



THESIS / THÈSE

MASTER IN COMPUTER SCIENCE

A multimedia English course for Chinese students

de Coster, Jean-Gobert

Award date:
2003

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.



Facultés Universitaires Notre Dame de la Paix, Namur
Institut d'informatique

A Multimedia English Course for Chinese Students

Jean-Gobert de Coster

Promotrice : Professeur Monique Noirhomme
Maître de stage : Professeur Peter Kelly

Mémoire présenté en vue de l'obtention du grade de Maître en Informatique
Année académique 2002-2003

Abstract. Qing Ma, a doctoral researcher at the "Ecole des Langues Vivantes" in Namur develops an English learning method designed for Chinese students in order to emphasize on the communicative approach of the language. In this paper I describe how I adapted this method into a multimedia Computer Assisted Language Learning program. I describe the whole designing process from scratch; explaining the discussion process that lead to a final implementation of the program and the interface. I then describe the interface and the testing process with the end users.

Keywords: CALL, language, China, Communication, interface design, English

Résumé. Qing ma, une chercheuse doctorante à l'Ecole des Langues Vivantes à Namur développe une méthode d'apprentissage de l'anglais destinée à des étudiants Chinois afin d'insister sur l'approche communicative du langage. Dans ce document, je décris comment j'ai adapté cette méthode dans un programme d'Apprentissage de Langue Assisté par Ordinateur. Je décris le processus de création à partir du point de départ ; expliquant le processus de discussion qui a mené à l'implémentation finale du programme et de l'interface. Je décris ensuite l'interface et le processus de test avec les utilisateurs finaux.

Mots-clés : Apprentissage des langues assisté par ordinateur, langage, Chine, communication, développement d'interface, anglais.

Acknowledgements

To Professor Monique Noirhomme,
for the supervision of this work and for her great help;

To Professor Peter Kelly,
for proposing me this subject and for his great help and understanding;

To Qing Ma,
for the great times I had in collaborating with her on her project;

To Nora Condon,
for her invaluable help, especially for the final correction of this paper;

To all the other members of the ELV I met during the various meetings,
for all I learned from them during these meetings;

To all my friends, and especially Emmeline, Cedric, Hugues, Régis, Mathieu,
Thibaut, Marie and Audrey,
for caring so much about my progress and for always being there when I needed
them;

To my family,
For their support, understanding and patience

To my classmates,
for all the good times we had during the run of our studies.

Table of contents

1	Introduction	10
2	Computer Assisted Language Learning	12
2.1	CALL History	12
2.2	CALL State of the Art	14
2.2.1	<i>Text-to-Speech Synthesis</i>	15
2.2.2	<i>Automatic Speech Recognition (ASR)</i>	16
2.2.3	<i>Broadband Multicast videoconferencing</i>	17
2.3	CALL design principles.....	17
2.3.1	<i>The Role of Metaphor</i>	17
2.3.2	<i>Design of the Main Screen</i>	18
2.3.3	<i>Design of Icons</i>	19
2.3.4	<i>CALL Efficiency Model</i>	19
3	Context Analysis	21
3.1	Wufun	21
3.1.1	<i>Objective of the Program</i>	21
3.1.2	<i>Tasks</i>	22
3.2	Users' Profile	22
3.2.1	<i>The users' English level</i>	23
3.2.2	<i>The users' Experience of CALL</i>	23
3.2.3	<i>Users' motivation</i>	25
3.3	Computer equipment.....	25
4	Specifications	26
4.1	Introduction.....	26
4.2	The testing unit.....	26
4.2.1	<i>First Page: Overview of the story</i>	26
4.2.2	<i>Second Page: Word Focus (WF)</i>	27
4.2.3	<i>Third Page: The Story</i>	27
4.2.4	<i>Fourth Page: Word Memorization Aids (WMA)</i>	28
4.2.5	<i>Pages 5-10: Exercises</i>	28
4.2.6	<i>Page 11: Idioms and proverbs</i>	30
4.2.7	<i>Page 12: Humor and true stories</i>	30
4.2.8	<i>Page 13: Small dictation exercise</i>	31
4.3	Users' freedom	31
4.4	Marking system	32
4.5	User information	33
4.5.1	<i>General information</i>	34
4.5.2	<i>Unit information</i>	35
4.5.3	<i>Page Information</i>	35
4.5.4	<i>Try information</i>	36
4.5.5	<i>Final structure</i>	37
4.6	Audio support	38

4.7	Helping the user	38
4.8	Analyzing the results	39
5	Functional Analysis	40
5.1	Introduction.....	40
5.2	Tasks.....	40
5.2.1	<i>User Identification</i>	41
5.2.2	<i>User creation</i>	42
5.2.3	<i>Navigation</i>	43
5.2.4	<i>Unit Selection</i>	44
5.2.5	<i>Page Selection</i>	45
5.2.6	<i>Page Interaction</i>	46
5.2.7	<i>View Instructions</i>	48
5.2.8	<i>View Word Focus</i>	49
5.2.9	<i>View Word Memorization Aids</i>	50
5.2.10	<i>Do an Exercise</i>	51
5.2.11	<i>View the Idioms</i>	53
5.2.12	<i>View the Humor and True Stories</i>	54
5.2.13	<i>Do the Dictation Exercise</i>	55
5.3	Profile Management	56
5.3.1	<i>CheckLoginInformation</i>	56
5.3.2	<i>LoadProfile</i>	56
5.3.3	<i>CheckUserInformation</i>	57
5.3.4	<i>CreateProfile</i>	58
5.3.5	<i>SetLastUnit</i>	59
5.3.6	<i>UpdateLastAccessiblePage</i>	59
5.3.7	<i>SaveTry</i>	59
5.3.8	<i>GetInformation</i>	60
5.3.9	<i>SetInformation</i>	60
5.4	Navigation Management	60
5.4.1	<i>LoadLastUnit</i>	61
5.4.2	<i>LoadUnit</i>	61
5.4.3	<i>LoadPage</i>	61
5.4.4	<i>DisplayTranslation</i>	61
5.4.5	<i>DisplayFocusWord</i>	62
5.4.6	<i>DisplayWordMemo</i>	62
5.4.7	<i>DisplayTryRecord</i>	62
5.4.8	<i>DisplayCorrectAnswers</i>	62
5.4.9	<i>DisplayIdiom</i>	63
5.4.10	<i>DisplayAnimatedMan</i>	63
5.4.11	<i>DisplayHumorStory</i>	63
5.4.12	<i>DisplayDictation</i>	63
5.5	Sound Management.....	64
5.5.1	<i>PlaySound</i>	64
5.5.2	<i>TestAllSoundsListened</i>	64
5.6	Exercise Management.....	65
5.6.1	<i>CreateTryRecord</i>	65
6	Technical choices and implementation issues	66
6.1	Choice of the Programming Environment	66
6.1.1	<i>Object Oriented</i>	66

Table of Contents

6.1.2	<i>Choice of the Language</i>	66
6.1.3	<i>Operating system</i>	67
6.1.4	<i>NetBeans IDE</i>	67
6.2	Data Structure	67
6.2.1	<i>The Profile</i>	67
6.2.2	<i>The Units</i>	68
6.2.3	<i>The sounds</i>	68
6.2.4	<i>Storing the data</i>	69
6.3	Problems Encountered	70
6.3.1	<i>Time Management</i>	70
6.3.2	<i>Mastering the NetBeans IDE</i>	70
6.3.3	<i>Drag and Drop</i>	71
6.3.4	<i>Fill In the Blanks</i>	71
6.3.5	<i>Checking the Answers</i>	71
7	Interface	72
7.1	<i>Choice of a metaphor</i>	72
7.2	<i>Graphic choices</i>	73
7.2.1	<i>Colors</i>	73
7.2.2	<i>Window size</i>	73
7.3	<i>Icons</i>	73
7.4	<i>Welcome Screen</i>	76
7.5	<i>User Identification</i>	77
7.6	<i>User Creation</i>	78
7.7	<i>The Main Menu Screen</i>	79
7.8	<i>Navigating the Unit</i>	81
7.9	<i>The Pages</i>	83
7.9.1	<i>Overview of the story</i>	83
7.9.2	<i>Word Focus</i>	84
7.9.3	<i>The Story</i>	85
7.9.4	<i>Word Memorization aids</i>	86
7.9.5	<i>Exercise 1</i>	86
7.9.6	<i>Exercise 2</i>	87
7.9.7	<i>Exercise 3</i>	87
7.9.8	<i>Exercises 4 and 5</i>	87
7.9.9	<i>Exercise 6</i>	88
7.9.10	<i>Idioms and Proverbs</i>	89
7.9.11	<i>Humor and true stories</i>	90
7.9.12	<i>Dictation Exercise</i>	90
8	Evaluation	91
8.1	<i>Introduction</i>	91
8.2	<i>Testing procedure</i>	91
8.3	<i>The Testing</i>	92
8.3.1	<i>Results of the receptive test</i>	92
8.3.2	<i>Results of the productive test</i>	93
8.3.3	<i>The usage of the program</i>	93

Table of Contents

8.4	The results of the questionnaire	94
8.4.1	<i>The design of the exercises</i>	95
8.4.2	<i>The scoring and answers checking system</i>	95
8.4.3	<i>The help and instructions system</i>	95
8.4.4	<i>The users' freedom</i>	96
8.4.5	<i>Interface design</i>	96
8.4.6	<i>Graphic design</i>	96
8.4.7	<i>Sounds</i>	96
8.4.8	<i>Overall Satisfaction</i>	96
8.5	Conclusion.....	97
9	Conclusion	98
10	Bibliography	99
11	Annexes	100

Index of Tables and figures

<i>Fig 2-2 : CALL efficiency model</i>	19
<i>Fig 3-1 : Users' English Level</i>	23
<i>Fig 3-2 : Users' Experience of CALL</i>	23
<i>Fig 3-3 : Material Used for Language Learning</i>	24
<i>Fig 3-4 : Users' Motivation to Use CALL</i>	25
<i>Table 3-1 : Three Gorges University's Computer Equipment</i>	25
<i>Fig 4-1 : structure of a profile</i>	33
<i>Fig 4-2 : A profile's data representation</i>	37
<i>Fig 5-1 : The USER_IDENTIFICATION task</i>	41
<i>Fig 5-2 : The USER_CREATION task</i>	42
<i>Fig 5-3 : The NAVIGATION task</i>	43
<i>Fig 5-4 : The UNIT_SELECTION task</i>	44
<i>Fig 5-5 : The PAGE_SELECTION task</i>	45
<i>Fig 5-6 : The PAGE_INTERACTION task</i>	47
<i>Fig 5-7 : The VIEW_INSTRUCTIONS task</i>	48
<i>Fig 5-8 : The VIEW_WORDFOCUS task</i>	49
<i>Fig 5-9 : The VIEW_WORDMEMO task</i>	50
<i>Fig 5-10 : The DO_EXERCISE task</i>	52
<i>Fig 5-11 : The VIEW_IDIOMS task</i>	53
<i>Fig 5-12 : The VIEW_HUMOR task</i>	54
<i>Fig 5-13 : The DO_DICTATION task</i>	55
<i>Table 6-1 :Java Vs C++</i>	66
<i>Fig 7-1 : Help display</i>	75
<i>Fig 7-2 : Quit wufun</i>	75
<i>Fig 7-3 Logout</i>	76
<i>Fig 7-4 : Translation</i>	76
<i>Fig 7-5 : The welcome screen</i>	76
<i>Fig 7-6 : The Identification page</i>	77
<i>Fig 7-7 : Login Failure</i>	77
<i>Fig 7-8 : Wrong Password</i>	77
<i>Fig 7-9 the user creation page</i>	78
<i>Fig 7-10 : User Creation errors</i>	79
<i>Fig 7-11 New user created</i>	79
<i>Fig 7-12 : The main menu as seen when first beginning a unit</i>	80
<i>Fig 7-13 : The main menu when all the pages have been completed</i>	80
<i>Fig 7-14 : the WufunBrowser/ViewerPanel</i>	81
<i>Fig 7-15 the WufunExercisePanel</i>	82
<i>Fig 7-16 : Score display</i>	83
<i>Fig 7-17 : Overview of the story</i>	84
<i>Fig 7-18 : The word focus page</i>	84
<i>Fig 7-19 : The popup word focus</i>	85
<i>Fig 7-20 : word memorization aids</i>	86

<i>Fig 7-21 Exercise 1</i>	86
<i>Fig 7-22 Exercise 2</i>	87
<i>Fig 7-23 Exercise 3</i>	87
<i>Fig 7-24 : Exercise 4</i>	88
<i>Fig 7-25 Exercise 5</i>	88
<i>Fig 7-26 Exercise 6</i>	89
<i>Fig 7-27 Idioms and Proverbs</i>	89
<i>Fig 7-28 : Humor and true stories</i>	90
<i>Fig 7-29 Dictation Exercise</i>	90
<i>Table 8-1 : Results of the receptive test</i>	92
<i>Table 8-2 : Results of the productive test</i>	93
<i>Table 8-3 : Usage of the program</i>	94

1 Introduction

Language learning is in constant evolution, thanks to the important research being done in the field all over the world. Whilst technologies evolve, computers also take a growing importance in language learning, and over the last decades, the use of Computer Assisted Language Learning programs has become more and more widespread. Now many schools, all over the world, are equipped in Computer Assisted Language Learning software, as it turns out to be a high motivation factor for the students.

In China, language learning is perceived much differently than in our western culture, mainly because it is not considered as a way to communicate, but rather as a way to obtain a degree. Qing Ma, a doctoral researcher from the Ecole des Langues Vivantes (ELV) in Namur develops a method specially designed for Chinese students, to emphasize more on the communicative approach of language learning. In order for her method to be efficient, it has to rely a lot on the use of sound and image; a written support was already made, but now she strives to test this method on a computer, in order to include the use of various Medias.

My role in this project was to adapt this support into a program, taking advantage of the several multimedia applications of computers. The program should guide the user through the different stages of the learning process, use several Medias to put the words into a context, make sure the learner hears the words' correct pronunciation and give them a base of exercises that can be automatically corrected. Also this program will serve Qing Ma's research purposes, providing statistical information about the use of the program by the students and their results.

This paper will be structured in 7 chapters:

In chapter 2 I will introduce the reader to some background information about Computer Assisted Language Learning.

In chapter 3 I will introduce the program and its tasks, describe the profile of the end users and the equipment the program will be designed for.

In chapter 4 I will explain what were Qing Ma's requirements, as well as explain what solutions were proposed and describe the first major design choices relative to these requirements.

In chapter 5 I will identify the tasks of the program, in accordance with the requirements of chapter 4. This analysis will enable the identification of the main functions that will have to be implemented in the program. These functions will then be formally specified.

In chapter 6 I will explain what were the technical choices for the implementation, and what were the main problems encountered when actually doing the implementation.

In chapter 7 I will describe the design of the interface and present various screen shots that should give the reader a good idea of what the program looks like.

Finally, in chapter 8 I will describe how the program was tested and evaluated with the final users.

2 Computer Assisted Language Learning

2.1 CALL History

CALL stands for Computer Assisted Language Learning. Mike Levy defined CALL as being “*the search and study of applications of the computer in language learning and teaching*” [LEV 1997].

CALL programs made their first apparition in the 60's, in particular with the development of PLATO (Programmed Logic for Automatic Teaching Operations), a mainframe system designed to deliver learning packages to a large number of students, still used today in many establishments. PLATO used an authoring language simply called TUTOR. The invention of the mouse in 1966 and of the microchip in 1969 prepared the field for more popular use of personal computers and opened up new perspectives for CALL programs.

In 1971, the TICCIT (Time-shared, Interactive Computer Controlled Information Television) was initiated at Brigham Young University. As the name states, the system used both computer and television technologies. This system is devised solely around a specific theory of teaching and learning, where the teacher could contribute to the material contents, but not the instructional strategy. This early use of video and audio in language learning makes TICCIT the first ever multimedia Computer Assisted Instruction (CAI) program. The main advantage of this framework is that the teacher does not need to be familiar with any authoring language, only with the instructional design built into the system.

In the late 70's, the School of Language Studies in Ealing College of Higher Education (London) became the NCCALL: National Centre for Computer Assisted Language Learning, and later publishes the first CALL newsletter: CallBoard. So far, the development in CALL had been based on empiricist theories; the creation of a CALL community opened up perspectives for sharing experience and set up more effective theories on CALL. Some new methods began to appear such as Community Language Learning or the Total Physical Response. Those techniques, called humanistic methods, took into account the emotions, feelings and affective dimension of the user.

In the early 80's, the introduction of microcomputer caused a large boom in CALL. Specialized journals like CALICO made their appearance. At that time, many language teachers started getting interested in programming, as microcomputers were often supplied with a version of basic. Whilst in the 60's and 70's CALL programmers would have to have access to a mainframe, and build up programs in a team based effort, CALL could now develop through the sharing of small fragments of CALL concepts and the evolution became more a community effort, the only constraint being the teachers' programming abilities. Also, some authoring languages became a popular pick for teachers to develop their programs. Amongst those is HyperCard, probably one of the first systems that used hypertext (even though this concept was already known in the mid 60's), combined together with Xerox's "Notecards". Other authoring programs like "Storyboard" were based on intelligent text-reconstruction, using clue parts like the title, introduction and textual clues. Those were invaluable tools for the later development of CALL, as they were able to promote language awareness regardless of any specific learning strategy.

Even though a great step forward was made in the designing of CALL programs by individuals with the use of a microcomputer, some mainframe designs were also brought to daylight, such as MIT's (Massachusetts Institute of Technology) Athena Language Learning Project; an eight-year research program on the innovative use of computer in education. Athena was at the basis of the development of the MUSE multimedia authoring environment. MUSE was a basis structure for the building of hypertext and hypermedia systems, to provide cross-referencing of audio, video and graphic material. Athena was also the starting point of some artificial intelligence techniques for natural language processing.

The 90's were the pioneering days of the Internet as a publicly available communication tool. The introduction of "gopher", a program meant to explore the Internet and the release of the World Wide Web by CERN in 1992, brought perspectives of passing multimedia material over long distances. With the appearance of E-mail, the International Email Tandem Network (IETN), an approach to language learning by computer mediated communication using the internet made it possible for universities around the world to pair students in Email tandems and learn languages on a reciprocal basis. With this kind of approach, the learners had to work around their difficulties through analogical reflection and contextual analysis of their conversations rather than being directly corrected by their teachers.

The 90's also saw the creation of the CAMILLE (Computer-Aided Multimedia Interactive Language Learning) consortium, a partnership between British, French, Spanish and Dutch collaborators. The central point in this project is the learning environment. According to the CAMILLE project, language learning is to be considered as a whole, made up of various teaching resources, including a textbook as well as a computing environment, a grammar, a dictionary and multiple media resources. In order to familiarize the student with his working environment, CAMILLE uses the metaphor of a desktop.

By the 90's, computers were fully capable of supporting highly demanding multimedia material at an affordable price, capable of supporting projects like CAMILLE, though the plan to use the Internet as a medium to transfer such heavy data as video is still far from being achieved, as a single picture already takes time to transfer fully. But with the increasing storage capacities on a single computer, new initiatives could be given birth to, such as the Oral Language Archive (OLA). OLA's goal is to establish a collection of digitized sound recordings for foreign language learners, accessible throughout the world via the internet, together with a set of management tools to enable users to easily locate the sounds they need.

Nowadays, numerous new technologies have been developed that have many applications in CALL, we shall discuss these in the next point.

2.2 CALL State of the Art

As we have seen in the historical overview, CALL is in constant evolution in live with to technological improvements. There are many new technologies that could be used in CALL programs, but only few really have reached a critical mass, or widespread use and acceptance by the CALL community at large [LEV 2002]. Written language has been dealt with in CALL for a long time, and spoken language for communication was seldom addressed. Now with the present technologies, the ability to correct this phenomenon is no longer an issue. The current policy of research is to correct this and to make considerable use of the available speech technologies. That is why these technologies are the ones we will be looking at in this chapter.

2.2.1 Text-to-Speech Synthesis

Text-to-Speech Synthesis (TTSS) appeared in the mid 80's. It can be defined as "the automatic production of speech through a grapheme-to-phoneme transcription of a sentence". As we can see in Figure 2-1, the text first goes through a Natural Language Processor (NLP), which uses linguistic formalisms, inference engines and logical inferences to produce a set of phonemes (phonetic transcription of the words read) and a prosody (the desired intonation and rhythm of the words). These will then be processed through a Digital Signal Processor (DSP) that will use different mathematical models, algorithms and diverse computations to produce the final speech.

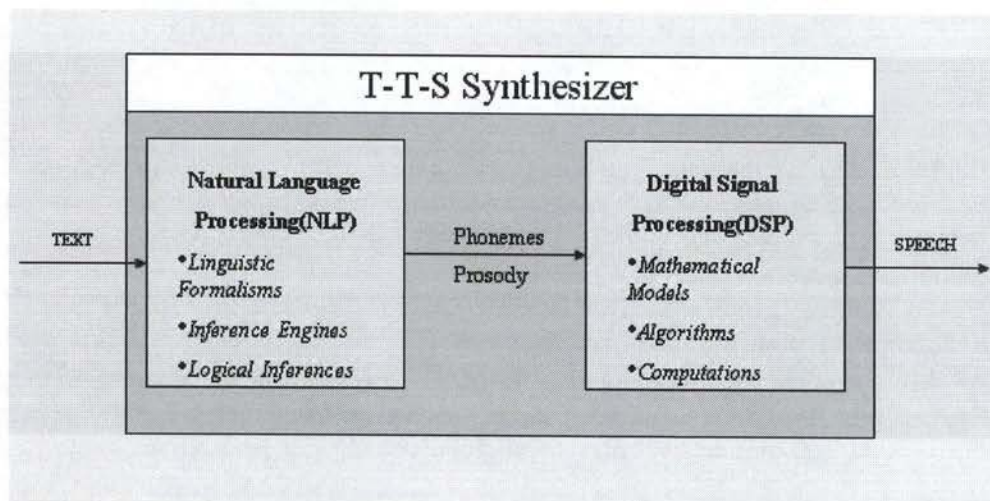


Fig 2-1: Text-to-Speech Synthesis [DUT]

The NLP analyzes the input sequence to parse it into a list of manageable words, determines the speech categories of each of these words in terms of intonation, then compares each word to its neighbors to eliminate certain categories of speech through heuristics and fully associates a general speech category to match the expected prosody. Phonemes can then be generated in various ways: either with a phonetic dictionary based on a lexicon strictly focused on phonemes, or with a rule-based strategy determining, for each letter, what is the best sound-variation to use.

The DSP is based on the idea that natural speech is restricted by articulation constraints, having downsides on the phonetic transition from one phoneme to another. The idea is to simulate the human articulation process, taking those constraints into account, in order to produce an understandable speech. This can be achieved either explicitly by using rules of translation of one phoneme to another, or implicitly by storing examples of transitions in a transition database. The DSP generates segments of speech from the NLP's input, and then concatenates these segments with the chosen transitions. The resulting output is then the synthesized speech.

2.2.2 Automatic Speech Recognition (ASR)

ASR uses complex statistical models, able to characterize the properties of the sounds to be recognized. In order to perform such recognition, the sound input has to go through 5 stages:

Capture and Digitization: The user inputs his speech through a microphone and the ASR Digitization component filters a maximum amount of alien environmental noise, as well as blanks, and converts the speech signal into a digitalized form.

Spectral representation:

At this stage the digitized input is put in a spectrum form in order to better represent the aspects of the signal that are important to distinguish. This is pretty much what the human ear does to distinguish relevant properties of the heard sound.

Segmentation:

There are actually two ways to decompose a speech signal: either by using a frame-based model, or by segmentation. The first technique decomposes the speech into uniform time frames that are then analyzed individually by a statistical model. The second technique splits the signal into phonetic segments and then uses the statistical model to highlight the dynamic properties of each word of the speech. The advantage of this second method is an increase in accuracy of the speech decomposition and in computing speed.

Phonetic Modeling:

Each phonetic segment is captured in a vector depicting the properties of the segments, and then a statistical model computes all the probable distributions of the segment into a multi-dimensional phonetic space. This allows any variation in the signal, such as remaining noise, speaking rates, accents etc., to be taken into account. A better model implies a better recognition.

Search and Match:

The modeled speech can then be matched with the expected speech and can even be analyzed to determine where the flaws in the input speech were.

The effectiveness of speech-interactive CALL is determined less by the capabilities of the speech recognizer than by the design of the language learning activity and feedback, and the inclusion of repair strategies to safeguard against recognizer error.

2.2.3 Broadband Multicast videoconferencing

As we saw in the historical overview, the internet has many uses in language learning, especially the ability to provide interaction between paired students (as for the International Email Tandem Network). At that time, transfer rates did not allow efficient use of video and sound to further enhance this interaction, but now, broadband networks have a much higher capacity to carry heavy information streams such as video. Through the use of web based resources, students can now engage in a collaborative task over a network which offers high quality videoconferencing. In such a design, the key is to use a task-based pedagogy in order to stress even further the idea of reciprocal peer tutoring such as the one we described for the IETN.

2.3 CALL design principles

2.3.1 The Role of Metaphor

When building an interface, the developer translates his conceptual model of the program into an interface. On the other hand, the user will translate the resulting interface into another conceptual model. In order to match up as closely as possible these two models, the developer analyses the design task from a cognitive point of view, applying his knowledge of the human mind to take into account the user's mental representation of the task [LON & VAN 2001]. In order to help the user to interpret the interface correctly, the use of metaphors can play a persuasive role, as the human mind tends to conceptualize complex entities in terms of simpler ones. A usual metaphor for CALL is the use of a mini-world, but other metaphors should not be ruled out, as long as they remain a sound choice in comparison with the researched representation of the task by the user.

According to Colin Lonfils and Johan Vanparys [LON & VAN 2001], the choice of the metaphor “*takes place on the basis of two parameters: (i) compatibility with the user’s habits and expectations, based on his or her knowledge of the real world, and (ii) internal coherence of the mini-world*”. The first point is mainly determined by the icons used in the program and the navigation system, as the second point is determined by the coherence inside the software (avoiding the introduction of representations that are foreign to the metaphor) and the coherence with other software (respecting the standards).

2.3.2 Design of the Main Screen

The main Screen in any CALL program has to meet three requirements [LON & VAN 2001]:

- 1) Give direct access to the main tasks: “*The main screen functions as an entrance to those parts of the software which provide whatever environment is necessary to perform each main task*”. It is thus important to clearly determine what the main tasks are.
- 2) Make the chosen metaphor explicit: The user has to identify the metaphor as soon as possible, the main screen being the first screen the user sees when he enters a program; it is there that this identification has to occur.
- 3) Make the overall structure transparent: The more freedom the user gets, the more he needs to clearly understand the program’s overall structure. The representation of the structure by the user is based on two axes, namely a special representation determined by the motion of the user in the program, and a graphical representation determined by the images used to illustrate the metaphor.

2.3.3 Design of Icons

Icons, as we stated before, are very important to ensure the correct comprehension of the chosen metaphor. Colin Lonfils and Johan Vanparys propose a set of guidelines for the design of icons [LON & VAN 2001]. The goal when choosing designs for icons is to keep the student's focus on the learning process rather than on the understanding of those icons. They have to be kept simple, distinguishable and consistent. If these first main guidelines are followed, the rate of "misfires" (when the user gets an unexpected result of his actions on the program) should be close to nil. Also, the number of icons has to be kept minimal, in order for the user to spot the icon he is looking for in the glimpse of an eye. The meaning associated with buttons also has to be clear and stick as much as possible to the metaphor. For example, if the metaphor is a school, you do not "escape" from class, but you "leave" the room. Other important guidelines are to keep the action related to each icon unique (i.e., no two actions will have the same icon). Finally, the user has to be able to interpret the icon in terms of his knowledge of the real world. This can have consequences on the program's interpretation in various cultures, as the representation of the world can change in different cultures.

2.3.4 CALL Efficiency Model

The user's action on his learning process has a final impact on the efficiency of CALL, as stressed in Qing Ma's CALL efficiency model (Figure 2-2).

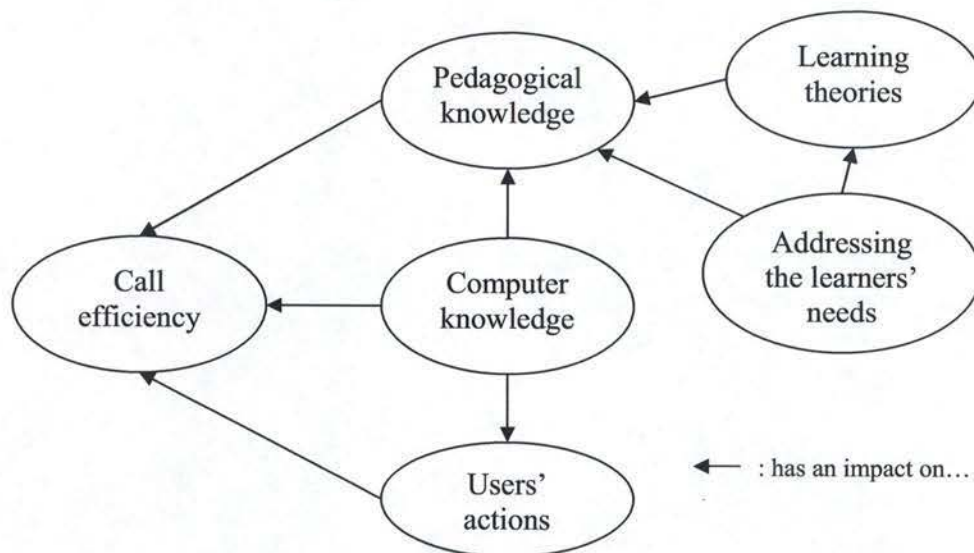


Fig 2-2 : CALL efficiency model

CALL efficiency depends on three factors. First, to have a sound pedagogical underpinning is vital to ensure the quality of a CALL program. This is a commonly agreed phenomenon in CALL research. It has been demonstrated that the quality of a CALL program is determined by the methodology behind it rather than the technology. Two factors have to be considered in determining the methodology, namely the learning theories and the issue to address the learners' needs. Learners' needs will decide what learning theories to choose. Second, technology is also a very important component of the model. Technology enables to realize the pedagogical basis in the method's design; it allows visualizing the design model in a more efficient way. Third, user's actions also have an impact on the quality of the program. If a sophisticatedly designed CALL program is completely under the control of the student, they will do something to harm the learning results, skipping some parts or spending too little time on the program. Therefore, user's action should be controlled to some degree, and the role of technology is to control the user's actions.

3 Context Analysis

3.1 *Wufun*

3.1.1 Objective of the Program

Chinese students see language learning in a very different way to the way we do in Europe. In China, the English exam is a compulsory national exam, and whoever fails that exam will be refused his degree, so language learning is perceived as a way of achieving your studies. In European culture, language learning is seen as a way to communicate more than a way to obtain a diploma; also the chances that a European will have to actually communicate in English in his future life are much higher than those of a Chinese. The language barrier is also greater than for us, because of the total lack of similarity in vocabulary between English and Chinese, whereas French speakers, for instance, find it much easier to memorize words and use them in their right context.

Qing Ma, a Chinese doctoral researcher at the “Ecole des Langues Vivantes” in Namur (ELV), develops a new learning method that attempts to reconcile the learning of the 4000 “focus words” the students have to know in order to pass the exam, and the association of those words with a context (i.e. she is helping the students to pass their exam and to be able to communicate orally). Several memorization aids are also provided for more difficult words. The point is to provide the students with efficient vocabulary learning strategies. This method also emphasizes more the use of audio support in a more or less constraining way. This is very important, because the main goal is to mend the lack of communicative of the language. A computer support seems to be the most appropriate solution to achieve these goals; even more so, as China has always been very receptive to technological advances and especially school computerization. We can therefore assume a high receptivity by the students and to such a program. Teachers are less enthusiastic about this because some of them are afraid of being replaced by machines and to eventually put out of work.

The program will be called *Wufun*, where *Wufun* is the composition of the names of the two Universities at the origin of the project: Wuhan University in China and the FUNDP (Facultés Universitaires Notre Dame de la Paix) in Namur, Belgium.

3.1.2 Tasks

Wufun will consist of several learning units, revolving around a given theme. Each of these units is then made of several pages. These pages will either contain learning material or exercises. The learning material will give the student the information required to do the exercises.

In order to record information about the users' progress, and also to enable Qing Ma to analyze this information in order to evaluate her method, the user will first have to identify himself before entering the program.

Three main tasks can thus be identified that will further be described in the following chapters:

- 1) User Identification
- 2) Navigation through the units and pages
- 3) Interaction with the pages (exercises or learning material)

3.2 Users' Profile

The program will primarily be used as an experimental project by Chinese students. A questionnaire was handed out to 83 students from various orientations in order for Qing Ma to determine what she needed to focus on in her method, and also to know more about the users' knowledge of English and computers. Below I will go over the parts of the questionnaire that were relevant from my point of view as programmer in terms of the future design choices to be made.

3.2.1 The users' English level

To the question "How do you self evaluate your English level", the students answered as shown in Figure 3-1.

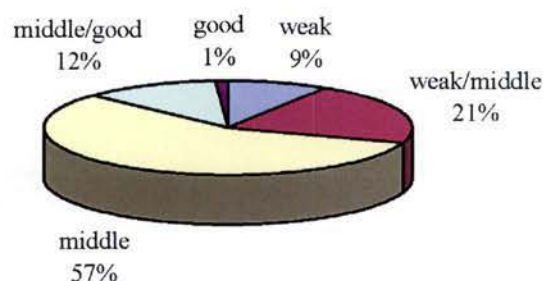


Fig 3-1 : Users' English Level

The questioned users say overall that they have an average knowledge of English. Having an interface in English should thus not cause any trouble for the student, but a Chinese translation of the help system will have to be provided, as well as for the explanation of the exercises.

3.2.2 The users' Experience of CALL

To the question: "How often have you used a computer to learn a foreign language", the students answered as shown in Figure 3-2.

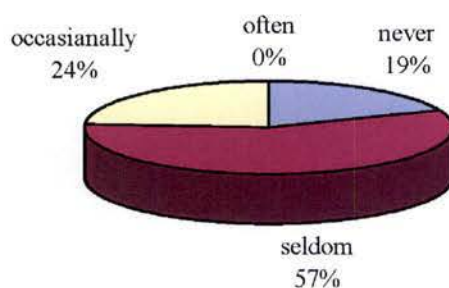


Fig 3-2 : Users' Experience of CALL

An additional question to this one was to know what kind of computer material they had used (the students could choose more than one answer). In Figure 3-3 we can see that the interest for the web as a language learning tool is widespread.

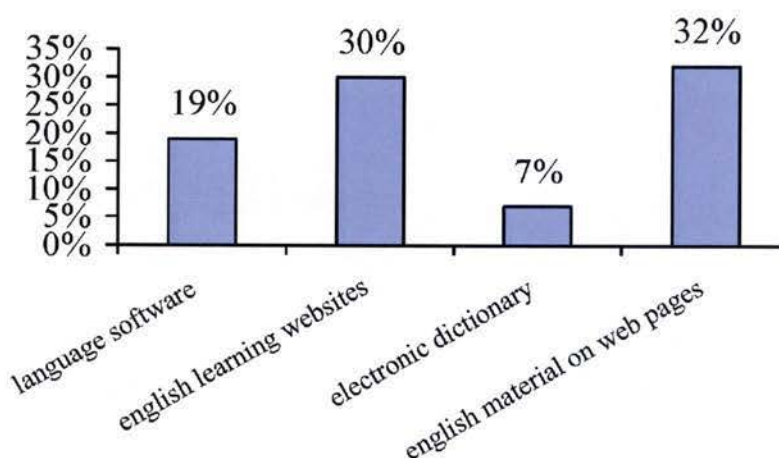


Fig 3-3 : Material Used for Language Learning

Overall, we notice that the users only have a restricted experience of CALL programs. The interface will thus have to be as intuitive as possible and consider that it is the first time the user simply uses a CALL program. An explicit help system should be set up to allow a completely autonomous use of the program.

3.2.3 Users' motivation

To the question: "Would you like to try to study with an English learning software" the users answered as shown on Figure 3-4.

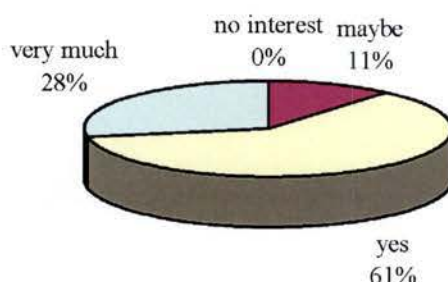


Fig 3-4 : Users' Motivation to Use CALL

We can hereby conclude that our target users are positive that the program will have an impact on their learning and that we can use the program to motivate them to further use the program in the future.

3.3 Computer equipment

The university where the program will be tested is the Three Gorges University. As we can see in table 3-1, it is well equipped with computers. Students taking part in the experiment will each have access to a distinct workstation. Whilst the program is principally designed for experimental use, we have to keep in sight the fact that it is mainly a self-learning method. Thus, as a rule of thumb, we will consider that the program will be used on a personal computer and not in a network environment. Compatibility also has to be a major focus: the program will have to run without any trouble on a mid-range configuration, and remain platform-independent.

Processor type	Memory	Hard Disk space	Operating System
Intel Pentium 4	256 Mb DDR	40 Gb	Windows XP

Table 3-1 : Three Gorges University's Computer Equipment

4 Specifications

4.1 Introduction

Before starting to put the program together, many decisions had to be taken in order to clearly determine what were Qing Ma's needs, how the user would interact with the program, what incentives should be used to motivate the users, and many other aspects of the program. This was mostly done through informal discussions, brainstorming and mutual expertise sharing in the various domains of knowledge involved in the program. The people taking part in these discussions were teachers, doctoral researchers, computer specialists and the director of the multimedia research centre in Namur. My contribution in this process was mainly to explain the possibilities and restrictions of computers in order to accomplish the various tasks required, but also to share my experience as a final user of computer programs.

Most of the decisions made regarding the final specifications of the program were the result of this process of discussion: it greatly contributed to raising essential questions that had a clear impact on the overall structure of the program.

4.2 The testing unit

The program will be used by Qing Ma in order to test her method. We thus restricted the program to only one learning unit, called "A Good Laugh". We will now see exactly how this unit has to be designed and further describe some more of the program's mechanisms, as well as explain the pedagogical objective researched by Qing Ma through each page.

4.2.1 First Page: Overview of the story

Each unit is based on a story that makes use of a certain number of focus words. In order to introduce the user to the story and to some important contextual uses of focus words, a first overview will be presented in the form of a sequence of images accompanied by a spoken text. The user will be able to browse sequentially through the images and listen to the text. Once he has listened to the sound, the user will be allowed to view the written text. What we want here is for the user to actually listen to all the sounds, so he will only be able to proceed to the next page if he has listened to each sound (be it during this session or any previous one).

Pedagogically, the aim of this page is to act as an “advance organizer” (i.e. a device that activates relevant background knowledge). Videos and images are considered effective facilitators of comprehension in second language learning and the retention of meaningful verbal material [MA 2002].

4.2.2 Second Page: Word Focus (WF)

On the second page, the user will be able to view the focus words, listen to them, and see their explanation as well as the Chinese translation. It is important that the user watches those focus words, for they will help him when doing the exercises. The user might want to refer to this page often, so a way to access them easily should be provided on every page of the unit. Once the user has viewed at least one word, he can go on to the next page.

Qing Ma’s pedagogical aim with this page is, in accordance with S. Merlet’s theory, to “*create word representations that will facilitate other levels of processing*” [MA 2002].

4.2.3 Third Page: The Story

On the third page, the user will be presented with the full story with some words missing (the first letter only of these words is given). The user will first have to try to fill in all the gaps without any help, (he still can refer to the word focus page) and then submit his try. Once he has submitted his try, his score will appear and he will be able to listen to each sentence separately in order to correct his performance. Only when he has listened to all the sounds can he gain access to the right answers. In order to move on to the next page, the user only has to have submitted a try, so listening to the sounds is here not mandatory.

The aim of this page is to teach the user to study the context and draw on the information which was presented earlier in the focus words and the overview of the story. Also, as for all exercise pages, the use of sound to further help the user to correct his answers should “*wean the learner away from the habit of heavily relying on written text and [...] foster the habit of listening*” [MA 2002].

4.2.4 Fourth Page: Word Memorization Aids (WMA)

The fourth page provides some hints to help the user remember some words. As English is so very different from Chinese, this is of a great help to the users and it is necessary, just like for word focus, to provide easy access to this page throughout the unit. Also, there may sometimes be a sentence stressing the pronunciation (ex: Damage was done to the dam.). For those sentences, the user has to be able to listen to the sentence by pressing a button. Here too, viewing only one WMA enables the user to proceed to the next page.

Word memorization aids play an important pedagogical role, for they circumvent the language barrier to teach the user how to use words he already knows in order to generate new ones.

4.2.5 Pages 5-10: Exercises

There are various types of exercises in Wufun, but they all obey some common rules, in terms of the way the user advances through the unit. First and foremost, the user has to understand what he has to do, so the instructions on how to do the exercise have to be displayed and, in order to avoid misunderstanding, a translation has to be provided too. We will see how this was done in another chapter.

First, the user has to submit his answers that are then recorded as his genuine result for this session. Once this is done, his score is shown and all the wrong answers are clearly indicated (correct answers in green, wrong answers in red). He then has either the choice to move on to the next page, or to listen to the sounds to help him get the right answers. Only once he has listened to all the answers can he view the correct written answers.

These exercises are not meant to be listening comprehension exercises, so the tries submitted after the initial one for this session are corrected, but not recorded; also, on the next session, the sounds will be disabled again, so the user has to use his knowledge of previous sessions in order to find the answers, and not rely on sounds to better his score.

The user should also be able to view his previously recorded scores (this means scores from previous sessions) in order to see his progress. If the user wishes the right answers to be displayed, he should still be able to see where his wrong answers were (i.e. display in red).

The general pedagogical purpose of the exercise pages is twofold [MA 2002]:

- 1) Rehearsing what has been learned in the previous pages
- 2) Through different exercise types, gradually familiarize the user with the words' meanings and contextual use

Now let us see the different exercises

Exercise 1: Fill in blanks

In this exercise the user will simply have to fill in the blanks in a certain number of given sentences. There is not much to say about this exercise, as it is quite straightforward in terms of presentation. Each sentence is displayed on a separate line and a sound button is shown next to each sentence that will play the full sentence when clicked on (provided the sounds are enabled).

Exercise 2: Word association

In this exercise the user will have to match a certain number of words with other words from a list. The solution chosen to present this exercise was by using a drag and drop interface. The user will be shown a series of tags with words written on them, together with the words to match and a place to where they can drag and drop the corresponding tag.

Upon dragging any tag, the sound corresponding to the tag's word will be played, creating a passive listening experience. At any time, the user can move back a tag to the tag list and replace it by another tag. He can also directly replace a tag with another one.

Next to each word that must be matched, a sound button will be displayed, that plays the correct answer when pressed, but will only be enabled once the user has confirmed his answers.

Exercise 3: Sentence restatement

In this exercise, the user will get a certain number of pairs of sentences. The first of these sentences will have a part in italics and the other will have one or more blanks. The purpose is to restate the sentence by finding an alternate way to express the part in italics. It is very much the same as for exercise 1 except that there may be more than one blank per sentence.

Exercise 4 and 5: Fill in blanks with words from a list

These exercises are a mix between exercises 1 and 2. The user will have a list of word-tags and sentences with a blank. The tag with the word corresponding to the blank is to be dropped into the space provided. The tags work exactly in the same way as they do for exercise 2 and the sentences are presented exactly the same way as in exercise 1.

Exercise 6: Match sentences

In this exercise, the user will be provided with two series of the same number of sentences and they will have to match the sentences expressing the same statement. Each sentence of the first series is indicated by a number, while each sentence of the second series is indicated by a letter. The user will have to match the correct letter with each number by using a little pull-down menu. A sound button also has to be provided for each answer, just as for any other exercise.

4.2.6 Page 11: Idioms and proverbs

In this page, a certain number of idiomatic expressions are shown to the user. He can click on a button and a little animated man will say the expression out loud and explain it in English. Upon listening, an image illustrating the expression will also be displayed. In order to move on to the next page, the user has to listen to all the idioms.

The purpose of this page is obviously to help the user to remember some commonly used idiomatic expressions using vocabulary from the focus words and incorporate them in their vocabulary [MA 2002].

4.2.7 Page 12: Humor and true stories

In this page, the user can listen to a joke or a true story putting some of the focus words in context by clicking on a button. He then can display the text of the story to see if he has understood correctly. Only once he has listened to all stories will he be able to move on to the next page.

The humor and true stories page's pedagogical goal is to "*arouse the learner's awareness of the cultural elements underlying language learning. [...] each story or joke more or less reflects a facet of the western culture*" [MA 2002]. Again, a listening approach is adopted to encourage the habit of listening.

4.2.8 Page 13: Small dictation exercise

Here the user will be able to listen to a small dictation text and can try to rewrite it in the space provided. Once he has listened to the text, he can display the correct text and see for himself if he has understood correctly. Despite its name, this is not really an exercise; it will not be corrected, nor will the user get any score for it.

Upon reaching this page for the first time, regardless of whether he has listened to the text or not, the user's average will be calculated for the unit as this page is the last one. He will be granted a medal corresponding to the average of the best scores obtained on all submitted exercises.

The obvious pedagogical aim of this page is to provide the user with further practice in listening comprehension, but through the contents of the text itself, other purposes can be served, like introducing some western culture, or playing on the various meanings of words. Actually, this exercise is an open door for Qing Ma to make it serve any additional pedagogical purpose.

4.3 Users' freedom

The users' level of freedom is a key factor when making any CALL program. It will rate the way the program will guide the user and the way it will be perceived by the user. Also, according to the CALL efficiency model described in chapter 2.3.4, the users' actions have a direct impact on the efficiency of the CALL program; thus, controlling the level of freedom comes down to controlling the method's efficiency.

More often than not, CALL programs focus on maximizing freedom, enabling users to determine what pedagogical approach suits them best, but what we want to do is to specifically determine a particular pedagogical approach. Giving the user total freedom would therefore negate the very reason for the existence of the program. Also, in R. Goodfellow's terms, *"If learners are left on their own to study the CALL materials and have little knowledge of language learning in general, a CALL program based on a sophisticated self-learning strategy could prove more of a hindrance than a help. No matter how thoughtful the design is, as long as the control is in the hands of the learner, there is a possibility of a mismatch between human and machine that will lead to mutual incomprehension"*.

The cultural dimension also plays a major role in this freedom issue, as the Chinese have long been forced to obey rules, and have been culturally used to face an established order. Despite this, it is vital not to take away a certain degree of freedom, for it can be a good motivating factor. Ultimately, a suitable compromise between freedom and guidance has to be found.

The final decision was to leave the user free to choose which unit he wanted to work on, the units being totally independent, while having to work through those units in a linear manner. In each unit, though, he can only work on a page if he has already completed the previous page (i.e. he can not work on page 4 if he has not done pages 1, 2 and 3). Of course, the user should remain totally free to travel back and forth in the parts he has already completed.

This clearly depicts a bidirectional navigating system in which the user is guided through a single unit by a “previous page” – “next page” interface and still can choose what unit he wants to work on through a menu-type interface. Also, within one unit, the user should be provided with an easy way to jump to parts he has already completed and also to be able to know where he last stopped.

4.4 Marking system

The program will do exercises. The user receives a feedback of his performance thanks to a marking system. Setting up a good marking system is primordial. It has to motivate the user to achieve better results while avoiding frustration.

First, it is of course necessary to give the user feedback on how many right answers he got but he also has to have a more visual approach of his performance. Our first idea was to display a smiling man when a good score was achieved, a perplexed man in the case of lesser score and a crying man in the case of a mediocre score, all highlighted by a sound corresponding to the man’s facial expression. This solution was however abandoned as it turned out, according to research in the field, that this kind of feedback was irritating.

Eventually, a medals system was judged more appropriate and beneficial for the learning process. A good score would be granted a golden medal, a lesser score a silver medal, a mediocre score a bronze one and a bad score a spoon. The spoon can be interpreted as a metaphor of the need to eat in order to grow. Also, the user will receive a “unit medal”, upon finishing all parts of a single unit, which depicts his overall score on that unit.

4.5 User information

As we saw in the task analysis (chapter 3.1.2), the user will have to identify himself in order to create an account. This account will contain information that will be used by all the programs' tasks. It is thus of great importance to clearly determine what information is to be stored in that account and how all this information is to be arranged. As the information obtained would enable Qing Ma to create a "profile" of each user and evaluate the efficiency of her method by doing so, we worked analogically to the idea of a "profile" and put the information in a structure that clearly separates every relevant aspect of such a profile.

What characterizes a profile is of course the user's personal information, so this information will be at the root of our profile structure. In order to understand the structure of a user's profile, it is important to look closer at how a unit is designed. The program will be made up of several "units", each unit revolving around a particular theme and a specific vocabulary. Each unit contains several parts which we will refer to, analogically, as "pages", where pages can either be exercises or any other learning material. For each exercise page, the user can submit answers and thereby obtain a certain score; this giving of answers will be further called a "try" and can be done several times for a single page. Figure 4.1 summarizes the structure we have just described.

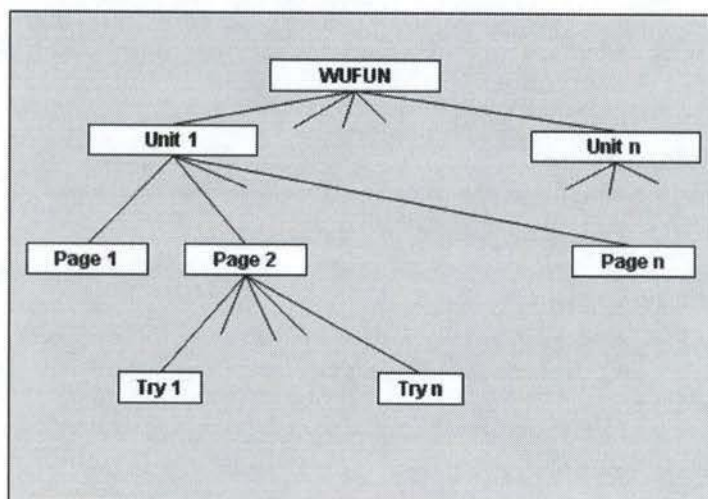


Fig 4-1 : structure of a profile

As we can see from this figure, page 1 and n have no tries attached to them; this is to stress the fact that, in terms of inner structure, exercise pages and non-exercise pages are no different except for the fact that exercise pages have tries attached to them.

The design of a profile should be based on this structure, as each node clearly distinguishes the relevant information we are seeking. To each node will be attached several types of information:

- 1) User's personal information: This information contains the description of the user and his login information.
- 2) Achievement information: This information describes the user's achievement, in terms of the scores obtained, at this level of the tree.
- 3) Statistical information: This information gives a feedback of the way the user used the program and its various features.
- 4) Historical information: This information gives feedback regarding the user's progress through the program¹.

Now let us explore what information is needed at each node. Note that only a little of this information will be viewable by the user, as most of it is there for Qing Ma to analyze in order to evaluate the efficiency of her method.

4.5.1 General information

The general information is found at the root of the user's profile; it includes information about the user's personal data and his use of the program.

User's personal information:

This information includes first name, last name, gender, age, studies (referred to as "specialty"), login and password. The login, password, first name and last name cannot be left blank; the others are of less importance and can thus be left blank.

Statistical information:

This information includes the number of times the user logged in and total connection time.

Historical Information:

The last unit the user worked on. This information is there in order to enable the user to resume the program with the same unit he was busy with the last time he left the program.

Information accessible to the user:

All this information is typically for research purposes, so none of it will be actually viewable by the user. Even though he knows what his personal information is, there is no need for him to view it after his profile has been created.

¹ We will see in the section about the user's freedom why this is so important

4.5.2 Unit information

The following information is stored in a data structure representing the user's achievement for each unit.

User's achievement information:

This information includes the average score on this unit's exercises and the overall medal for this unit².

Statistical information:

This information includes the number of times the user referred to focus words and memorization aids.

Historical information:

The only historical information is the last page of the unit the user worked on.

Information accessible to the user:

The overall medal is the only information that will be viewable by the user. We avoid showing the average score for two reasons: first because we have primarily chosen to make use of medals for the user to evaluate his overall performance, and secondly because having two evaluation systems can be confusing for the user and will eventually lead him to choose to evaluate his performance based only on one of the two.

4.5.3 Page Information

The following information is stored in a data structure representing the user's overall achievement for each page.

User's achievement information:

The average score on this page
The best medal earned for this page
The best score obtained on this page

Statistical information:

The total number of tries on this page

Information accessible to the user:

There is no information here directly viewable by the user; however, because he can see his results on the previous tries, he will nevertheless know the total number of tries on this page.

² How average score is calculated and the role of medals will be explained later. For now, it is sufficient to know that the medal is evaluated based on a fixed range of scores.

4.5.4 Try information

The following information is stored in a data structure representing the user's achievement for each try on an exercise.

User's achievement information:

The medal obtained on this try

The score obtained on this try

The answers given on this try

Statistical information:

The time spent on this try

The date when this try was submitted

Information accessible to the user:

With the exception of the answers he has given to his previous tries, the user will be able to see all the above information on each of his tries.

4.5.5 Final structure

Figure 4-2 illustrates the final data representation of a profile. Note that the choice is still left open as to how this data is to be stored (database, text file, etc.). This issue will be treated in the chapter on technical choices.

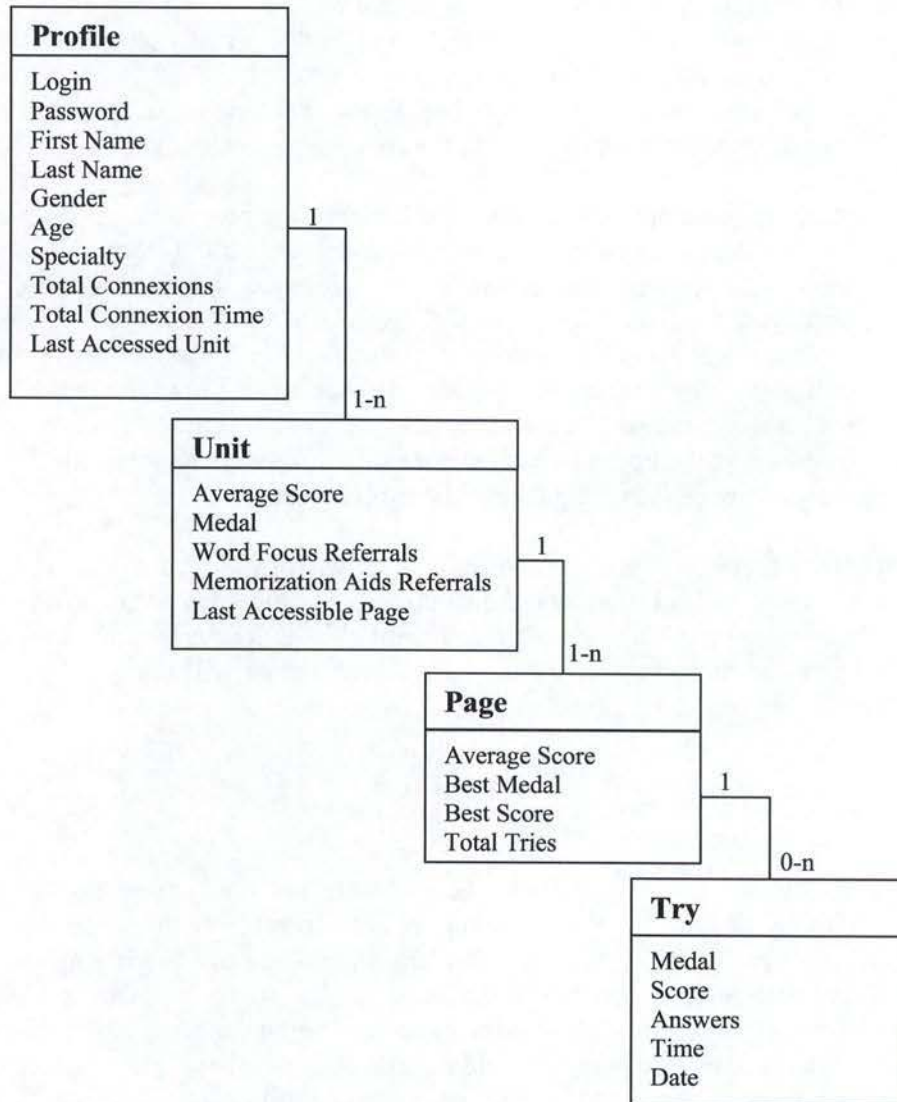


Fig 4-2 : A profile's data representation

4.6 Audio support

Audio support is one of the most important parts of the program. The base problem is that if a Chinese user has a way to avoid listening to sounds, he will [KEL et AL 1996]. As the method is principally focused on the use of sound, it is thus necessary to force the user to listen to sounds without inhibiting his progress. To achieve this goal, two objectives were defined:

- 1) Try as much as possible to play sounds automatically in a way that the user has no way of avoiding listening to sounds
- 2) In the cases where the objective above can not be achieved, provide incentives to the user to listen to the sounds.

In order to meet these objectives, several solutions were proposed:

- For memorization aids and focus words, the sound should be played whenever the user chooses a word for which he wants information to be displayed.
- For exercises where the user has to choose words from a list, the sound should be played whenever the user chooses a word from the list.
- For other exercises, the user should be able to carry on even without listening to the correct answer, but written answers should only be made available once the user has listened to all the sounds.

For the two first solutions, the use of pull down menus and “drag and drop” interfaces seems to lead to acceptable solutions. For the third point, it is vital that sound buttons are perfectly visible and identifiable as to the function they serve. It also has to be made perfectly clear to the user that he will not get the answers until he has listened to all the sounds.

4.7 Helping the user

As we said in the context analysis, the users are not really used to computers. Therefore they have to be provided with an easy to reach help on the use of the program. For pedagogical purposes, this help should be provided in English, but a translation will have to be made available to the users in case they do not understand. Because this translation is really optional and is not part of the learning process, the translation should not instantly be given; instead, the user is invited to take the initiative to consult the translation himself.

The parts of the program that a help/explanation system should be provided for are:

- The login and user creation
- The navigation system
- The instructions for the exercises

The parts of the program that will have to have a translation are:

- All the parts for which a help/explanation system is provided
- The focus words
- The word memorization aids

4.8 Analyzing the results

As I explained, the students' results will have to be analyzed by Qing Ma in order to evaluate the efficiency of her method. Therefore, a way must be provided for her to separately analyze the results of each student, having total access to all the information described in the previous section (except for the historical information which is only useful for the program itself).

It is important to note that this analysis tool has to remain independent of the program, because it will eventually not remain a research tool, but instead a fully operational self-learning program to be used on an individual personal computer. At that point, the results' analysis will no longer be part of the program, so it is best to keep them separate right from the start.

5 Functional Analysis

5.1 Introduction

In the specifications, we expressed several important points for the functional analysis:

- A user is defined and identified by a profile
- The user's use of the program is strongly defined by navigational restrictions
- The navigation will rely on a bidirectional axis: unit navigation and page navigation
- The user can evaluate his progress by submitting tries to exercises
- Qing Ma will evaluate the efficiency of her method by analysing detailed feedback about each user's progress and results on exercises throughout the program
- The use of sounds is a preponderant key factor in Qing Ma's method's efficiency


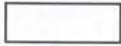


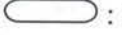
We can thus identify four recurrent keywords in the specifications: "profile", "navigation", "exercises" and "sound"

We can easily divide the program in four main parts:

- 1) Profile management
- 2) Navigation management
- 3) Exercise management
- 4) Sound management

5.2 Tasks

In order to identify all the functions I will need, I will first describe each task with a diagram of interaction between the user and the system. The notation used in these diagrams is to be interpreted in the following way:

-  : User's input
-  : Function
-  : Decision
-  : Alternative
-  : Link to another task

5.2.1 User Identification

To access the program, the user has to identify himself through a login and legitimate this identification by supplying a password. If the information is verified as correct, the profile corresponding to the given login is loaded, and if the loading occurs correctly, the user can then start navigating the program. In order to create a new profile, the user can also go to the user creation instead of logging in.

The task is described in figure 5-1.

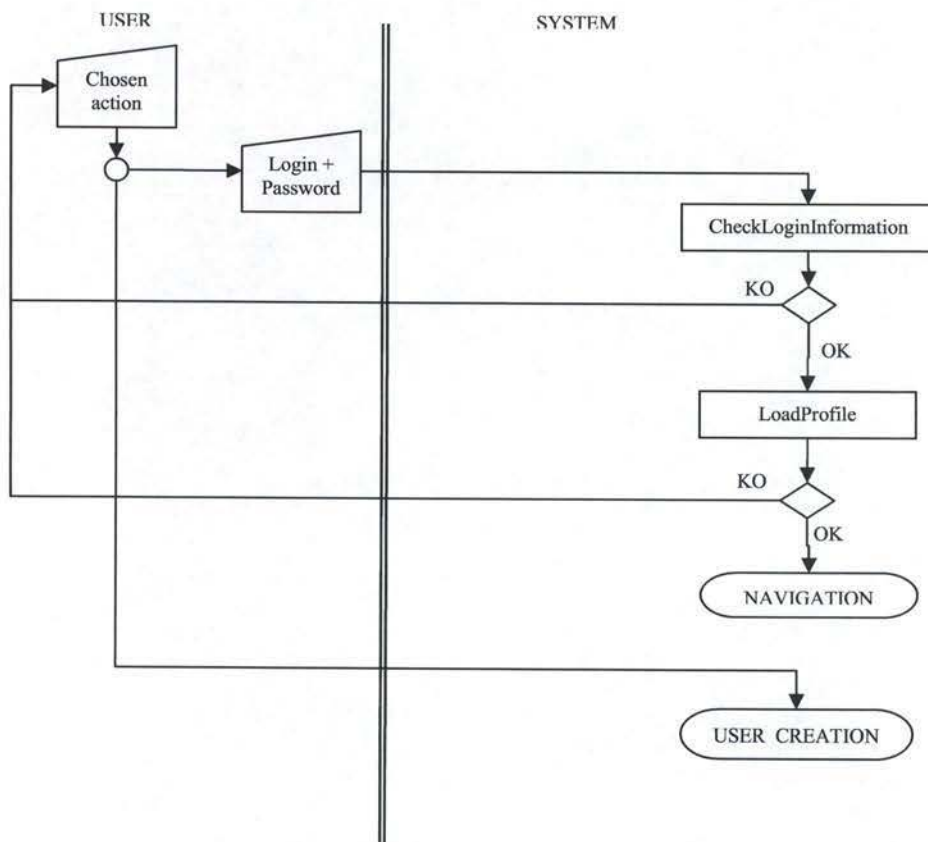


Fig 5-1 : The USER_IDENTIFICATION task

5.2.2 User creation

In order to be able to enter the learning program, the user has to create a profile. He first has to enter his personal information; if some information is missing, incorrect or if a profile already exists with the same login given by the user in his personal information, the user will have to re-enter the information correctly. If everything goes fine, the profile is created, and if no error occurs, the user is brought to the navigation. The user can also go back to the user Identification.

The user creation task is represented in figure 5-2.

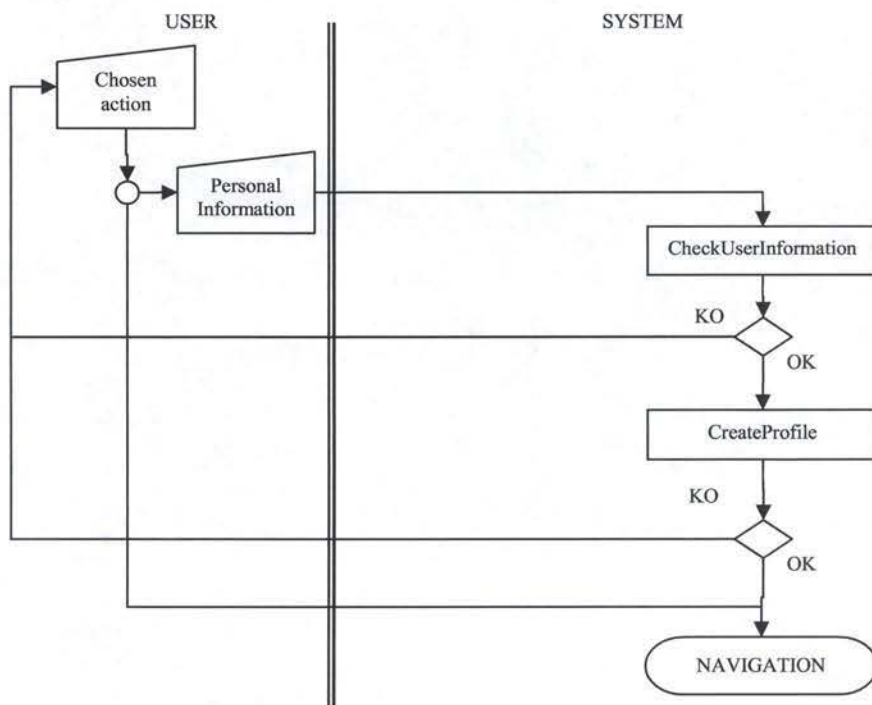


Fig 5-2 : The USER_CREATION task

5.2.3 Navigation

As we said before, the user can navigate through units and through pages. At this point the user can perform one of three tasks:

- 1) Change the active unit
- 2) Go to one of the unit's pages
- 3) Quit the learning program

When entering the navigation, we want a unit to be activated as default. The default unit should be the last unit the user worked on when he last quit the program, or the first unit if it is the first time he uses the program.

The navigation task is described in figure 5-3:

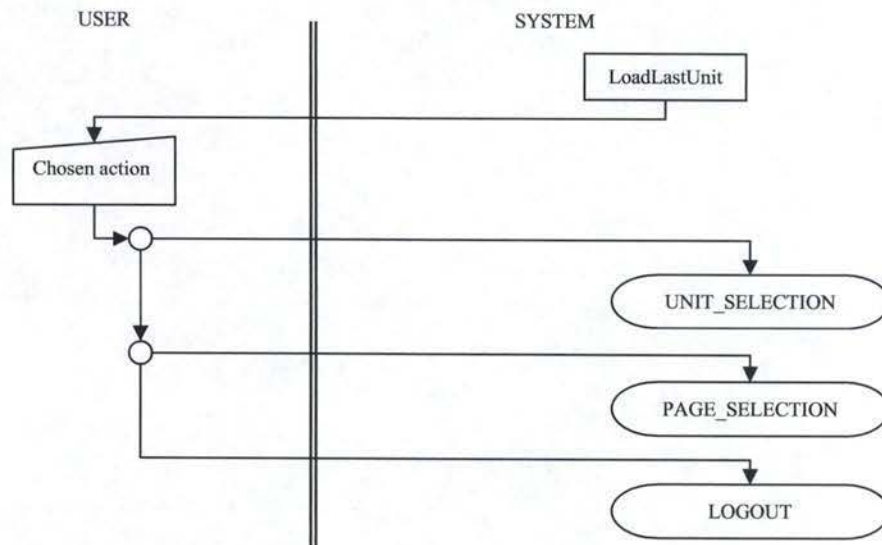


Fig 5-3 : The NAVIGATION task

5.2.4 Unit Selection

In the unit navigation task, the user chooses a unit, which is then set as the current unit and then the profile is updated so that the last accessed unit is the chosen unit. When this is done the user goes back to the navigation.

Figure 5-4 illustrates this.

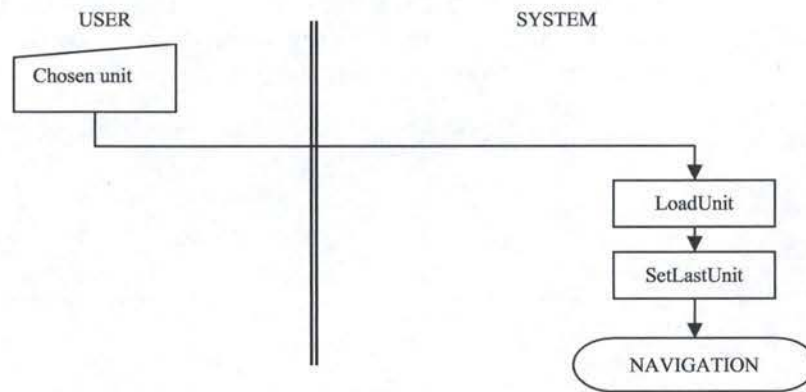


Fig 5-4 : The UNIT_SELECTION task

5.2.5 Page Selection

Here the user will choose a page from the pages he has access to in the active unit. Once the user has selected a page, the page is displayed and he can then interact with it. Figure 5-5 describes the PAGE_SELECTION task

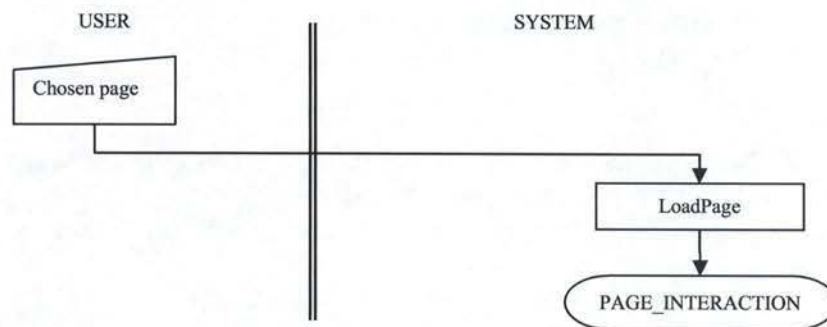


Fig 5-5 : The PAGE_SELECTION task

5.2.6 Page Interaction

When I talk about page interaction, I mean all the actions a user can perform when he is on a given page. First there are some generic actions available on every single page:

- Go to next/previous page
- View the instructions
- View the word focus
- View the word memorization aids
- Go to navigation
- Logout

Not all these actions are always enabled. For instance, quick viewing the focus words or the word memorization aids can only be done if the user has access to their respective pages. Going to the next page is only possible if the next page is accessible to the user.

Besides this, we have various interactions with the page that greatly vary and are defined by the page actually interacted with. We will thus identify a different task for each different type of page:

- Over viewing the story
- Doing an exercise
- Viewing Idioms
- Listening to humor and true story
- Doing the dictation exercise

These tasks will be described later, for now we just describe the Page Interaction as a whole in figure 5-6:

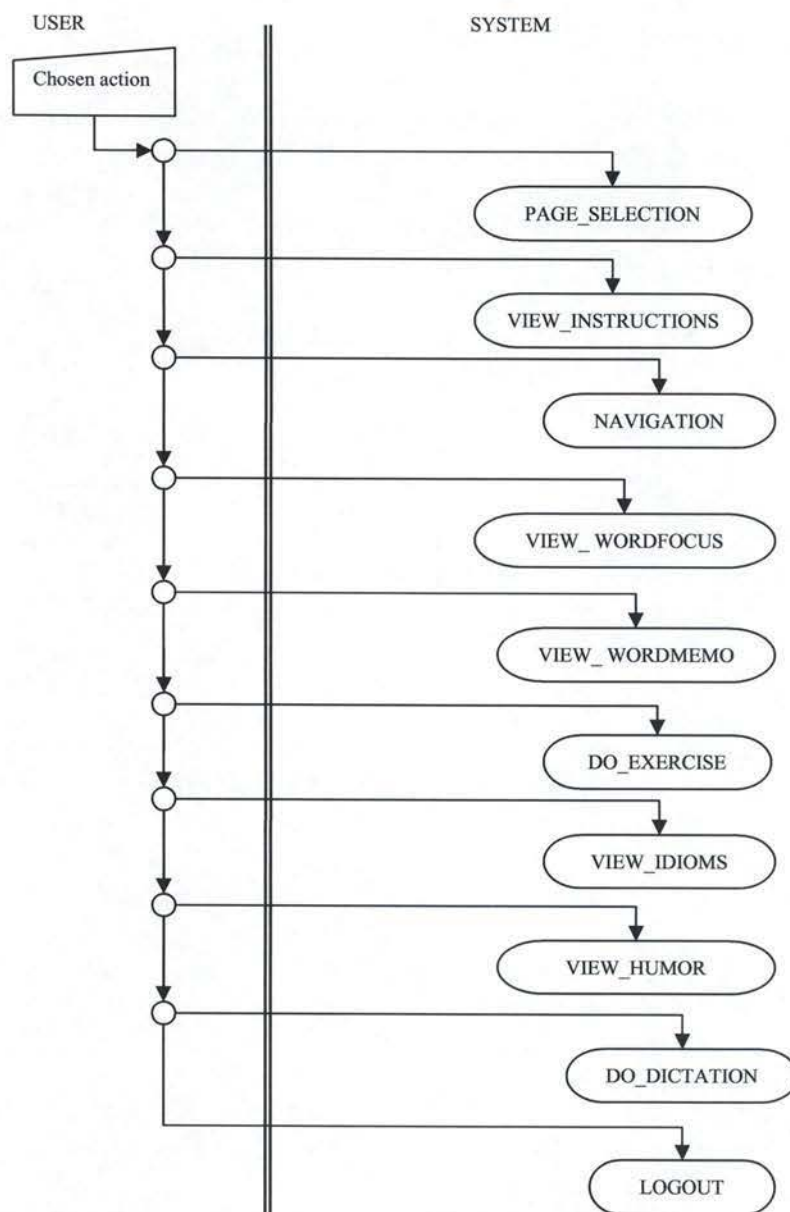


Fig 5-6 : The PAGE_INTERACTION task

5.2.7 Viewing Instructions

On any page, the user can view instructions on what he has to do in order to gain access to the next page. As we said before, a way also has to be provided to translate those instructions in Chinese. Once the user has viewed the instructions, he can go back to the page interaction. Figure 5-7 describes the VIEW_INSTRUCTIONS task.

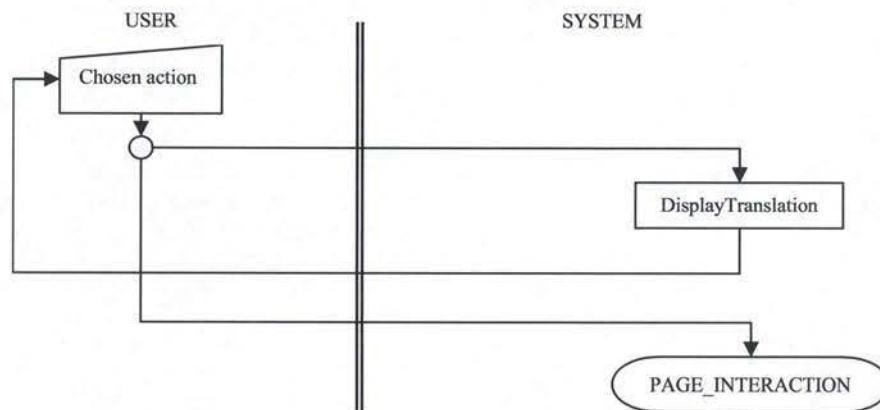


Fig 5-7 : The VIEW_INSTRUCTIONS task

5.2.8 Viewing Word Focus

In order to view the word focus, the user has to choose a word he wants to display. We want the sound corresponding to the chosen word to be played as the user selects the word to be displayed. The User can also ask for a translation of the selected word. Upon viewing a word, the last accessible page for this unit is updated. Once he is finished, the user goes back to the page interaction. This is described in figure 5-8.

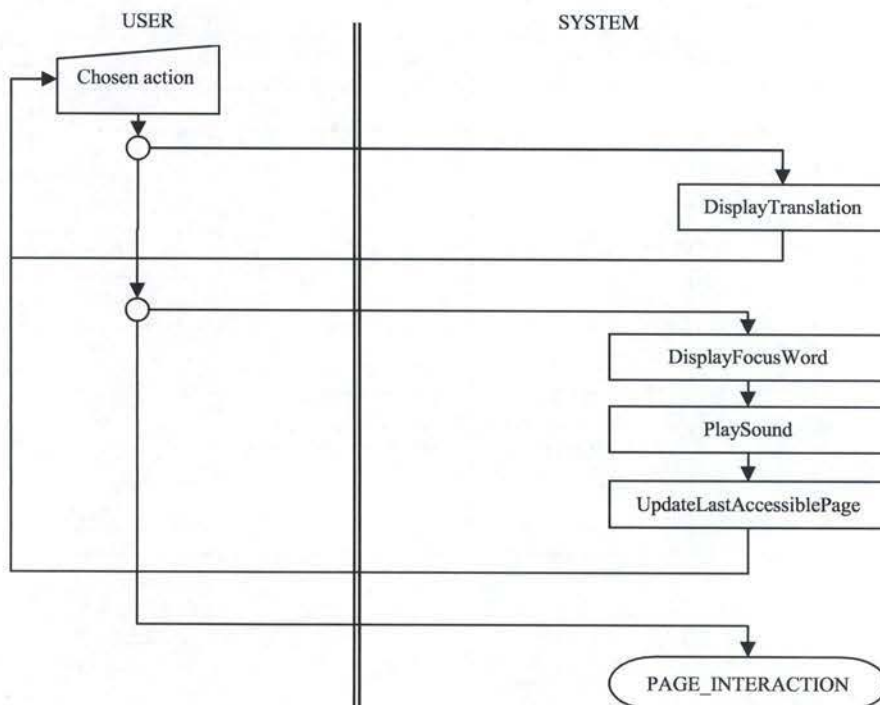


Fig 5-8 : The VIEW_WORDFOCUS task

5.2.9 Viewing Word Memorization Aids

Viewing word memorization aids works in the same way as viewing the word focus, except that some aids contain additional audio material that can be listened to by the user. VIEW_WORDMEMO is described in figure 5-9.

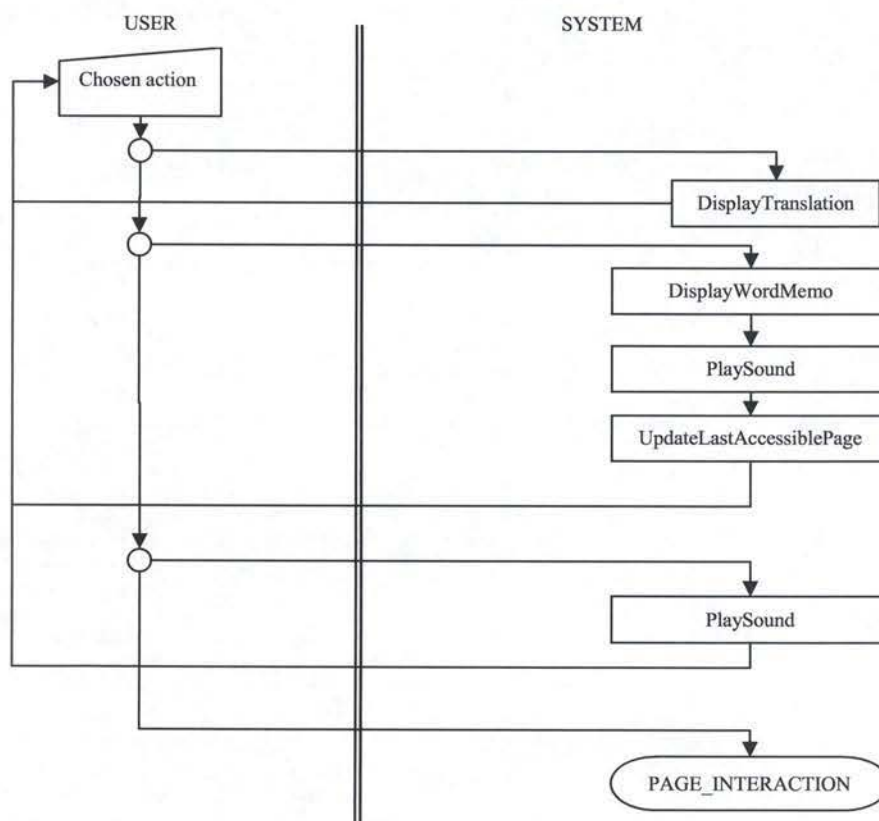


Fig 5-9 : The VIEW_WORDMEMO task

5.2.10 Doing an Exercise

All the exercises work the same way: the user gives his answers to the exercise in the way provided by the interface; the system then displays the score, adds a new try to the user's profile, colors the right answers green and the wrong ones red, and updates the last accessible page for this unit. Once this is done, the user can do the following things:

- 1) Listen to sounds
- 2) Give his answers again, though this will not record a new try in his profile
- 3) View the correct answers if he has listened to all the sounds before
- 4) View all his recorded tries on this exercise

Figure 5-10 describes the task.

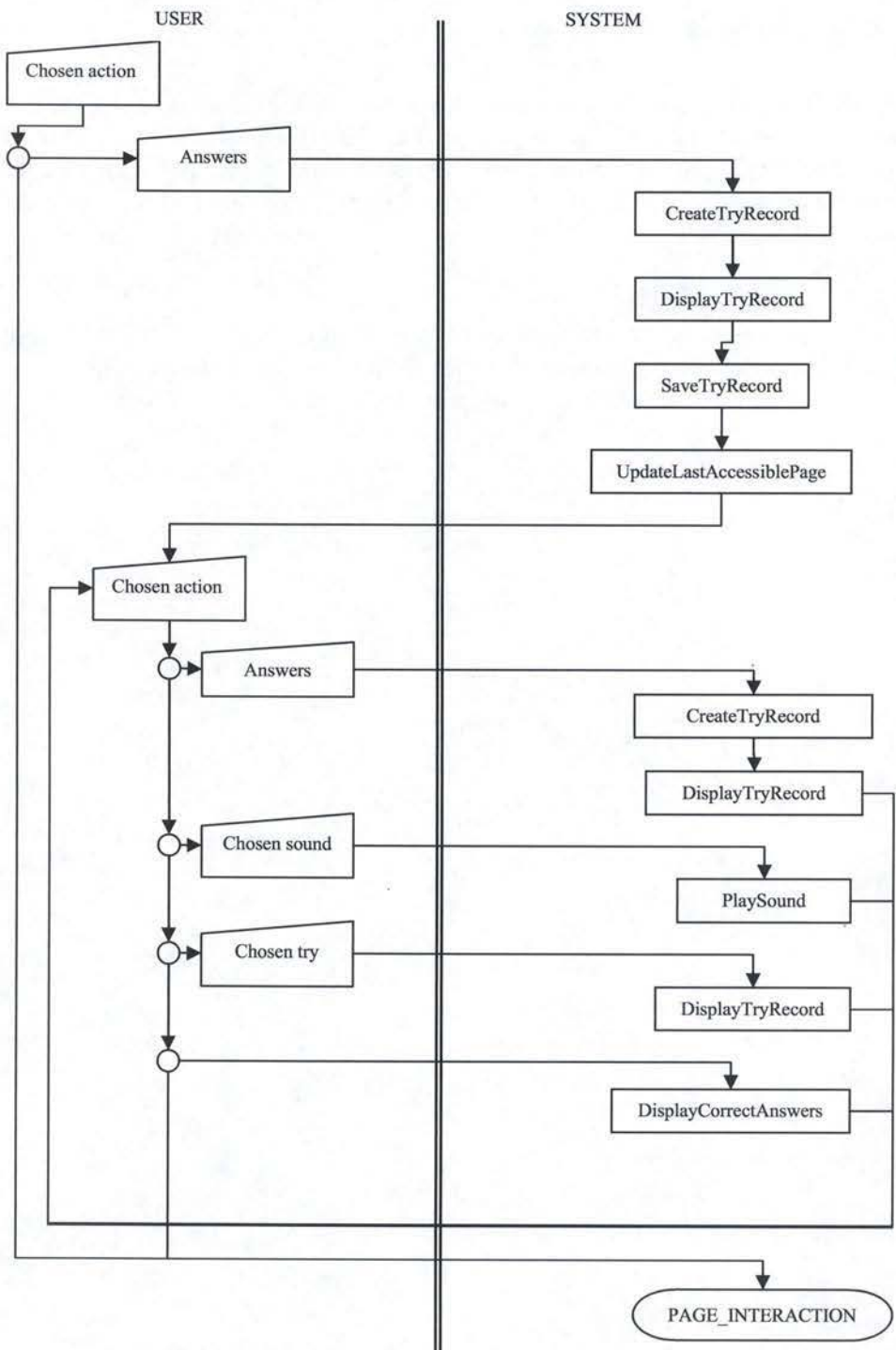


Fig 5-10 : The DO_EXERCISE task

5.2.11 Viewing the Idioms

On the idioms page, the user has to view each idiom in order to pass to the next page. Upon viewing each idiom, an animated man explains the meaning of the idiomatic expression. Once all idioms are viewed, the last accessible page is updated. This is described in figure 5-11.

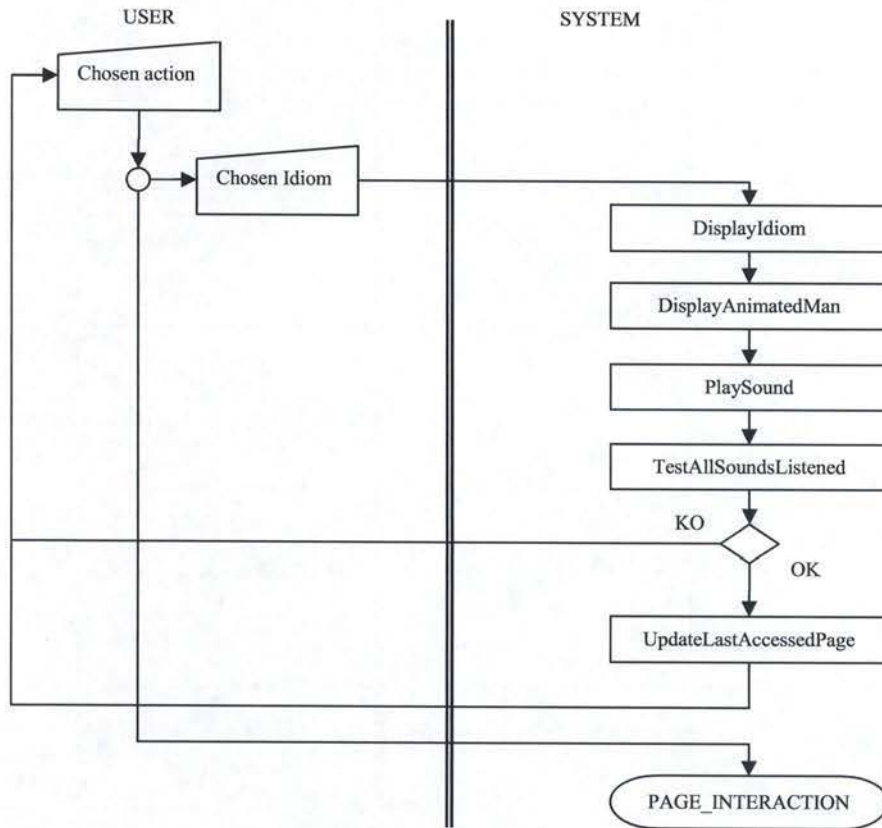


Fig 5-11 : The VIEW_IDIOMS task

5.2.12 Viewing the Humor and True Stories

On the “Humor and True Stories” page, the user can listen to several stories. Once a story is listened to, he will be able to view the text. When all stories are listened to, the last accessible page is updated. This is described in figure 5-12.

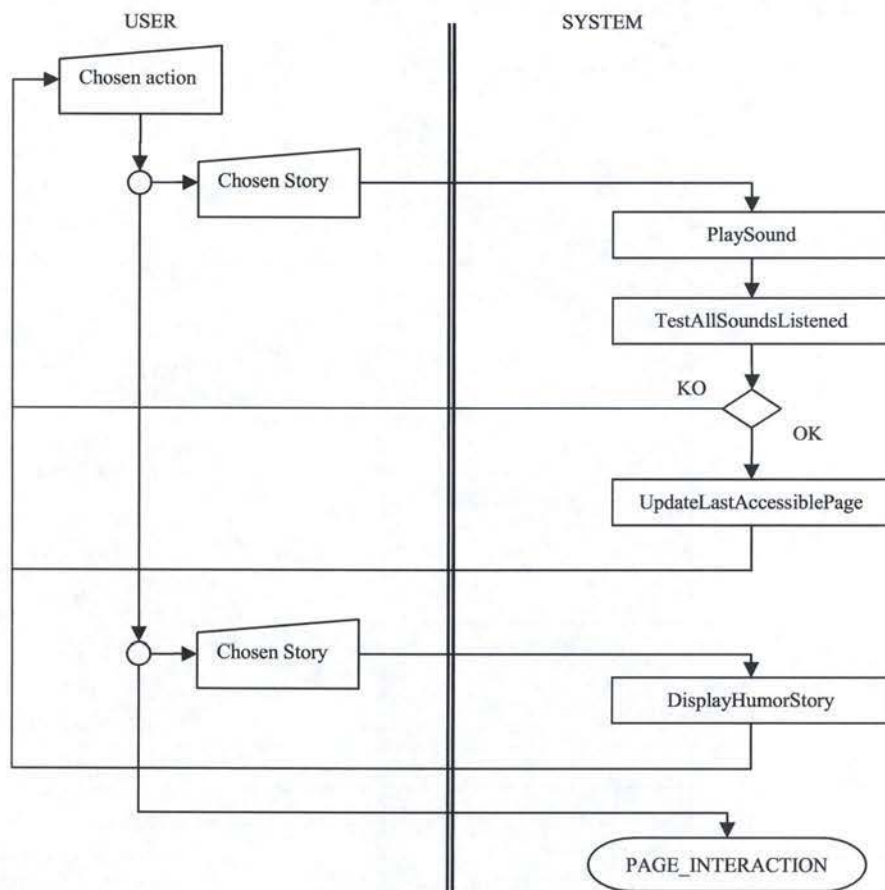


Fig 5-12 : The VIEW_HUMOR task

5.2.13 Doing the Dictation Exercise

On the “Dictation Exercise” page, the user can listen to the dictation, try to write down the text and view the correct text. This is described in figure 5-13

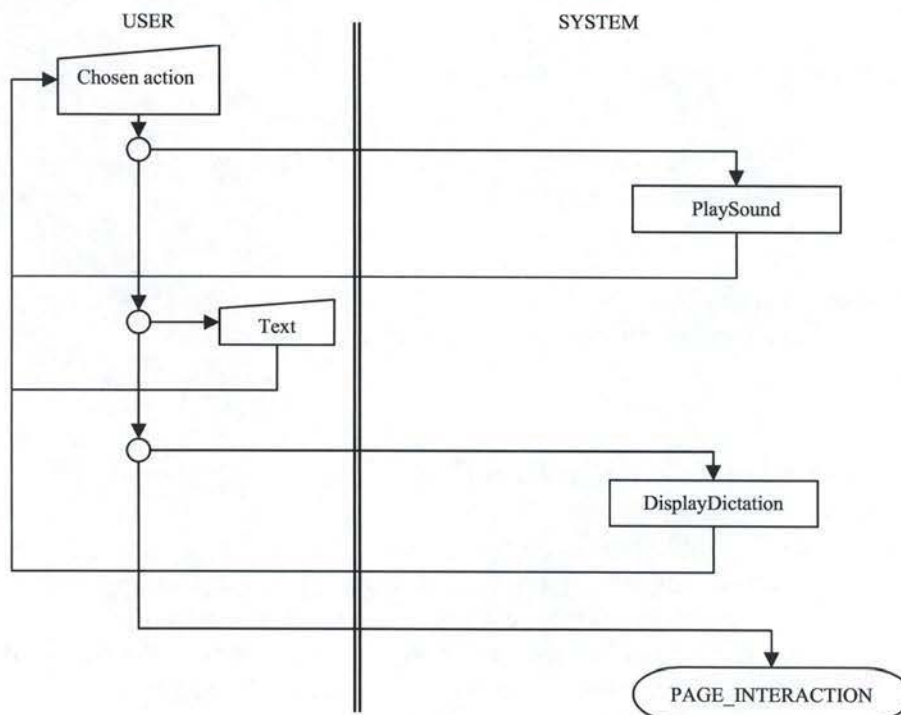


Fig 5-13 : The DO_DICTATION task

5.3 Profile Management

The functions in profile management are all the functions that alter information in a profile, retrieve information from it, save or load a profile to or from the hard disk, etc.

5.3.1 CheckLoginInformation

CheckLoginInformation checks if the login supplied by the user exists, and then checks if the password supplied by the user matches the password associated to that login.

The information given by the user is referred to as LoginInformation and contains the following information:

Login

Password

Boolean CheckLoginInformation(LoginInformation)

Pre: LoginInformation is initialized

Post: one of the following scenarios occurs

- 1) If the login given by the user does not exist, he is prompted that no login corresponds to the one supplied and “false” is returned
- 2) Else if the login exists, but the given password does not match, the user is prompted that the password he supplied is incorrect, and “false” is returned
- 3) Else “true” is returned

5.3.2 LoadProfile

LoadProfile retrieves the profile corresponding to a given login from the hard disk and loads it in the volatile memory. We refer to the “volatile” version of the profile as the “active” profile.

Boolean LoadProfile(Login)

Pre: Login is initialized and refers to an existing profile

Post: If an unexpected error occurs when loading the profile, the user is prompted that his profile could not be loaded and “false” is returned

Else “true” is returned and the profile referred to by “Login” is the active profile

5.3.3 CheckUserInformation

When creating a user, we will have to be attentive to the following things:

- 1) The first name, last name, login and password cannot be left blank
- 2) The login cannot be an already existing one
- 3) The user has to confirm his password in order to make sure he did not misspell it

The information given by the user is referred to as UserInformation, and contains the following information:

First Name (mandatory)

Last Name (mandatory)

Age

Gender

Specialty

Login (mandatory)

Password (mandatory)

Confirmed Password (mandatory)

Boolean CheckUserInformation(UserInformation)

Pre: UserInformation is initialized

Post:

- 1) If one or more of the mandatory information is blank, the user is prompted that the information is missing and "false" is returned.
- 2) Else if the login already exists, the user is prompted that he has to choose another login and "false" is returned.
- 3) Else if password confirmation doesn't match the initial password, the user is prompted that he has to re-enter the password and confirm it again and "false" is returned
- 4) Else true is returned

5.3.4 CreateProfile

With a given set of personal information, CreateProfile initializes a new profile that is then set as the active profile and saved on disk.

Boolean CreateProfile(UserInformation)

Pre: UserInformation is initialized and has been validated by CheckUserInformation

Post: If an unexpected error occurs, the user is prompted that his profile could not be created and “false” is returned

Else, a profile is created and initialized with the information hereunder³; saved on disk, and true is returned.

The profile is initialized as follows as specified in figure 4-2 (see chapter 4.2)

General information:

- Login: Value given in UserInformation
- Password: Value given in UserInformation
- First Name: Value given in UserInformation
- Last Name: Value given in UserInformation
- Gender: Value given in UserInformation
- Age: Value given in UserInformation
- Specialty: Value given in UserInformation
- Total Connections: 0
- Total Connection Time: 0
- Last Accessed Unit: 1
- Units: Several initialized units⁴

Each unit:

- Average Score: 0
- Medal: Spoon
- Word Focus Referrals: 0
- Memorization Aids Referrals: 0
- Last Accessible Page: 1
- Pages: One initialized page per page for this unit

³ The final format of the information is not yet defined; the given values are abstract and their final type will be clarified in the “technical choices” chapter.

⁴ The number of units associated to a profile is the same as the number of the units implemented for the program, which is only one for the testing unit.

Each page:

- Average Score: 0
- Best Medal: spoon
- Best Score: 0
- Total Tries: 0
- Tries: null

5.3.5 SetLastUnit

SetLastUnit sets the active profile's value for the Last Accessed Unit as the UnitNumber referring to the active unit

Void SetLastUnit(UnitNumber)

Pre: The active profile is initialized, the active unit is initialized

Post: The active profile's Last Accessed Unit is the active unit and is saved on disk

5.3.6 UpdateLastAccessiblePage

If the current page's PageNumber corresponds to the Last Accessible Page of the active unit, UpdateLastAccessiblePage sets the active profile's value for the Last Accessible Page of the active unit as the PageNumber following the active page's PageNumber hierarchically. The active profile is then saved on disk

Pre: The active profile is initialized, the active unit is initialized and the active page is initialized

Post: The Last Accessible Page, in the Unit corresponding to the active unit, in the active Profile is greater than, or equal to, the PageNumber following the active page's PageNumber hierarchically, and the active profile is saved on disk

5.3.7 SaveTry

SaveTry adds a given Try to the Page corresponding to the active page, in the unit corresponding to the active unit, in the active profile and then saves the profile to disk

Void SaveTry(TryRecord)

Pre: TryRecord is an initialized Try; the active profile is initialized and corresponds to an existing profile; the active unit is initialized and is associated to an existing unit in the active profile; the active page is initialized, is an exercise page, and is associated to an existing page in the active unit

Post: The given TryRecord is added to the tries of the active page of the active unit of the active profile and the active profile is saved to the disk

5.3.8 GetInformation

Although this function is found nowhere in the tasks we described, it is an important function for profile management as it is used inside its main functions. GetInformation retrieves specific information from the active profile. This is actually a generic function; there are as many variations of it as there are types of information in a profile.

Here is a relevant but not exhaustive list of those variations.

Password GetPassword()

Pre: The active profile is initialized

Post: The password of the active profile's user is returned

Medal GetUnitMedal(UnitNumber)

Pre: The active profile is initialized; UnitNumber refers to an existing unit

Post: The medal of the given unit is returned

TotalTries GetPageTotalTries(UnitNumber, PageNumber)

Pre: A profile is active and initialized; UnitNumber refers to an existing unit, PageNumber refers to an existing page in UnitNumber

Post: The Total Tries on the given page of the given unit is returned

Score GetTryScore(UnitNumber, PageNumber, TryNumber)

Pre: A profile is active and initialized; UnitNumber refers to an existing unit, PageNumber refers to an existing page in UnitNumber, TryNumber refers to an existing try in PageNumber

Post: The Score obtained on the given try of the given page of the given unit is returned

5.3.9 SetInformation

SetInformation is the opposite of the getInformation function. It changes values in a profile and is also an important function. It works just analogically to GetInformation and is not worth detailing any further.

5.4 Navigation Management

Navigation management deals with all the functions that alter the users' working environment, whether it is by changing the active unit and page, or by giving access to functions or tasks that were disabled. Some functions in navigation management are closely interacting with profile management, in order to correctly set up the users' working environment.

5.4.1 LoadLastUnit

When the user goes to the navigation, the pages accessible from there should be the pages of the unit he last accessed. The set of accessible pages is defined by what we will further call the “active unit”.

Void LoadLastUnit()

Pre: The active profile is initialized

Post: The active Unit is the unit last accessed by the user whose profile is the active profile

5.4.2 LoadUnit

LoadUnit sets the unit corresponding to the given unit number as the active unit.

Void LoadUnit(UnitNumber)

Pre: UnitNumber is initialized and refers to an existing unit

Post: The active unit is the unit referred to by the given UnitNumber

5.4.3 LoadPage

LoadPage sets the page corresponding to the given page number as the active page.

Void LoadPage(PageNumber)

Pre: PageNumber is initialized and refers to an existing page

Post: The active page is the page referred to by the given PageNumber

5.4.4 DisplayTranslation

DisplayTranslation displays the translation associated to a text or word referred to by a TranslationName

Void DisplayTranslation(TranslationName)

Pre: TranslationName is initialized and refers to an existing translation

Post: The translation associated by TranslationName is displayed

5.4.5 DisplayFocusWord

DisplayFocusWord displays the description of a given focus word, referred to by a FocusWordName

Void DisplayFocusWord(FocusWordName)

Pre: FocusWordName is initialized and refers to an existing focus word

Post: The explanation of the focus word referred to by the given FocusWordName is displayed.

5.4.6 DisplayWordMemo

DisplayWordMemo displays the description of a given word memorization aid, referred to by a WordMemoName

Void DisplayWordMemo(WordMemoName)

Pre: WordMemoName is initialized and refers to an existing word memorization aid

Post: The explanation of the word memorization aid referred to by the given WordMemoName is displayed.

5.4.7 DisplayTryRecord

DisplayTryRecord displays a given Try's information

Void DisplayTryRecord(TryRecord)

Pre: TryRecord is an initialized Try

Post: The information in Try is displayed

5.4.8 DisplayCorrectAnswers

DisplayCorrectAnswers displays the correct answers of the active page

Void DisplayCorrectAnswers()

Pre: The active page is initialized and is an exercise page

Post: The correct answers of the active page are displayed

5.4.9 DisplayIdiom

DisplayIdiom displays the Idiom referred to by a given IdiomName and sets that idioms state to “viewed”.

Void DisplayIdiom(IdiomName)

Pre: IdiomName is initialized and refers to an existing idiom

Post: The idiom referred to by the given IdiomName is displayed and its state is “viewed”

5.4.10 DisplayAnimatedMan

DisplayAnimatedMan displays the animated man referred to by the given AnimatedManName

Void DisplayAnimatedMan(AnimatedManName)

Pre: AnimatedManName is initialized and refers to an existing animated man

Post: The animated man referred to by the given AnimatedManName is displayed

5.4.11 DisplayHumorStory

DisplayHumorStory displays the the humor or true story referred to by the given HumorStoryName

Void DisplayHumorStory(HumorStoryName)

Pre: HumorStoryName is initialized and refers to an existing humor or true story

Post: The humor or true story referred to by the given HumorStoryName is displayed

5.4.12 DisplayDictation

DisplayDictation displays the dictation referred to by the given DictationName

Void DisplayDictation(DictationName)

Pre: DictationName is initialized and refers to an existing dictation

Post: The dictation referred to by the given DictationName is displayed

5.5 Sound Management

Sound Management is apart from navigation management to draw a line between the visual environment and the sonic environment. Also the sound manager will keep track of the sounds listened on each page in order to determine which sounds have been listened to, and which haven't. We will also talk about an "active sound" just like we talked about an "active profile" or an "active unit". For the users' sake, it is better to keep it that only one sound can be active at a time. A sound is active until another sound is activated, or until the sound is completely played.

5.5.1 PlaySound

PlaySound plays a given sound referred to by a SoundName

Void PlaySound(SoundName)

Pre: SoundName is initialized and refers to an existing sound

Post: The sound referred to by the given SoundName is played and is the active sound

5.5.2 TestAllSoundsListened

Tests if all the sounds on the active page are listened

Boolean TestAllSoundsListened()

Pre: The active page is initialized

Post: If all the sounds on the active page are listened, true is returned

Else false is returned

5.6 Exercise Management

5.6.1 CreateTryRecord

When a user has finished an exercise a Try is created and initialized according to the comparison between the answers given by the user and the correct answers, respectively referred to by GivenAnswers for the former, and CorrectAnswers for the latter.

Void CreateTryRecord(GivenAnswers, CorrectAnswers)

Pre: GivenAnswers and CorrectAnswers are initialized and are both sets of the same number of elements representing answers

Post: A new Try was created and initialized as follows

- 1) Score is a ratio between the number of elements in GivenAnswers that match the corresponding element in CorrectAnswers, and the total number of answers.
- 2) Medal is a spoon if the ratio is less than 50%, a bronze medal if the ratio is between 50% and 70%, a silver medal if the ratio is between 70% and 90%, a gold medal if the ration is over 90%
- 3) Time is the time elapsed between the time when the exercise page was last accessed and the time when the answers were given.
- 4) Date is the date when the answers were given
- 5) Answers corresponds to GivenAnswers

6 Technical choices and implementation issues

6.1 Choice of the Programming Environment

6.1.1 Object Oriented

As it was known that there would be a graphical user interface for the program, the options for the language to use for the program were quickly narrowed down to one single programming paradigm: Object Oriented. Another possible solution was to implement it as a website; making a flash interface was indeed an attractive option, but the structure of the profile and my wider experience in this field made me definitely set my choice on Object Oriented Programming.

6.1.2 Choice of the Language

Choosing amongst the various object oriented environments turned out to be quite difficult. Two of them seemed very diverse and interesting choices: Java and C++.

Table 6-1 summarizes the main differences between Java and C++

	Java	C++
Portability	Totally portable	Not always portable
Memory Management	Garbage Collector	Controlled by the programmer
Inheritance	Single inheritance	Multiple inheritance
Packages Standardization	Totally standardized	Barely Standardized
Documentation	Very accessible, clear and thorough	Widespread, often unclear and incomplete
Controllability	Less controllable	Very controllable
Speed of Treatments	Relatively slow	Relatively fast
My Experience	Moderate	Close to none

Table 6-1 :Java Vs C++

Eventually, my choice was to use Java, mainly because I had more experience with Java, and also because of its standardization, portability, and friendly documentation. Nevertheless, C++ was definitely a good choice too.

6.1.3 Operating system

Programming in Java implies a total portability on any operating system, so I could have programmed on any Operating System. Eventually I ended up working with various platforms; having Windows 98 on my portable computer and Windows 2000 on the office's computer or even Linux.

6.1.4 NetBeans IDE

With the amplitude of the interface, programming with a notepad was of course out of the question. I definitely had to go for an Interface Development Environment (IDE). After asking some friends, I heard about the NetBeans IDE, which is an open source, very well designed project entirely programmed in Java. I chose to implement entirely the program with this IDE, for it offers very attractive features, like automatic suggestions of functions, built-in javadoc, instant syntax check etc.

6.2 Data Structure

Now that the programming is set, the structure of the data can be clarified

6.2.1 The Profile

As we saw in chapter 4, a profile contains several units, which contain several pages, themselves containing several tries. Each of those elements can be easily identified as individual types of objects, so one class should be made for each such type. Also, because the number of units can be easily determined by the programmer and the number of pages in a unit too, I decided to put the profile's units in an array of units and each unit's pages in an array of pages. The same is not true for each page's tries, because the numbers of tries vary greatly, and should rather be put in a more dynamic structure. A Vector seemed to be the perfect solution. The complete typing of a profile is as follows:

Profile:

- User Information(firs name, last name, gender, age, specialty, login and password): String[]
- Total Connections: int
- Total Connection Time: int
- Last Accessed Unit: int
- Units: Unit[]⁵

⁵ The number of units associated with a profile is the same as the number of the units implemented for the program, which is only one for the testing unit.

Unit:

- Average Score: float (in %)
- Medal: int (0 for spoon, 1 for bronze, 2 for silver, 3 for gold)
- Word Focus Referrals: int
- Memorization Aids Referrals: int
- Last Accessible Page: int
- Pages: Page[]

Page

- Average Score: float (in %)
- Best Medal: int
- Best Score: float (in %)
- Total Tries: int
- Tries: Vector (containing Try objects)

Try

- Date: String
- Answers: String[]
- Time: String
- Score: String⁶
- Medal: int

6.2.2 The Units

The issue of knowing how a unit could be represented in a generic way (not the Unit in the Profile, but the actual unit's contents) was really a problem. Only one unit was fully specified so far, and the exercises could vary very much from one unit to another. I eventually decide to make one graphical class for each page of the unit. A unit would thus be a set of graphical classes that are each different and implement a separate page.

6.2.3 The sounds

Wufun makes a considerable use of sounds, so determining how to handle this sound was an important decision. There were mainly two ways of dealing with sound:

- Recording the sounds
- Using Text-To-Speech technology

⁶ I chose to represent a score by a String instead of by an array of two int [score, maxscore] to make its display easier. By parsing the score all the needed information can be extracted.

Although Text-To-Speech was very attractive, for it would reduce considerably the space occupied by the program on the hard disk, it has many issues. I tested several TTS synthesizers, and performed a few tests. In order to be suitable for the projected use, the synthesizer would have to satisfy all of these tests. The evaluation points were the following:

- Sound quality
- Flow of speech
- Intonation quality
- Receptivity to heterophonic homographs⁷

The three first points were arbitrary; the last one was tested through the use of three sentences:

“Have you read that book?”

“I read that book yesterday!”

“I couldn’t read mine yesterday, my friend borrowed it.”

I tested any synthesizer I could lay my hands on over the internet, but none were found to be even close to the expected results. For the great majority of synthesizers, the sheer quality of sound was so bad I did not even bother testing anything else. Some programs were worth paying attention to in terms of the flow of speech, but were quickly eliminated on the intonation quality. Actually, only one synthesizer really rendered a good quality of speech: AT&T’s but did not pronounce the two first sentences correctly [ATT]. Some Synthesizers, like the MBrola Synthesizer [DUT] were actually able to pronounce all three sentences correctly, but with a quality too poor to use in the program.

The solution to record sounds was thus eventually chosen. The question to know what kind of compression to use was quickly solved, as Java does not propose a built-in class to manage mp3 or wma. The sounds were thus recorded in 8 bits, mono, wave sounds.

6.2.4 Storing the data

The choice of the way user data was to be stored was left open. Using a database was a viable choice, but other solutions like a text file or XML were also possible ones. Another interesting solution was to serialize the state of the profile on disk (i.e. dump the state of the instantiated Profile into a file).

⁷ Words written in the same way, but pronounced differently

First what had to be taken in consideration was that the program is supposed to be used as a self learning method to use at home, so there would not be a lot of profiles stored on a single computer. Most likely, it would not exceed a dozen profiles. Also for her experiments, Qing Ma will have to be able to gather the profiles present on several computers and centralize them in order to analyze them more easily. All this combined clearly made me decide that the database solution was not suitable. Any of the three other choices (XML, text file and serialization) seemed equivalent, but in terms of implementation, the serialization was by far the easiest way to go. In order to find the right file from a given profile easily, I decided to name the saved profiles after their users' login, and added ".wuf" as file extension. E.g.: If someone creates a profile with "bob" as login, his profile would be stored in a file named "bob.wuf".

6.3 Problems Encountered

In this section, I will go over the most important problems I encountered during the implementation. This is definitely not an exhaustive list, but those I mention are the ones that have had a major impact on the final result or have kept my attention for a significant amount of time.

6.3.1 Time Management

One of the biggest issues when making the program was to stay on schedule. Trips to China were scheduled long ahead of time, so there was no being-on-day-late kind of option. This forced me at some point to put aside the internal quality of the program, while ensuring that it did what it had to do in the eyes of the users. Therefore, the program, while it does exactly what it has to do, has major flaws in terms of internal coherence. These flaws have to be corrected in the future, in order to make the task easier for another programmer who would want to work on the program and add new units.

6.3.2 Mastering the NetBeans IDE

NetBeans is an amazing program, but it takes some time before you can figure out how to do some things. As time went by, and as I progressed further in the program, I discovered new ways of doing things, and usually those ways were more efficient than the ones I used before. But because time was such an issue (cf. Time Management), I could not go back and redo the other parts the new way. This also has made a great difference between the way the first pages of the unit were made and the way the last ones were.

6.3.3 Drag and Drop

Some exercises required a drag-and-drop interface. Java has a built in drag and drop class, but it is initially meant to manage drag-and-drop between two applications. This caused some problems because I needed to be able to move around my object back and forth, which was not possible with Java's built in Class, because the reference to the original object was not carried over the drop (i.e.: you could only drop a given information, but not the whole object). Eventually I decided to implement the drag and drop myself. It acts differently from usual drag and drop because it requires a click to drag and another click to drop. I tried to implement it with a release-click interface, but as I had problems with the interception of the MouseRelease event, I decided to stick with the solution I already had.

6.3.4 Fill In the Blanks

The exercises where you had to fill in the blanks were hard to implement too. The trick was to display written text, together with a JTextField, while making sure that if the sentence was long, it would be written on two or more lines. After many unsuccessful attempts to find a way to determine if the sentence was too long, it occurred to me that I could use a FlowLayout and put each single word of the sentence in a JLabel. The FlowLayout made the overflowing words display on the next line. The next thing I wanted to do was to find a method to parse a given sentence and display it correctly in a FlowLayout Panel. But as I started to have problems when sentences were over two lines long, I decided to carry on with something else, in order to be on schedule.

6.3.5 Checking the Answers

The last major problem I had was in respect of the checking of the answers. I wanted to find a way to sequentially read all JTextFields on a given page, without having to actually keep a list of all the pages' JTextFields; I thus needed to implement a generic method for checking the answers. Eventually, I chose to implement a CheckAnswers method for each page its own.

7 Interface

7.1 *Choice of a metaphor*

We saw in section 2.3.1 that using a metaphor can play a major role in helping the user understanding the program. Most CALL programs use a mini-world as metaphor, where the user can freely visit various rooms as if he were in a house or a school. However, this is mostly true for a program that leaves a high degree of freedom to the user. In WUFUN, the users' freedom is very restricted, so using the usual metaphor of a mini-world with an apparent freedom would mislead the user. Chinese culture, which is mostly influenced by Confucianism, is very much based on stages and cycles. Such a cycle and stages metaphor would be a great choice.

The idea for the metaphor is to consider each unit as a sphere belonging to a system of spheres. Each of those spheres is made of several elements (depicting the pages) that form a cycle. Each sphere would have a colour determined by the overall medal already obtained for the corresponding unit, so the user can see at a the progress and thus the stage he has gained is for each unit.

Even though the idea of the metaphor is clear, it is not entirely depicted in the interface on this version of the program. Indeed, the main objective of this test version of the program is not to test the efficiency of the program, but to evaluate Qing Ma's method. Moreover, the tests will be carried out in a supervised environment (more on this in chapter 8), so the tested users will be able to ask the supervisor for help. As time is an important factor, the design of a sophisticated interface is secondary, as all we want so far is to have a functional program that does what it is intended to do. It remains important to design the interface so that its later enhancement toward the researched metaphor will be easy.

7.2 Graphic choices

7.2.1 Colors

When thinking of a design for the interface, I thought that a good choice for the colors would make the use of the program friendlier. Colors that are rough to the eyes tend to give a headache and to tire out the user, while sober colors tend to take away some form of liveliness from the program. Initially I chose a blue and yellow interface, but it quickly turned out to be way too aggressive. I eventually chose to use white and a bluish violet as the main colors; white being used for the background and violet for the buttons and the frames. The text would be left in black.

7.2.2 Window size

Because you can never expect a computer to have a given resolution, I had to opt for a window size that would work on the most current resolution setups. An 800*600 screen resolution seemed to be the best choice, as it would be perfectly viewable in any higher resolution and it gets very rare to see anyone use any lower resolution.

7.3 Icons

Designing the icons in order to make them correctly suit the spheres-cycles metaphor requires the work of a graphics designer. This would have been costly and at this point, we preferred to first focus on making the program work and chose temporary icons that will eventually be replaced by icons actually related to the spheres-cycle metaphor. Yet, the icons still have to be wisely chosen in comparison to the related action.

The following temporary icons were chosen:



Home Icon: Go back to the welcome page. The link between the icon and the action is obvious; it just brings you to the home page.



Question Icon: Display the help (figure 7-1). Here again, the choice of this icon is trivial.



Exit Icon: Quit the program (figure 7-2). The image says it all.



Door Icon: Logout (figure 7-3). Here the interpretation may be less trivial. It can be related to leaving the class room. Definitely, this icon has to be changed to better suit the sphere/cycle metaphor.



Menu Icon: Go to the Menu. This icon is a sort of miniature of the unit menu. It is closely related to the sphere/cycle metaphor and will probably remain so.



Fast Arrow Icon: Go to the last accessible page. The meaning of this icon is trivial, as its use is widely accepted and widespread. It could be kept this way for the sphere/cycle metaphor, but another icon could also be thought of.



Yellow Arrow Icons: Go to Previous/Next page. The same remark as for the fast arrow is valid.



Dictionary: Translate (figure 7-4). Typical icon for a translation, it will very likely remain the same with the sphere/cycle metaphor



Speaker Icon: Play sound. The same remark as for the dictionary is valid.



Clap Icon: Go to “Overview of the story”. The idea of a clap comes from the fact that in the overview, the user will see small scenes that are part of the story.



Spyglass Icon: Go to “word focus”. A spyglass allows one to focus on something you would not normally see. This is just what the word focus is: a tool to help you understand words better. Here again, it is good for the cycle/sphere metaphor.



Book Icon: Go to “The Story”. This icon was chosen for the analogy to a storybook. However it definitely has to be changed because of its resemblance to the translation icon (dictionary). A storyteller icon might be a better choice.



Magic Hat Icon: Go to “Word Memorization Aids”. Word memorization aid is like a wizard who helps you remember words.



Teacher Icon: Go to “exercises”. The teacher can easily be associated to the exercises.



Yin-Yang Icon: Go to “Idiomatic expressions”. Idiomatic expressions are often words of wisdom. The Yin-Yang refers to wisdom, so idioms can be associated with this.



Jester Dog Icon: Go to “humor and true stories”. The jester is always associated to humor, so the link between humor and the jester dog is obvious.



Road Sign Icon: Change Unit. The road sign indicates a change of direction; this is just what changing unit does. Though, this icon should be changed because in the sphere/cycle metaphor this will bring the user to the sphere system, which has nothing to do with a road.

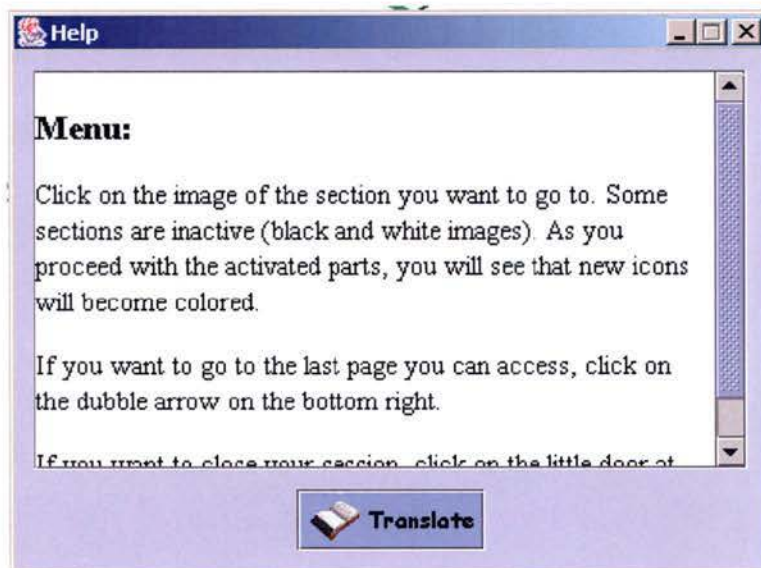


Fig 7-1 : Help display

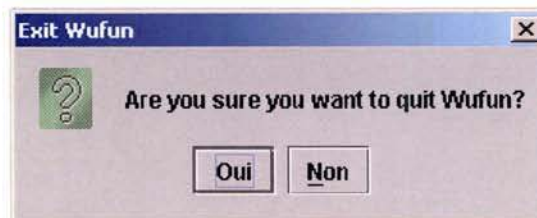


Fig 7-2 : Quit wufun

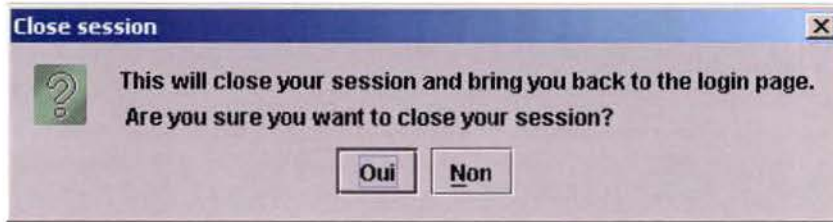


Fig 7-3 Logout

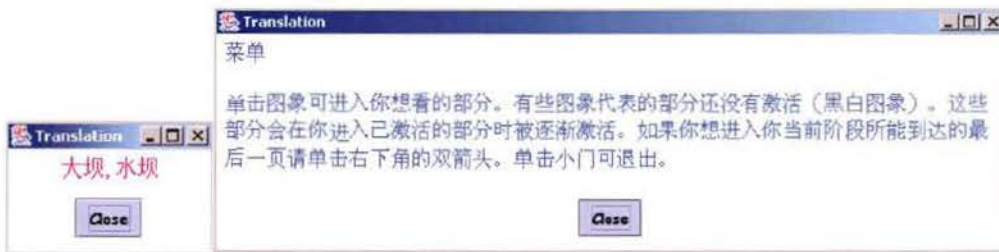


Fig 7-4 : Translation

7.4 Welcome Screen

When starting the program the user is first welcomed with a simple screen (figure 7-5). Clicking on start will bring him to the user identification page.



Fig 7-5 : The welcome screen

7.5 User Identification

The user identification page is as you would expect any user identification page to be (figure 7-6). The user can fill in his login and password and then click on “done”. If the login does not exist, the user is asked if he wants to create a new account (figure 7-7), in which case he will be taken to the user creation page. If the password is incorrect, a warning message displays (figure 7-8). Clicking on “new user” button brings to the user creation page.



Fig 7-6 : The Identification page



Fig 7-7 : Login Failure

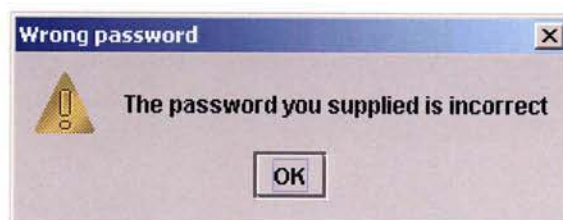


Fig 7-8 : Wrong Password

7.6 User Creation

The user creation screen provides an interface for the user to submit his personal information (figure 7-9). Clicking “done” will result in either an error message if some information is missing or if the login already exists or if the passwords don’t match (figure 7-10). Otherwise, the user is brought to the main menu and a message displays, confirming that his profile was created (figure 7-11).

New User Form

General Information

First Name*: Last Name*:

Age: Sex: m f Specialty:

Your Account

Your Login*: Your Password*:

Confirm Password*:

Done **Cancel**

* Compulsory fields

Fig 7-9 the user creation page

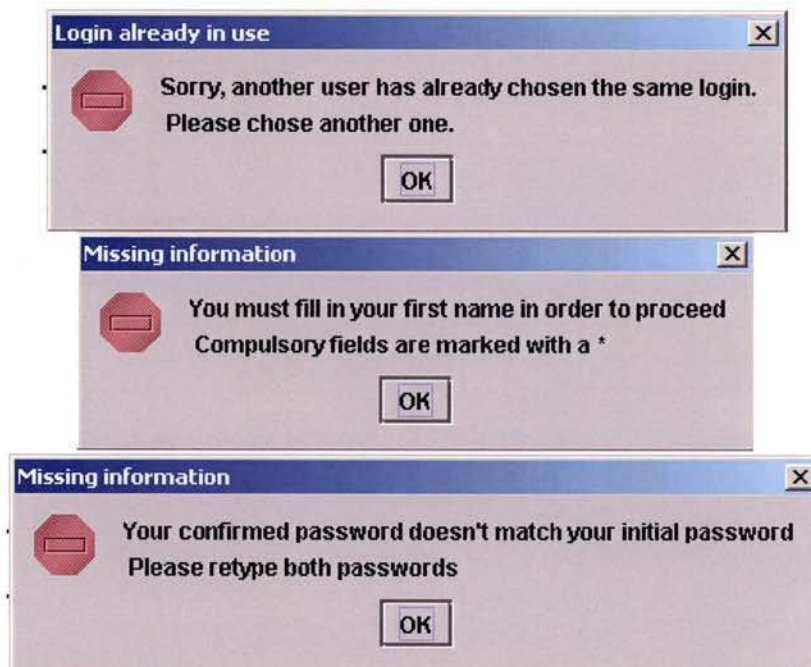


Fig 7-10 : User Creation errors

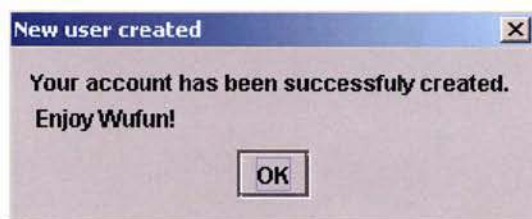


Fig 7-11 New user created

7.7 The Main Menu Screen

The metaphor of cycles is already present in this version of the menu (figures 7-12 and 7-13). The setup of the icons in a circle will stay the same as it clearly shows the idea of cycles.



Fig 7-12 : The main menu as seen when first beginning a unit.



Fig 7-13 : The main menu when all the pages have been completed

As you can see, the main page becomes more colorful as the user progresses in a unit. Also, the unit's overall medal is displayed at the center of the menu. This is to reflect the fact that the central focus of the user is to achieve a higher level of knowledge, illustrated by a better medal.

7.8 Navigating the Unit

The unit navigator is actually made up of two parts (figure 7-14):

- 1) The WufunBrowserPanel: This panel is where all the browsing icons are displayed
- 2) The WufunViewerPanel: This is where the actual pages are displayed

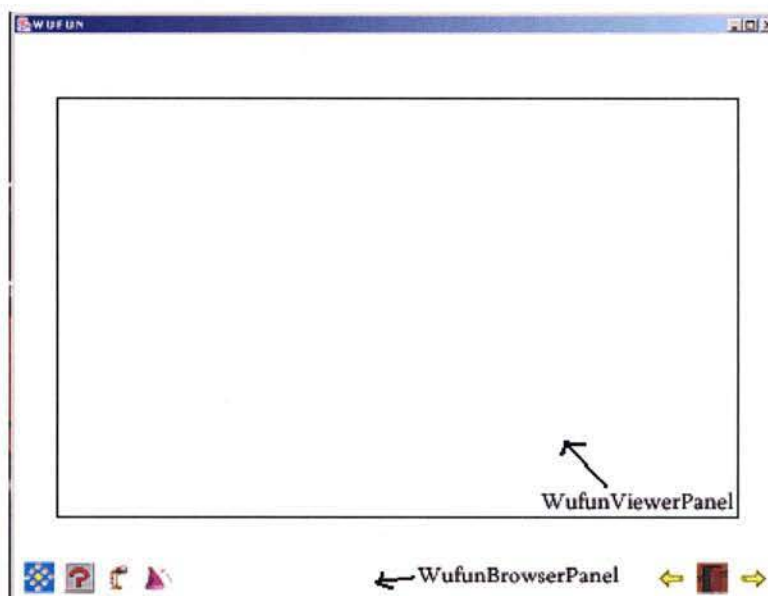


Fig 7-14 : the WufunBrowser/ViewerPanel

Pages that are exercise pages are also made up of two parts that are put inside a WufunViewerPanel (figure 7-15):

- 1) The WufunExercisePanel: This is where all the exercise-related buttons are displayed
- 2) The ExercisePanel: This is the actual exercise

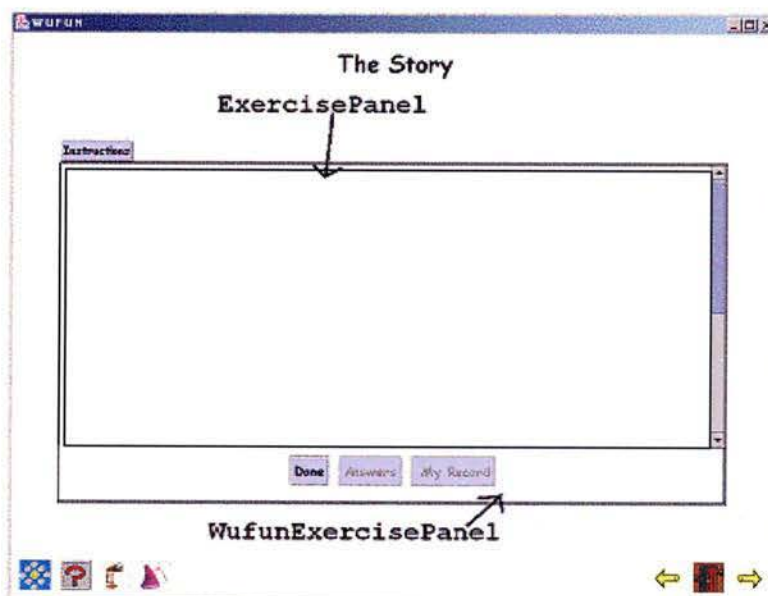


Fig 7-15 the WufunExercisePanel

Let us take a closer look at both the WufunExercisePanel and the WufunBrowserPanel:

In the WufunBrowserPanel, three icons may be disabled (grayed): the Spyglass Icon, the Magic Hat Icon and the Next Page Icon. Spyglass is disabled until the word focus page is accessible, the magic hat is disabled until the word memorization aids are accessible, and the next page arrow is obviously disabled if the next page is not accessible.

In the WufunExercisePanel, two buttons may be disabled: the "My Record" button that becomes enabled as soon as the user clicks on "done", and the "Answers" button that is enabled as soon as all the sounds have been listened.

The "done" button corrects the exercise and displays the score (figure 7-16)



Fig 7-16 : Score display

The “My Record” button displays the score again as in figure (7-16)
The “Answers” button replaces the answers given by the user with the correct answers.

7.9 The Pages

Now that we have explained how navigation works, we can easily go over the pages of Wufun. Note that on each of these pages, there is an instructions button. This button shows the same window as in figure 7-1 but gives the instructions for what the user has to do on each page.

7.9.1 Overview of the story

On the Overview of the story page (figure 7-17) the user has to view several images and listen to the explanations of the images there is. A little “image browser” stands there with buttons to go to the first/previous/next/last image. Upon clicking on the sound icon, the explanations are played. Only when he has once listened to the sound can he view the text.

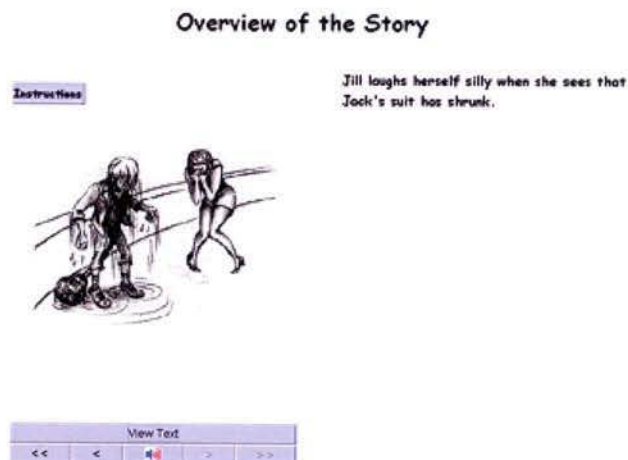


Fig 7-17 : Overview of the story

7.9.2 Word Focus

There are two versions of the word focus: the word focus page (fig 7-18) and the word focus popup (fig 7-19) which is a smaller, but not a different, version of the other one. The Word focus page is displayed when accessing it through the main menu or by the next-page-previous-page interface. The popup window is displayed when accessed through the WufunBrowserPanel. The user selects a word with the pulldown menu. When the word is clicked, the sound is played. The translation button translates only the word, not the explanations.

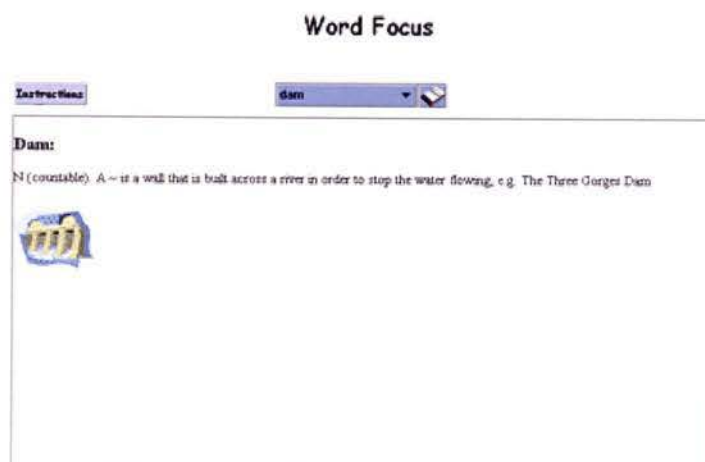


Fig 7-18 : The word focus page

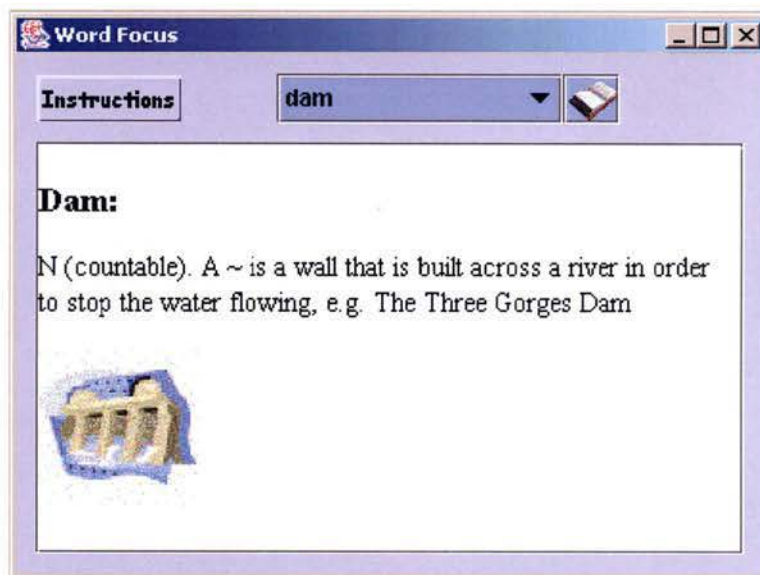


Fig 7-19 : The popup word focus

7.9.3 The Story

The story is a gapped text exercise. The user has to fill in the blanks in the provided text fields. At the front of each sentence you have a sound icon that becomes enabled when the answers are given by the user. Figure 7-20 shows the Story inside its WufunExercisePanel

The Story

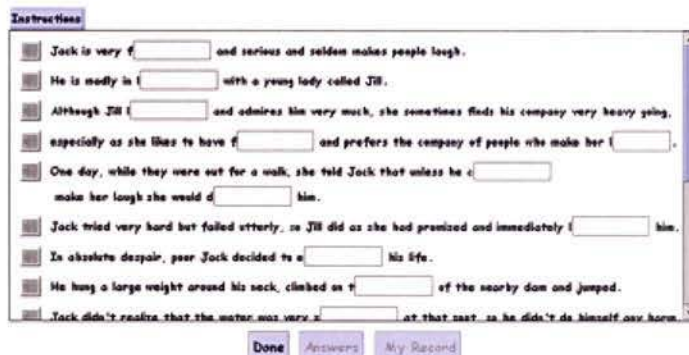


Fig 7-20 : The Story page

7.9.4 Word Memorization aids

As for the word focus page, there are two versions this page: the word memorization aids page (fig 7-21) and the word memorization aids popup (same as for the word focus popup) Like for word focus, the user selects a word with the pull-down menu and the sound of the word is played. Some word memorization aids have illustration sentences that can be listened to by using the sound button at the bottom.

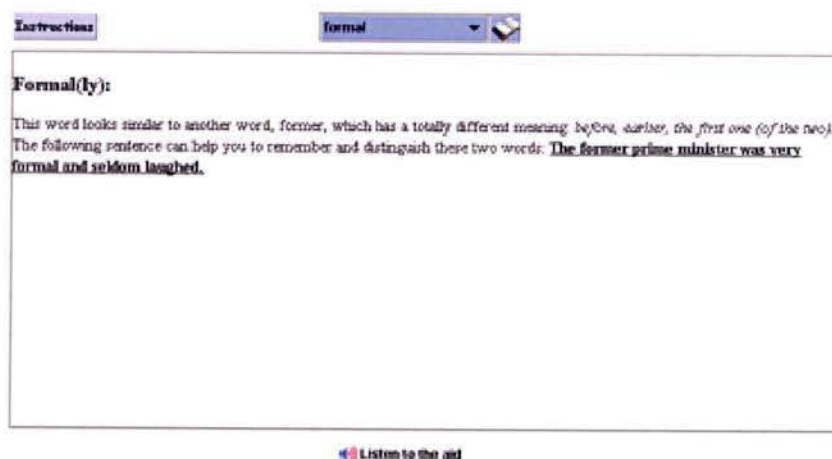


Fig 7-21 : word memorization aids

7.9.5 Exercise 1

A simple gapped text exercise. The user fills in the gaps and submits his answers. The sounds to the left of each sentence are initially disabled. They become enabled once the answers are given by the user.

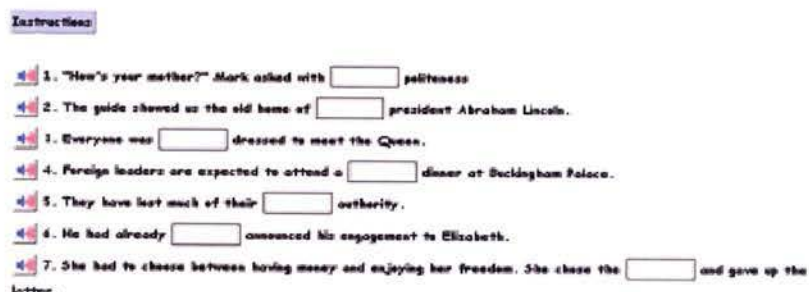


Fig 7-22 Exercise 1

7.9.6 Exercise 2

This exercise works with a drag-and-drop interface. The tags at the top can be clicked on and they become the dragged word. By clicking on the target text field, the word then displays in the field. If the user clicks somewhere else than on a text field, the dragged word goes back to the tags. If a word was already displayed, the new one replaces the old one, which returns to the tags at the top.



Fig 7-23 Exercise 2

7.9.7 Exercise 3

This is once again a gapped text. It works in the exact same way as exercise 1.



Fig 7-24 Exercise 3

7.9.8 Exercises 4 and 5

These two exercises work exactly the same way. It is some kind of hybrid between exercise 1 and 2. The user drags and drops the tags in the gaps.

Instructions

rubbish meal quantity sign dam policy despair stolen

1. There was a by the roadside: No dumping!
2. The thieves dumped the car and escaped on foot.
3. The government dumped the proposed of the preceding government.
4. They dumped the in the sea, thus polluting the environment.
5. The foreign company dumped a large of poor quality goods on the Belgian market.

Fig 7-25 : Exercise 4

Instructions

pouring dam look suggestion landed talk so

1. Your clothes are full of holes and make you ridiculous.
2. It is ridiculous to go out in the rain without an umbrella.
3. I like that boy very much because he is funny.
4. They consider my ridiculous and refuse to accept it.
5. I liked the you gave: it was so funny!

Fig 7-26 Exercise 5

7.9.9 Exercise 6

In exercise 6, the user has to associate a sentence marked with a number to another sentence marked with a letter. On the right side of the screen there is a number with a pull-down menu next to it. With this pull-down menu the user can select the corresponding letter.

Instructions

1. A funny thing happened to him on his way to class this morning
2. After he ate that watermelon, he felt funny.
3. He doesn't think that what he did was very funny.
4. He had a funny look on his face.
5. He has a funny way of looking at things.

- A. What he did was not in his view very nice.
- B. His outlook on life is rather strange.
- C. His stomach was a little upset after eating.
- D. He was wearing a strange expression.
- E. Something odd occurred earlier today.

1:

2:

3:

4:

5:

Fig 7-27 Exercise 6

7.9.10 Idioms and Proverbs

In the Idioms and Proverbs page, the user can click on one of the three sentence button. This will display the illustration and the little man below will be animated and explain the meaning of the idiom or proverb.

Instructions

He is in the depths of despair

I am fit to burst

I split my sides laughing



Fig 7-28 Idioms and Proverbs

7.9.11 Humor and true stories

In the humor and true stories page, the user can listen to the stories, try to figure out what they mean and then click on the “view text” button to display the text in the provided space. The “view text” button remains disabled until the sound button is clicked on.

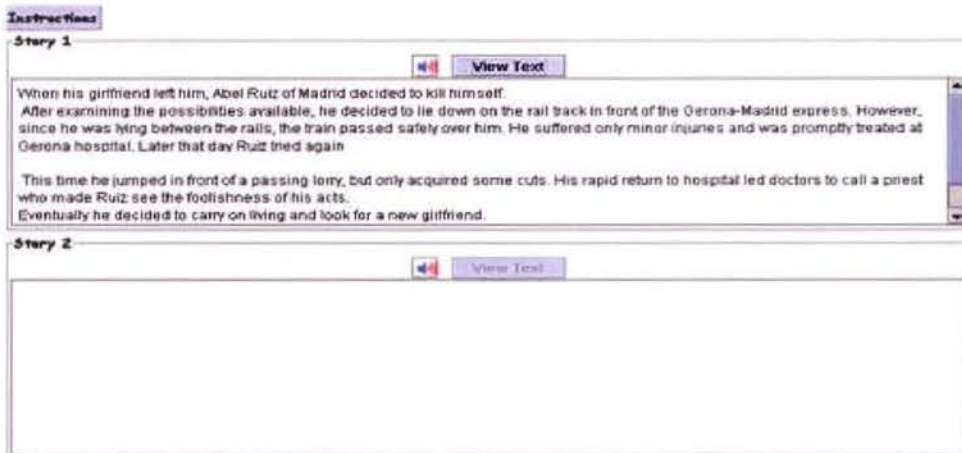


Fig 7-29 : Humor and true stories

7.9.12 Dictation Exercise

In the dictation exercise, the user can, listen to the dictation and try to write the text down in the space provided. Once he has listened to the text, he can view the text in order to check his transcription.

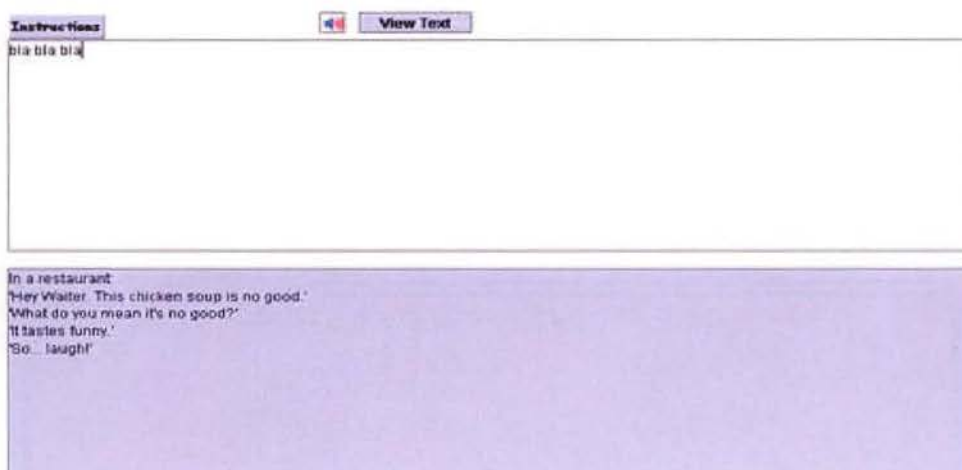


Fig 7-30 Dictation Exercise

8 Evaluation

8.1 Introduction

Now that the testing unit is finished, it will have to be tested by the final users. Two objectives are set with this evaluation:

- 1) Testing the efficiency of Qing Ma's method when used with computers.
- 2) Testing the receptivity of the students to the program.

The testing will be done at the Three Gorges University and will be carried out by Qing Ma.

8.2 Testing procedure

The testing will be done by Qing Ma, as it is part of her thesis. She decided upon a testing procedure that will work as follow:

A first test will be carried out in order to evaluate the students' receptive knowledge of the unit's vocabulary. This is done with a series of written exercises.

A second test is then carried out to evaluate the productive knowledge of the items that appear in Wufun. This is also done with written exercises where the students will have to produce sentences with elements from the tested unit.

After this a questionnaire is handed out in order to see what learning strategies the students use and what their expectations of the program are.

The students are then asked to work with the software through the whole unit. They can work as long as they want until they are either fed up with it or believe they have nothing more to learn.

After using the program, the students are given a second questionnaire in order to find out how satisfied they are with the program and the method as well as any comments and suggestions they may have.

The two first tests (receptive vocabulary and productive vocabulary) are then carried out once again in order to evaluate the improvement of the students after using the program.

Two months after the experiment, the two tests will be carried out for the third time in order to see how well the students remembered the lessons.

8.3 The Testing

The experiment was carried out by a group of 14 volunteer students at Three Gorges University in China. It was also carried out with another group of 23 designated students at the same University, but the results for this group are, as of today, not yet available.

8.3.1 Results of the receptive test

Table 8-1 shows the evolution of the score on the receptive test before (1.1) and after (1.2) using the program.

	1.1	1.2	Increase
1	16	24	8
2	17	23	6
3	17	22	5
4	18	21	3
5	15	19	4
6	15	20	5
7	11	21	10
8	15	23	8
9	12	22	10
10	15	20	5
11	13	22	9
12	16	24	8
13	11	22	11
14	19	22	3
Avg.	15	21,8	6,8

Table 8-1 : Results of the receptive test

The increase is obvious for all the tested students. An average increase of a little over 45% is definitely a good result. The method is thus efficient in the learning of receptive vocabulary.

8.3.2 Results of the productive test

Table 8-2 shows the evolution of the score on the productive test before (2.1) and after (2.2) using the program.

	2.1	2.2	Increase
1	12,5	16,5	4
2	9,5	11,5	2
3	7	16,5	9,5
4	13,5	18	4,5
5	9	13	4
6	11,5	15	3,5
7	10,5	19	8,5
8	15	20,5	5,5
9	8,5	13,5	5
10	9	15	6
11	11	18	9
12	13	17,5	4,5
13	7	16	9
14	13,5	19,5	6
Avg.	10,8	16,4	5,8

Table 8-2 : Results of the productive test

Here again, the increase is obvious for all the tested students. An average increase of over 53% shows that the method is efficient in the learning of productive vocabulary.

8.3.3 The usage of the program

Table 8-3 shows how the students have been using the program.

Total time is the total time, in minutes, spent on the program.

WMA is the number of times they referred to the word memorization aids page.

WF is the number of times they referred to the word focus page.

Ex Time is the total time in minutes spent on the exercises.

	Total time	WMA	WF	Ex Time
1	82	4	15	21
2	84	8	43	34
3	113	12	48	13
4	83	4	7	14
5	96	7	26	24
6	78	4	7	24
7	90	15	17	19
8	53	4	9	19
9	112	13	19	27
10	74	6	16	26
11	71	9	16	11
12	64	3	5	26
13	95	24	32	35
14	64	3	5	26
Avg.	83	8	19	23

Table 8-3 : Usage of the program

Note that the time spent on exercises includes the time to read the instructions and the time it took in order to give the answers for the first time; it does not include the time taken to further correct the answers and listen to the sounds. As there are 6 exercise pages, considering the average of 23 minutes spent on exercises, it roughly took 4 minutes per exercise, which means that the students didn't struggle to understand what they had to do, and how to answer the questions. This is positive as it means that the interface for doing exercises was intuitive, and this is just what we wanted.

We can see a great disparity in the word focus and word memorization aids referrals. The fact that some students consulted the word focus as much as 48 times probably means that they have been using the quick access to the word focus on the browser bar. The lesser use of word memorization aids was expected, as they are less directly helpful when doing exercises.

8.4 The results of the questionnaire

The first questionnaire was more an informative questionnaire for Qing Ma's use, so I'll only talk about the results of the second questionnaire. It consisted of 50 questions about the students' experience of the program, what they liked or disliked in the program, be it the method, the looks or the ease of use of the program. The answers to the questions were ratings ranging from 1 to 5. The full questionnaire can be found in the annexes.

Oppositely to previous sections, the results of the questionnaire are known for both groups, so they are taken in account here.

I will go over the questions that have a direct connexion with my own work.

8.4.1 The design of the exercises

When asked what they thought of the layout of the exercises, the students rated it 4.0 out of 5.0. This is very positive, as the exercises are very important for the learning process. A student not liking the layout of the exercises will not like using the program and it will have a bad impact on the method. Only one student gave it a rating of 2.

8.4.2 The scoring and answers checking system

When asked how they liked the scoring system, it received an average rating of 4.3 which shows that they are actually encouraged by the medals scoring system, which was exactly the objective. They also all thought that the scores obtained were fair (4.8).

The question "Do you like the answers checking system for the exercises?" was granted a 4.3 average rating, additionally, to the question if they thought that having sound answers before written answers was helping their practice of listening comprehension, they gave a 4.4 average rating.

8.4.3 The help and instructions system

The students liked the help system with a rating of 3.9 average, they also thought the instructions to be clear (3.9). Actually some students thought that it was not totally necessary to have a Chinese translation of the instructions, as the rating to the question "Do you think it is necessary to give Chinese translations?" was only of 3.9, with one student giving a rating of 2. Nevertheless some students also gave that question a rating of 5, so it is clear that the translation was necessary, but the instructions were clear enough for several students to be able to go without translation. The question "Do you often refer to the help system" was rated a 3.6 average this means that overall, the students thought the help system to be helpful but not always necessary.

8.4.4 The users' freedom

It is interesting to see that we really have two opposed opinions on the freedom's issue. To the question "Do you feel all right when you have to follow the guidelines instead of doing what you want?" some students gave a rating of 1 while some others gave it a rating of 5. The average comes down to 3.3; this is a bit lower than I would have expected, but it remains above average, which shows that overall, the users don't really mind this restriction.

8.4.5 Interface design

The users have been overall very receptive and positive about the design of the interface. They liked the general design of the program with a rating of 4.2 and rated the easiness of use of the program with an average 4.2 rating. They thought overall that the navigation system (the browser bar and the main menu) were logically designed with a rating of 4.0.

8.4.6 Graphic design

The graphic design was probably the most debated part in the questionnaire. The question "Do you like the colors used in the interface" was overall rated 3.6, and the white background was rated 3.2. This is definitely a part that can and has to be improved. The images used in the program for the icons or the illustrations were accepted more positively with a 3.9 average rating.

8.4.7 Sounds

The question was asked to the student if they liked the quality of the sounds. This was rated with an average of 4.2 which was expected because the sounds were recorded with a high quality.

8.4.8 Overall Satisfaction

Finally the students were surveyed about their overall satisfaction of the program. They said they enjoyed using the program with a rating of 4.0, which is a sign that the program is well on the way to be used as a self learning method. They also rated 3.9 the meeting of their initial expectations by the software. This is positive too. Finally, 93% of them said that they would like to use the program in the future when it will be expanded to more units.

8.5 Conclusion

As it is, the program proved to work perfectly, no user had to complaint about not being able to do what he wanted to do (to the exception of the freedom restriction of course). The results of the questionnaire were also positively encouraging, while showing that there were yet a few weaknesses to correct, especially in the colors. Qing Ma was also pleased with the result, which was for me the most important point.

9 Conclusion

For now over 40 years, Computer Assisted Language has been evolving, serving numerous purposes and always striving to improve the experience of language by the users. While today most CALL programs work with the idea that the users want to learn English to communicate, Wufun strives to actually generate that will to communicate that is lacking the Chinese students. This is what makes Wufun not just another language learning program, but more an innovative and uncommon program. Even though it is not the first program to work in cross cultural pragmatics, it stresses that CALL definitely has yet many paths to explore and tons of innovative applications yet to be discovered.

The evaluation of the program proved Qing Ma's method to be efficient, and the fact that the final users welcomed the program in a great majority and seemed to think that it met their expectations is very encouraging and allows hoping that more such projects will further be developed.

As of today, Wufun is only at an experimental state. The interface has yet to be improved in order to stick to the cycles and spheres metaphor. Qing Ma also suggested that the addition of videos would be a great improvement in the program. Also in terms of its internal implementation, the program is still a bit messy and some parts could have been implemented in a much better way. If I were to start the program over again, I'd try to make a unit creation interface in order to make the job of future programmers much easier.

On a more personal perspective, I found the development of this CALL program to be a very interesting and challenging experience in many points of view. I have had the chance to take part to all the stages of development of this project and had a taste of what the work of a programmer really is. It has been very instructive to take part in the many discussions, exchange mutual knowledge about linguistics, CALL theories and to get in touch with the Chinese culture, which I had no knowledge of before. I thought it to be very enriching to think about how people from another culture would perceive the program compared to the way I perceived it.

Now my hope is that Qing Ma's project will interest other people and bring up a greater interest in the cultural implications of CALL design.

10 Bibliography

[ATT]: AT&T Lab Research Text To Speech Interactive Multi-Lingual Demo
<http://www.research.att.com/projects/tts/demo.html>

[DUT]: Thierry Dutoit: A short introduction to text-to-speech synthesis
<http://tcts.fpms.ac.be/synthesis/mbrola.html>

[KEL et AL 1996]: Kelly, P., Vanparys, J., Li, X., Zimmer, C.: "A comparison of the perceptions and practices of Chinese and French-speaking Belgian university students in the learning of English: the prelude to an improved programme of lexical expansion". *ITL review of Applied Linguistics*. (113-114), pp. 275-303.

[LEV 1997]: Levy, M: "Computer Assisted Language Learning: Context and Conceptualization", Clarendon Press

[LEV 1999]: Levy, M: "Theory and Design in a Multimedia CALL Project in Cross-Cultural Pragmatics", *Swets & Zeitlinger, Computer Assisted Language Learning*, Vol. 12, No. 1, pp. 29-57

[LEV 2002]: Levy, M: "CALL by design: discourse, products and processes", *Cambridge University Press, ReCALL*, Vol. 14, No 1, pp 58-84

[LON & VAN 2001]: Lonfils, C & Vanparys, J: "How To Design User-Friendly Call Interfaces", *Swets & Zeitlinger, Computer Assisted Language Learning*, Vol. 14, No. 5, pp. 405-417

[MA 2002]: Ma, Q: "Theory and design in a Multimedia CALL project: WUFUN"

11 Annexes

The annexes can be found in the included CD Rom. Here follows the descriptions of the different CD's components:

Wufun.zip:

This archive file contains the compiled program, the source code and a few batch files to run the program. In order to run the program, the Java runtime environment has to be installed. If you work with windows, you can install this by running j2re-1_3_0_02-win-i.exe that is also found in this archive. Once this is done, you can extract the whole archive on your disk and run the program with wufun.bat

A Good Laugh.ppt:

This Power Point presentation is the initial description of the testing unit by Qing Ma.

Questionnaire before the experiment.doc:

This is the questionnaire that was handed out to the students before the tests were carried out.

Questionnaire after the experiment.doc:

This is the questionnaire that was handed out to the students after the tests were carried out.

thesis.doc, thesis.pdf:

These files are the Word and Acrobat Reader versions of this document.