process
30	The Questions and Examinations set by the system were realistic
31	How useful was modifying the instructional strategies during the learning process
32	Did this system has all the functions and capabilities you expect it to have
33	What is your English language skills
34	What is your level of satisfaction with the system?
35	Was the Learning provided adequate to answer all Questions and Examinations?
36	Did the tutoring system respond positively to the learners' needs?
37	Were the system thoughts controls helpful in the learning process?

Table A.2: The full list of the teachers' questionnaire's questions

	Questions
38	Were the system explanations helpful in the learning process?
39	The teacher's page was very helpful
40	The teacher's page was very easy to use
41	I need human assistance to explain to me how to use the teacher's page
42	The teacher's page provided me with all the functions I needed
43	Which feature of the system do you think can help the teacher the most?
44	Which feature of the system do you think can help the learner the most?

Table A.2: The full list of the teachers' questionnaire's questions

APPENDIX B. The paired t-test

The paired t-test is used to compare two population means in which you have two samples of observations are paired together (e.g. learners' test results before and after a particular course). The aim of the test is to find out if the tutor system leads to improvements in learners' knowledge/skills (i.e. test results).

Ho (Null hypothesis): the two means are the same.
H1 (Alternative hypothesis): the two means differ.

Procedure for carrying out a paired t-test

Let *x* = test results before the course, *y* = test results after the course, and *n* is the number of learners. To test the hypothesis, the procedure is as follows:
  1. Calculate the difference ($di = yi - xi$) between the two observations.
  2. Calculate the mean difference, μ.
  3. Calculate the standard deviation of the differences, S_d.
  4. Calculate the standard error of the mean difference, *SE*.

$$SE = \frac{S_d}{\sqrt{n}}$$

  5. Calculate the t-statistic as follow:

$$T_c = \frac{\mu}{SE}$$

  6. Use tables of the t-distribution to calculate the tabulated T_t. Under the null hypothesis, this statistic follows a t-distribution with *n−1* degrees of freedom.
  7. Compare your value for T_c to T_t.

APPENDIX C. Sample of log files for two learners

	First Learner	**Second Learner**
1	LEARNER : Learner login Log file opened in : 26-08-2009 LEARNER : First learner	LEARNER : Learner login Log file opened in : 27-08-2009 LEARNER : Second learner SYSTEM : You must take the Placement test LEARNER : Learner started the placement test SYSTEM : The Placement test results is : 20 %
2	SYSTEM : The learner performance is > Underachiever SYSTEM : The learner Help seeking is >Rare SYSTEM : The learner Engagement is >Not engaged	SYSTEM : The learner performance is -> Underachiever SYSTEM : The learner Help seeking is -> Rare SYSTEM : The learner Engagement is -> Not engaged
3	LEARNER : Starts lesson : Introduction to verbs SYSTEM : According to strategy 2 SYSTEM : LESSON USER-INTERFACE DISPALY -> It is mandatory to study the Practice SYSTEM : LESSON USER-INTERFACE DISPALY -> It is mandatory to study the Prerequisites SYSTEM : LESSON USER-INTERFACE DISPALY -> It is mandatory to study the Extra tutoring	LEARNER : Starts lesson : Introduction to verbs SYSTEM : According to strategy 2 SYSTEM : LESSON USER-INTERFACE DISPALY -> It is mandatory to study the Practice SYSTEM : LESSON USER-INTERFACE DISPALY -> STUDYING the pre-requisites is Optional for you SYSTEM : LESSON USER-INTERFACE DISPALY -> It is mandatory to study the Extra tutoring

Table C.1: Sample of log files for two learners

	First Learner	Second Learner
4	LEARNER: Starts the Practice session... SYSTEM : Collecting information about The Practice ... SYSTEM : PRACTICE USER-INTERFACE DISPALY -> Check the answers and click the feedback button to receive feedback LEARNER : submits practice questions answers LEARNER : Learner CLICK 'feedback' BUTTON SYSTEM : PRACTICE USER-INTERFACE DISPALY -> Answers feedback SYSTEM : PRACTICE USER-INTERFACE DISPALY -> Practices with Questions Examples SYSTEM : PRACTICE USER-INTERFACE DISPALY -> Check the answers and click the feedback button to receive feedback LEARNER : submits practice questions answers LEARNER : Learner CLICK 'feedback' BUTTON LEARNER : submits practice questions answers SYSTEM : PRACTICE USER-INTERFACE DISPALY -> Answers feedback SYSTEM : PRACTICE USER-INTERFACE DISPALY -> The first question answer is 3 and your answer was Correct SYSTEM : PRACTICE USER-INTERFACE DISPALY -> The second question answer is 3 but your answer was 2 SYSTEM : PRACTICE USER-INTERFACE DISPALY -> Answers feedback SYSTEM : PRACTICE USER-INTERFACE DISPALY -> Your previous grammar practices is Excellent and your current grammar practices is Excellent, Keep up the good work SYSTEM : PRACTICE USER-INTERFACE DISPALY -> Answers feedback	LEARNER: Starts the Practice session... SYSTEM : Collecting information about The Practice ... SYSTEM : PRACTICE USER-INTERFACE DISPALY -> Check the answers and click the feedback button to receive feedback LEARNER : submits practice questions answers LEARNER : Learner CLICK 'feedback' BUTTON SYSTEM : PRACTICE USER-INTERFACE DISPALY -> Answers feedback SYSTEM : PRACTICE USER-INTERFACE DISPALY -> Practices with Questions Examples SYSTEM : PRACTICE USER-INTERFACE DISPALY -> Check the answers and click the feedback button to receive feedback LEARNER : submits practice questions answers LEARNER : Learner CLICK 'feedback' BUTTON LEARNER : submits practice questions answers LEARNER : Learner CLICK 'feedback' BUTTON SYSTEM : PRACTICE USER-INTERFACE DISPALY -> Answers feedback SYSTEM : PRACTICE USER-INTERFACE DISPALY -> Excellent, you answer all the questions correctly SYSTEM : PRACTICE USER-INTERFACE DISPALY -> Answers feedback SYSTEM : PRACTICE USER-INTERFACE DISPALY -> Your previous grammar practices is Excellent and your current grammar practices is Excellent, Keep up the good work SYSTEM : PRACTICE USER-INTERFACE DISPALY -> Answers feedback

Table C.1: Sample of log files for two learners

	First Learner	Second Learner
4	SYSTEM : PRACTICE USER-INTERFACE DISPALY -> Your previous listening practices is Below average and your current listening practices is very weak, try to do more listening practices LEARNER : Learner CLICK 'back to lesson' BUTTON	SYSTEM : PRACTICE USER-INTERFACE DISPALY -> Your previous listening practices is Below average and your current listening practices is Excellent, you are progressing very well LEARNER : Learner CLICK 'back to lesson' BUTTON
5	LEARNER : Learner starts Question session SYSTEM : LOADING The Questions SYSTEM : The selected Question instructional strategy is : 6 LEARNER : The learner submit ANSWER 1 for Question 1 LEARNER : The learner submit ANSWER 2 for Question 2 LEARNER : The learner submit ANSWER 4 for Question 3 LEARNER : The learner submit ANSWER 1 for Question 4 LEARNER : the answer for Q1 submitted by the learner is correct LEARNER : the answer for Q2 submitted by the learner is correct LEARNER : the answer for Q3 submitted by the learner is correct LEARNER : The answer for Q4 submitted by the learner is incorrect SYSTEM : ALL questions answers are submitted SYSTEM : The final Grade is : 75 % SYSTEM : Analysis of the Answers SYSTEM : your final questions answer grade is Above the average, and from your previous interaction with the tutor system we can conclude that : SYSTEM : your answers to the previous questions is Very poor and you are Hardly ever spent time with the tutor system, nevertheless you Rarely seeks Help from the Tutor system to solve Questions SYSTEM: also you are Hardly ever Log in to the discovery based Learning.	LEARNER : Learner starts Question session SYSTEM : LOADING The Questions SYSTEM : The selected Question instructional strategy is : 6 LEARNER : The learner submit ANSWER 1 for Question 1 LEARNER : The learner submit ANSWER 2 for Question 2 LEARNER : The learner submit ANSWER 4 for Question 3 LEARNER : The learner submit ANSWER 2 for Question 4 LEARNER : the answer for Q1 submitted by the learner is correct LEARNER : the answer for Q2 submitted by the learner is correct LEARNER : the answer for Q3 submitted by the learner is correct LEARNER : the answer for Q4 submitted by the learner is correct SYSTEM : ALL questions answers are submitted SYSTEM : The final Grade is : 100 % SYSTEM : Analysis of the Answers SYSTEM : your final questions answer grade is Excellent, and from your previous interaction with the tutor system we can conclude that : SYSTEM : your answers to the previous questions is Excellent and you are Every now and then spent time with the tutor system, also you Often seeks Help from the Tutor system to solve Questions SYSTEM: also you are Hardly ever Log in to the discovery based Learning. SYSTEM :USER:INTERFACE: HAND clapping Picture
6	SYSTEM : The learner performance is -> Underachiever SYSTEM : The learner Help seeking is -> Rare SYSTEM : The learner Engagement is -> Often engaged	SYSTEM : The learner performance is -> Strong SYSTEM : The learner Help seeking is -> Medium SYSTEM : The learner Engagement is -> Often engaged

Table C.1: Sample of log files for two learners