NANYANG TECHNOLOGICAL UNIVERSITY

"Technologies for Language Learning"

By

Madhuvanti Pawar

School of Computer Engineering
2001

ABSTRACT

Once relegated to "novelty" status, Computer Assisted Language Learning (CALL) is fast achieving the recognition and popularity it deserves. Keeping the criteria for the ideally effective language instruction software in perspective and with a view to devising a near perfect solution for the learner as well as the teacher's needs, a tutoring system tailored for Kirrkirr users was developed. This report presents the Tutor, QuizMaster, Kirrkirr Profiles and HTML Generation that together with the original Kirrkirr form an effective, integrated and sophisticated tool to aid in Second Language Acquisition (SLA) for Kirrkirr Users. In reviewing the current version of Kirrkirr, the author demonstrates how to create a robust interactive learning environment that exploits the strengths of technology while working around its limitations. In conclusion, we draw on our review of these tools to identify directions of future research that might improve both the design and the overall performance of Kirrkirr.

ACKNOWLEDGEMENTS

CONTENTS

TABLE OF FIGURES

Chapter 1 - Introduction

1.1   Background Information

a. Language Instruction Software

A decade ago, the use of computers in the language classroom was of concern only to a small number of specialists. However, recent years have shown an explosion of interest in using computers for language teaching, learning and testing. Computers started off with playing the role of mechanical, non-judgmental tutors, which allowed students to work at their individual pace at repetitive language drills during CALL's (Computer Aided Language Learning) *behavioristic phase* (Warschauer & Healey, 1998). In its *communicative* phase (Warschauer & Healey, 1998), computer-based activities helped students perceive learning as a process of discovery, expression, and development. Presently, with integrative CALL (Warschauer, 1996b), came a new perspective on technology and language learning, a perspective that sought to integrate both various skills (e.g., listening, speaking, reading, and writing) and technology more fully into the language learning process.

While it may seem that CALL implies the sole involvement of the computer in language instruction, what it intends, arguably, is to provide the software as a complement to the language instructor's efforts. One of the reasons that administrators buy language instruction software today according to Warschauer & Healey (1998), is with the thought of providing a teaching solution that will:

- Provide realistic, native-speaker models of the language in a variety of media
- Determine the best next step for the learner and provide practice with that skill area
- Record what the student has done, along with an evaluation
- Be available at any hour and require no additional pay or benefits

It is not seen as a substitute for the instructor, but as a more flexible tutor that instructors can use for an add-on to instruction or that will be placed in libraries as language references and resources. So, what makes a good Tutoring application and can there be a unified solution to all the challenges that language learners face? This is the question that addresses the research problem here.

b. *Kirrkirr*: An Overview

Kirrkirr (the Warlpiri word for the kind of 'click' sound made by a young kangaroo) was originally developed by Jansz (1998). It is a bilingual dictionary of English and Warlpiri (an Australian Aboriginal language) developed for a community of about 3000 people in Central Australia, using Java and XML. The project's primary objective was to raise the literacy level of the Warlpiri community regardless of their level of language competence and to make the learning process fun and enjoyable through its interactive nature and highly customisable interface.
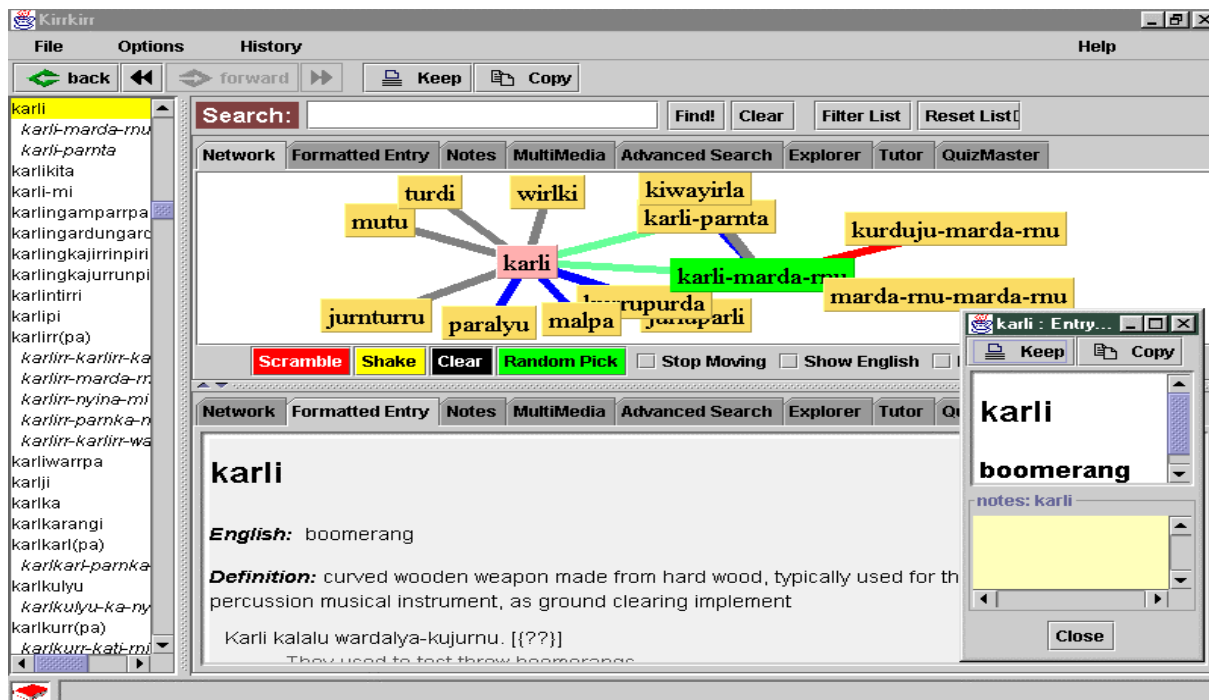
Figure 1: Components of *Kirrkirr*

It consists of many interesting features not found in traditional electronic dictionaries.

- Graph Layout: Rather than just looking up a word and returning the meaning of the word to the learner the program seeks to recreate the learning experience of flipping through a real dictionary (seeing and learning new words in the process). To achieve this, the program provides the learners with a visual animated representation (see Figure 1) of linkages of the word of interest to other words (synonym, antonym). This makes the learning experience effective, as the learners can pick up more words in the process.

- Formatted_Entries: Through the Formatted Entries tab, the learner can access the entry of his selected word. Dictionary entries are formatted in the familiar HTML format, with sub-entries nicely laid out with different colours. The learner can also click on a cross-referenced word and have the system jump to the entry for that word.

- Notes-Taking: To enhance learning, a note-taking feature is also included in the program, This enables the learner to make personal notes, simulating the real life activity of taking notes during lectures. This is particularly useful if the learner wants to note down questions/doubts regarding the word which he intends to follow up with his Warlpiri instructor. The notes taken can also be saved for revisions in future sessions.

- Multimedia: The learner can hear the correct pronounciation of the word in Warlpiri through the multimedia tab. The feature greatly helps the learner to acquire spoken Warlpiri.

- Search Function:The search function was included to enable the learner to query the database. Advance search capabilites include searching using regular expresions, fuzzy spelling or just plain spelling. Fuzzy spelling was also included to help the learners look for a word by the way it sounds if they are unaware of its spelling. Common spelling mistakes were also tolerated by the program: in the occurrence of these mistakes, the program will autocorrect the spelling before performing the search.

c. The need for another CALL tool

Language learning and testing modules were required to be integrated into Kirrkirr so that it could be more responsive to new learners of Warlpiri. Why could an existing tutoring and testing module not be integrated with Kirrkirr? Why was the development of yet another language instruction module required? The following help answer this vital question:

- Differing needs of a Target Language: One piece of software cannot be suitable for the process of acquisition of any and every target language. With the considerable differences between the target languages, the natures of the CALL tools for different languages also change significantly.

- Social and economic changes: The shift to global information-based economies has meant a dramatic increase in the need to deal with large amounts of information and to communicate across languages and cultures. In this information-rich time, students need the ability to respond and adapt to changes rather than training in a single way to approach a task. Hence CALL paradigms must adapt too in order to be effective (Warschauer & Healey, 1998).

- Change in Instructor roles: In these days of global interconnectedness, instructors are rarely the sole source of language information, and the literary corpus that may have been the basis of their foreign language training is not the only body of knowledge worth learning. The assumption from cognitive theory is that instructors do not pour information from their store into the heads of waiting and willing students, but that students actively interpret and organize the information they are given, fitting it into prior knowledge or revising prior knowledge in the light of what they have learned (Van Dijk & Kintsch, 1983; Dole, et al., 1991). As a result of all these changes, the instructor has become a facilitator of learning rather than the font of wisdom, and will find, select, and offer information in a variety of ways on the basis of what his/her students must learn in order to meet diverse needs.

With this in mind, a CALL tool specifically designed for the language of Warlpiri was needed, which could be ideally incorporated with the Kirrkirr dictionary browser. This would enhance Kirrkirr 's functionality as language instructional software.

1.2   Objectives and Scope

This project's primary objective was to improve Kirrkirr as a Language Instructional software. It involved the design and development of the Tutor and the QuizMaster for Kirrkirr. Tutor is a module that helps learners learn Warlpiri and Quizmaster's job is to test the learner's knowledge of the language. It is hoped that with Kirrkirr, they will transform it into an integrated and comprehensive language acquisition tool.

To further enhance the functionality and effectiveness of Kirrkirr as a CALL tool, additional features were added in. The functionalities of Profiles and the ability to generate HTML using the specified XSL file. It was hoped that these brought Kirrkirr one step closer to being a completely learner-friendly and feature-packed CALL tool.

1.3   Main Accomplishments

The main objectives mentioned above were accomplished, notably:

- ✓ A Warlpiri teaching <u>Tutor module</u> for Kirrkirr, which transforms it from an enhanced dictionary to a CALL tool, was successfully developed and integrated into Kirrkirr.

- ✓ A testing module called <u>QuizMaster</u> for Kirrkirr, which enhances the status of Kirrkirr to an integrated CALL and CBT (Computer Based Testing) application.

✓ <u>Profiles</u> – this allows different learners to have their own personalized settings. The settings include XSL file settings, Network settings and a record of the History of the learner's browsing.

✓ <u>Generation of HTML file</u> – this feature gives the user an easy way to create a customized dictionary in HTML. Once the learner has selected certain words, Kirrkirr can now make an HTML file with the definitions for all those words (using XSLT to translate into HTML). The learner can then use this file for any purpose that he/she desires.

1.4   Report Overview

The main scope of this thesis was the conversion of Kirrkirr into a CALL tool and as such deal with the various aspects of such a software and its effectiveness and the ramifications of those dealings on the overall evaluation of Kirrkirr. The organization of this report is as follows:

Chapter 2: talks about how a second language is acquired and thus what factors can contribute to making the process effective and meaningful. The author does so by connecting a set of hypotheses to the Second Language Acquisition (SLA) process and then formulating a related set of criteria that ought to be present in a tool that intends to facilitate SLA.

Chapter 3: walks through the software development cycle of Tutor, the teaching tool. It talks about the numerous design considerations and implementation strategies employed by the author to utilize the Object Oriented nature of Java and the existence of a number of utility methods from other classes. The implementation is broken down into two areas. One, the content, that which is displayed on the Tutor and the other the program, the actual software code.

Chapter 4: describes the Computer-Based Testing component added to Kirrkirr - The QuizMaster. Being an evaluation tool, this chapter looks in depth at the design considerations for QuizMaster. It then goes on to explain the design and the implementation methodologies employed to have an operational QuizMaster.

Chapter 5: Two more "tools" were included in this project, namely Profiles and HTML generation. This chapter explains each of these.

Chapter 6: deals with the evaluation of the Kirrkirr with the Tutor, QuizMaster, Profiles and HTML Generation. A usability evaluation walkthrough of Kirrkirr is developed is presented.

Chapter 7: Conclusions and Future Work summarizes and concludes the thesis and describes further work possible as a result of the successful completion of the thesis.

Chapter - 2 Factors making an effective CALL tool

What makes a learning tool effective? To address this issue, it is vital to understand the processes involved in second language acquisition and then delving into HOW those processes can be facilitated. This is why software designers consult Second Language Acquisition (SLA) theory, research and principles to suggest effective CALL designs.

However, SLA literature contains work representing diverse perspectives and approaches for investigating the process of second language development. This is why when addressing applied questions such as design and evaluation of CALL, it is necessary to select from the many approaches those that are relevant. Once the model of SLA is clearly in view, hypotheses outlined to smoothen the learning curve can be suitably formulated and then used to evaluate the effectiveness of the learning aid.

2.1   A Relevant Model of Second Language Acquisition

Chapelle (1998) hails this model as the one most useful or applicable to CALL software developers. It attempts to articulate what makes input comprehensible and how it is processed to influence the development of the learner's linguistic knowledge (Long, 1996). The model illustrated in figure 2 is a simplified version of the one outlined by Gass (1997). This Model aptly points out that:

- The target language input acts as the potential starting point for acquiring a second language.
- A lot of comprehensible input is what is needed for SLA.
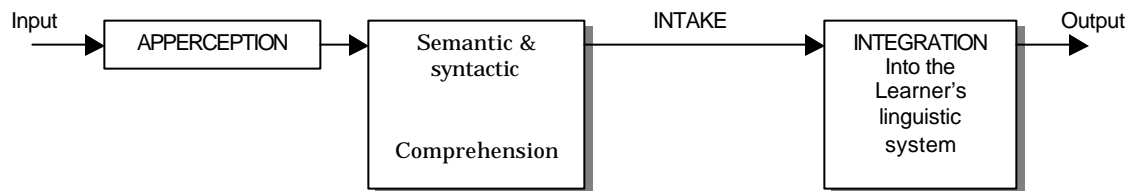
Figure 2: Basic components in the SLA process

Starting from the left of figure 2, INPUT refers to the target language that the learner is exposed to. Much target language input goes over the learners' head but only that which is apperceived has the potential to be acquired. An important aspect of the learners' APPERCEPTION is their noticing aspects of the input. Therefore, according to Chapelle (1998), one concern for designing instructional materials may be to include features that prompt learners to notice important aspects of the language.

The next box, COMPREHENSION, represents the hypothesis that understanding of the semantic content of a message can be accomplished either with or without any comprehension of the syntax. Semantic comprehension is not expected to help in the acquisition of the syntactic system because it may be accomplished through the recognition of isolated lexical items and interpretation of non-linguistic cues. When comprehension takes place through a combination of semantic and syntactic processing, the linguistic characteristics of the input can become INTAKE, that is, comprehended language that holds the potential for developing the learners' linguistic ability.

INTEGRATION is comprised of the processes for using or holding the intake in short term memory to influence the development of the linguistic system, which in turn affects the second language OUTPUT that the learner produces. The output is an observable result of the process, but it is also considered an important contributor, according to Chapelle (1998), to linguistic development in two ways:

- First, producing linguistic output forces learners to use the syntactic system and therefore to develop this aspect of their ability.

- Second, it elicits subsequent input from assessors, some of which may contain indications of problems with the learner's output which will result in the learner's noticing his/her mistakes, making new hypotheses, and producing more output. This process, referred to as *negotiation of meaning*, is believed to facilitate second language development (Long, 1996).

2.2 Hypotheses Relevant for Developing CALL

Based on the model of SLA in Figure 2 and its supporting theory and research, hypotheses about ideal conditions for SLA can be articulated (Long, 1996; Pica, 1994) in terms that are relevant for developing CALL. The hypotheses developed by Chapelle (1998) are reviewed here:

a. The linguistic characteristics of target language input need to be made salient

Even though Krashen (1982) believed that all target language input facilitates Second Language (L2) acquisition, current theory advances beyond that by pointing out that there exists useless target language noise among the target language input that may influence language development. The latter is different from the former because it is "apperceived" by the learner. Therefore, a learner's noticing of linguistic input plays an important role in making unknown target language forms into known and used forms (Schmidt & Frota, 1986).

Experimental research has shown that highlighting input in materials to prompt learners to notice particular syntactic forms positively influenced their acquisition (Doughty, 1991). In other words, development of principles for CALL design methods require effective "input enhancement" (Smith, 1991) and the instructional materials should facilitate apperception of input.

b. Learners should receive help in understanding semantic and syntactic aspects of linguistic input

Input can be defined as "*the potentially processable language, data which are made available by chance or by design to the language learner*" (Smith, 1993). Only a subset of all "*the potentially processable language data" that the learner receives is actually useful for the learner's language development*" (p. 167). Thus Input contains linguistic forms that the learner does not know. As a consequence, the learner needs help with specifics of the input in order to comprehend it both semantically and syntactically. This help, sometimes referred to as "modification" of the input, can consist of features like simplification, elaboration, or added redundancy (Larsen-Freeman & Long, 1991).

c. Learners need to have opportunities to produce target language output

Like input, which can be either uncomprehended noise or valuable for acquisition, output can be produced mindlessly or it can be created by the learner under conditions that facilitate acquisition. The latter type of production is called "comprehensible output" (Swain, 1985). It is target language that is intended to convey meaning to an interlocutor. In other words, not all production qualifies as valuable comprehensible output. It may be important that learners have an audience for the linguistic output they produce so that they attempt to use the language to construct meanings for communication rather than solely for practice.

d. Learners need to notice errors in their own output

According to Swain and Lapkin (1995):

"In producing the L2, a learner will on occasion become aware of (i.e., notice) a linguistic problem (brought to his/her attention either by external feedback (e.g., clarification requests) or internal

feedback). Noticing a problem 'pushes' the learner to modify his/her output. In doing so, the learner may sometimes be forced into a more syntactic processing mode than might occur in comprehension." (p. 373)

This hypothesis states that the syntactic mode of processing helps learners to improve the accuracy of their existing grammatical knowledge (Nobuyoshi & Ellis, 1993). The process of noticing can occur either through learners' own reflection and monitoring or through triggers provided by others.

e. Learners need to correct their linguistic output

When errors are recognized in comprehensible output, the process of the learner's self-correction is also believed to be beneficial. Error correction affords the opportunity to "focus on form" (Long, 1988). Focus on form is expected to be beneficial when it occurs during the process of attempting to construct meanings.

2.3　Suggested Criteria for CALL Software Development

As Chapelle (1998) aptly points out in a single sentence, "the idea of an effective CALL tool is to have software that uses the power of the computer to offer easy interaction with the material to be learned", including meaningful feedback and guidance; comprehensible information designed to fit the learning style of individual students; and ways for students to carry communication beyond an individual computer screen. Using each of the hypotheses about SLA, a corresponding suggestion for increasing effectiveness of CALL can be made.

a. Making key linguistic characteristics salient

Linguistic features can be made salient in CALL activities by highlighting them in a different color when they appear in writing on the screen. Relevant linguistic features can be chosen by the CALL designer or the instructor.

b. Offering modifications of linguistic input

Linguistic input can be provided through either written or aural language. Modifications of input can come in the form of repetition, simplification through restatements, non-verbal cues, decreased speed, reference materials, and change of input mode. "Modifications" might be provided by offering definitions of each of the words in a passage and simplifications of the phrases at the learners' request. This way, the instructional material should provide learners with comprehensible input rather than just input.

c. Providing opportunities for "comprehensible output"

Comprehensible output--either written or spoken--must be produced with the expectation that it is going to be "understood." In an interactive story, for example, the learner might participate in an episode with an academic adviser, who would pose a question such as "Have you decided what you'd like to take in the fall?" The learner would need to respond to the question with something like "I want to take Anthropology 204," or "Do you think I could take Anthropology 204?" It is important that learners expect their responses to be comprehended if they are syntactically well formed and pragmatically appropriate. "This expectation pushes them to attempt to use the target language forms attentively and constructively," points out Chapelle.

d. Providing opportunities for learners to notice their errors

The output that learners produce in CALL needs to be analyzed by the computer and/or reflected upon by the learners to identify errors. Output containing errors points out areas in which the learners' linguistic system contains gaps or misconceptions about target language use, and therefore these mark the key areas to which their attention should be drawn. The learner must have the opportunity to recheck the answer before entering it and hence the chance to notice errors.

e. Providing opportunities for learners to correct their linguistic output

When learners have focused their attention on their errors, they need to have the opportunity to correct them. They might be left to their own devices to make the corrections, but they might also be provided detailed error-specific help or access to more general reference materials.

f. Supporting modified interaction between the learner and the computer

The suggestions above imply that the learner engages in some form of interaction with the computer. These interactions need to move the learner toward a task goal and stop progress along the way to focus on the language. For example, in the advising interview mentioned above, a successful interaction between the learner and the program will result in a fall semester schedule for the student. The program's capability for engaging in such a goal-oriented conversation with natural language would require more than language recognition at the word and sentence level. It would require a "knowledge base" about the courses, schedules, and advising routines**.**

g. User Interface design

 Having a way to navigate and manage all of the facts at the learner's fingertips is essential. In the computer arena, "user interface"--the way software is written so that people will understand what to do with it--is a perennial concern. While much has been made of the graphical user interface of the Macintosh and Windows, replacing text with graphical objects (icons) is not enough to make software intelligible, much less effective.

h. Context AND Data

 Learners need help with more than the mechanics of operating a software program. They also need to know how to make the best use of it for their own purposes. Research has indicated that learners don't always know how to fit new information into an appropriate framework (Dole, et al., 1991; Gay, 1987; Van Dijk & Kintsch, 1983), so effective software has to give them context as well as data. CALL programs that respond to user input with nothing more than "Right" and "Wrong, try again" are clearly less helpful, thus less "intelligent" in these terms than they should be. Far better is software that tracks learner answers and looks for patterns, responding not only with whether the answer was correct but also *why* it was right or wrong and offering suggestions for further study, going on to a more advanced level or doing some extra work at the current or a previous level.

i. Control

Learners also often fall short in their ability to apply appropriate learning strategies to material (Oxford, 1990; Healey, 1993; Rubin, 1987), so effective software should be able to take naive learners in hand and help them along in figuring out how to use the software effectively, as an instructor would. Quite a bit of research has explored the idea of learner vs. computer control in computer-assisted instruction (Gay, 1987; Robinson, 1989; Johanesen & Tennyson, 1983; Schaeffer, 1981; inter alia). The general view is that learners, especially adults, feel better when they have a sense of control over the program. The most dictatorial programs are those that do not let the learner quit when he or she is ready to do so. The most flexible give learners the choice of what to do, when to do it, and for how long. Guided freedom would be a feature of an effective CALL, where the program would make suggestions, but the learner would make the choices.

j. Integration of Media

Another direction in current software is the integration of media. Foreign language instructors are particularly helped by access to a variety of media to help make the language come alive to students for whom it is largely a distant abstraction. Effective CALL will fit the medium to the learner, ensuring that the media work in concert to enhance understanding.

Chapter - 3 The Tutor

The Tutor was designed, developed and integrated into Kirrkirr, as its name suggests, to teach the dictionary users the language of Warlpiri. Written in Java using the Swing API, the SAX parser and the Document Object Model, it provides an easy insight into the language and also gets the users familiar with Kirrkirr using a hands-on approach.

Tutor is highly configurable. If the teacher wishes to change the approach or the information to be given to the learners, new worksheets can be added. Also multiple worksheets can be made available to the learners such that they can switch from one kind of a worksheet to another according to their preference and their vocabulary level. Currently, Tutor includes a Warlpiri worksheet by Simpson (2000) attached as appendix A1.

3.1   Design Considerations

Interactive applications require major design considerations compared to those simply working at the back-end. For a learner of second language, it is problematic enough learning the language; it would be even more cumbersome if his/her faculties were also to be engaged in getting familiar with the CALL tool itself. Especially with a learning tool, careful thought has to be put into the design of the code, how it will work and its User Interface to cater to the needs of the human learner and to make the experience for him/her as worthwhile as possible. The Tutor was hence designed with the intention of keeping the learning curve of the new learner with the software, as gradual as possible, so that he/she doesn't get lost in HOW to use the software itself and the attention remains on the language learning aspect. As such, the software was designed intuitively with a second language learner's perspective in view. The following are a few important pointers to the numerous design considerations that went into Tutor:

- Customizability - Tutor was not only required to be easy to use and un-intimidating to a new user but also interesting and appealing to an experienced user. While it is undesirable to have an application so feature packed that a novice user gets lost in the bevy of features, it is also pointless to develop a CALL module which bores the learner after awhile. This is important as one of the vital language learning strategies is repetition and it is definitely not desirable for an experienced learner to not want to use a CALL tool, simply because its boring. Tutor was hence designed to have a pleasing appearance and capable of changing to the desired worksheet and hence the difficulty level that the teacher might want to use with the learner.

- Modularity - Even though Tutor was developed for Kirrkirr, it was designed in such a way that it could easily be incorporated into any Java application or even made to exist as an independent CALL tool as occasion would demand. It was also required that the Tutor not be limited to a single language and be able to gracefully adapt to any change in its parent application. The Tutor was hence designed as a separate JAVA class that can be included in any JAVA program and was designed in a way that makes it capable of being used with any foreign language that a teacher may wish the learners to acquire.

- Facilitation of the learning process - Tutor has been designed on the basis of generally accepted theories about the way the mental lexicon is structured and operates. The natural word acquisition process (as this occurs in first language acquisition) consists of gradual acquisition of the various properties of a word through repeated exposures in a wide range of authentic contexts illustrative of its various features. Bearing this in mind, we are faced with a dilemma in an instructed second-language learning situation. On the one hand, there is not enough time for exposure to new words of the same intensity as in L1 (first language) acquisition. On the other hand, superficial exposure leads to shallow processing which fails to establish enough associations and links with other words for solid storage and efficient retrieval.
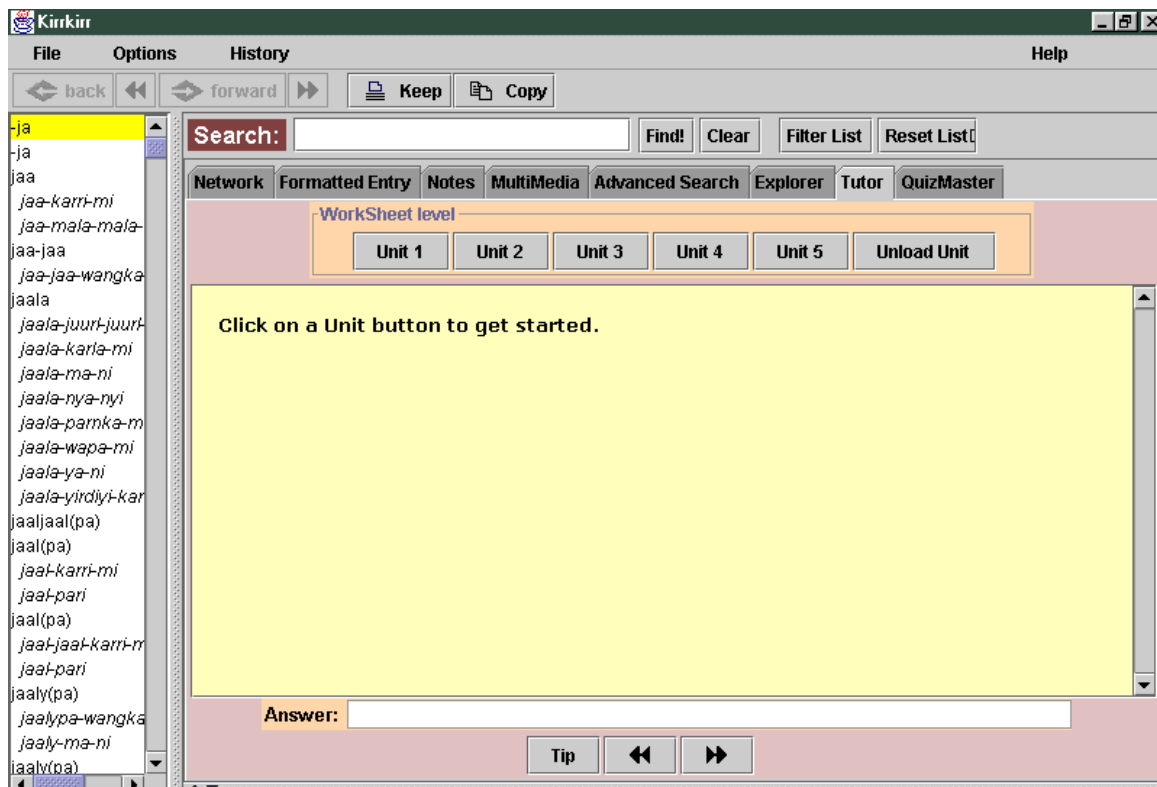
Figure 3: TUTOR interface

Obviously, there is no easy solution to this dilemma. The most realistic approach seems to be to create an environment that is maximally conducive to learning new words by striking a balance between the two contradictory demands. Furthermore, Mondria (1996) presents evidence that vocabulary test scores for two conditions (given vs. inferred meaning) indicated that there is no difference in long term retention effects between the two presentation methods and that, in teaching new words, giving the meaning is a more efficient method than having learners contextually infer it, because it takes less time. The Tutor hence intends to be effective by speeding up the acquisition process; it takes the learners systematically through the various stages by exposing them to carefully selected L2 material which illustrates the salient features of the new L2 word and/or the differences between the L2 word and its nearest L1 equivalent or counterpart.

3.2   Design

Tutor was designed to receive input from instructor - designed worksheets and generate files containing the output of the learners. Worksheets contain the questions along with the tips that the instructor perceives as useful to the language acquisition process. As figure 4 shows, tutor solely reads from the worksheet and displays the questions to the learners who then answer the questions. The answers are then written to a file for the instructor to evaluate and give feedback on. A sample of such an answer file generated is attached in the appendix A3.

This design facilitates the use of different worksheets and any language that the Tutor may be required to teach, as this is now dependant on the Worksheets created by the instructor. The Tutor simply provides a mechanism by which the information in the Worksheet is presented to the learners. This design also frees the programming aspect of the module from the linguistic aspect. The instructor, in this case, does not need to program. He/she just ought to be able to present the questions in a particular format.
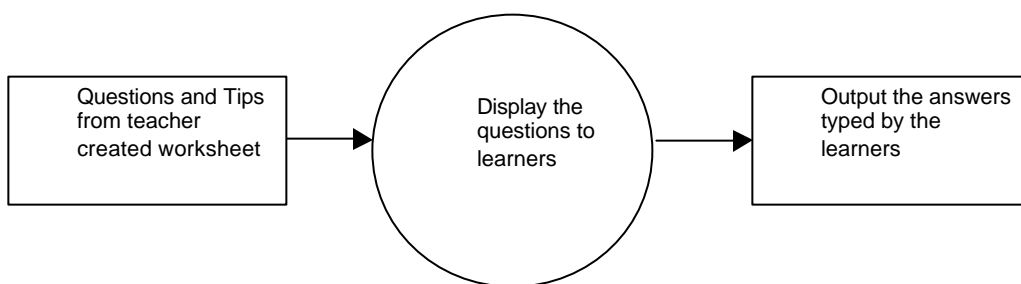
Figure 4: Program Flow of the Tutor

As illustrated in Figure 4, the learner starts by clicking on any of the "Unit" buttons to load a Worksheet. Any time during the lesson, the learner can unload the worksheet by clicking on the "Unload Unit" button. The lessons are displayed on the main yellow textbox and the answers to the questions are to be typed in the answer textbox below it. To navigate between the questions, the respective arrowed buttons are clicked. The "Tip" button displays the description of every question and may help in providing hints to the answers. When the end of the Unit is reached, the Unit can be unloaded and an answer file containing all the answers is generated for evaluation purposes.

This design was adopted in close accordance to the factors for effective CALL tools researched earlier. The Tip button was used so that the characteristic features of Warlpiri could be made "salient". Since the source of all linguistic information is the instructor, the Tutor directly obtains this information from the worksheet created by the instructor. The instructor, while creating the worksheet keeps in mind the key points of the target language that need to be brought to the learner's attention and writes them into the "DESCRIPTION" tag of the XML document. The "Tip" feature also provides for the "modification" of input factor mentioned in the previous chapter.

In order for the learner to have opportunities to produce target language, an answer text box was provided and the entire concept of worksheet with questionnaires was applied. This output from the learners is also stored for retrieval by the instructor. This design feature was included to allow learners to produce comprehensible output and be able to notice and correct their errors and at the same time receiving feedback from their instructors.

The availability of different worksheets that can be loaded by the learner provides the learner with a sense of control over his linguistic efforts and thus leads to greater satisfaction and hence better performance. The User Interface of Tutor was made in accordance to the standard look of the rest of the Kirrkirr. Small but important features like the following were included which were in accordance to numerous Design research suggestions and aimed to enhance the learning experience:

- Going to the next question on pressing enter

- Highlighting/Selection of the entire answer when the cursor is brought back to the answer text field.

- Disabling of forward and backward buttons when the learner reaches the end and beginning of the worksheet respectively.

- Disabling of the other worksheet buttons when any worksheet is loaded.

- Graying out of the buttons instead of removal, causes less confusion

It was also taken care to include the "error correction" feature in Tutor. According to factor (e) in the previous chapter, it is vital to let the learner correct his/her linguistic output. The Tutor does precisely that by allowing the learners to navigate between the questions of the worksheet and re-enter answers that they feel might have been wrong.
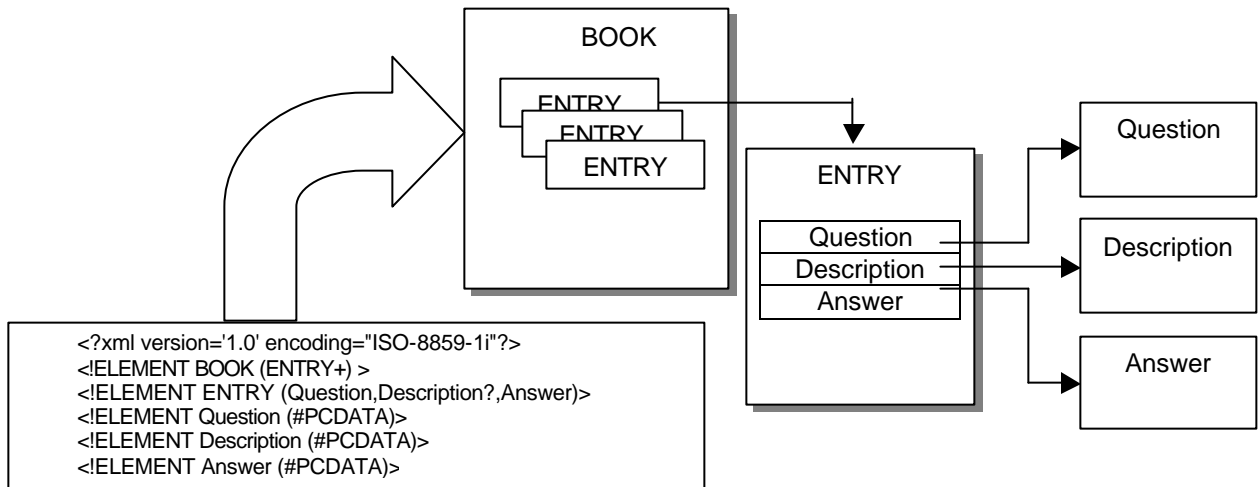
```
<?xml version='1.0' encoding="ISO-8859-1i"?>
<!ELEMENT BOOK (ENTRY+) >
<!ELEMENT ENTRY (Question,Description?,Answer)>
<!ELEMENT Question (#PCDATA)>
<!ELEMENT Description (#PCDATA)>
<!ELEMENT Answer (#PCDATA)>
```

Figure 5: DTD outline

3.3   Implementation

While it is important to come up with a suitable design for an application, its implementation requires a completely new set of skills. To recognize the technologies that are most suitable for a specific purpose and to be able to reform the design of the application according to the constraints of speed, efficiency and memory is a totally different ball game. This section talks about how Tutor was programmed and technical details of its working. This is dealt with in two sections, the first one talking about how the Tutor obtains its content and the second about how it deals with this content.

a. The Content

Tutor draws its content from XML files referred to as Worksheets that have been carefully designed by a teacher. The answers entered by the learner are also converted into a DOM object and stored as an XML file. There were a number of reasons for selecting XML. A few of the most prominent are:

- Plain Text - Since XML is not a binary format, one can create and edit files with anything from a standard text editor to a visual development environment. That makes it easy and useful for storing small amounts of data. At the other end of the spectrum, an XML front end to a database makes it possible to efficiently store large amounts of XML data as well. So XML provides scalability for anything from small configuration files to a large-scale teaching solution.

- Data Identification - XML tells you what kind of data you have, not how to display it. Because the markup tags identify the information and break up the data into parts, an email program can process it, a search program can look for messages sent to particular people, and an address book can extract the address information from the rest of the message. In short, because the different parts of the information have been identified, they can be used in different ways by different applications.

- Easily Processed - As mentioned earlier, regular and consistent notation makes it easier to build a program to process XML data. For example, in HTML a `<dt>` tag can be delimited by `</dt>`, another `<dt>`, `<dd>`, or `</dl>`. That makes for some difficult programming. But in XML, the `<dt>` tag must always have a `</dt>` terminator, or else it will be defined as a `<dt/>` tag. This restriction is a critical part of the constraints that make an XML document well-formed (Otherwise, the XML parser won't be able to read the data.) And since XML is a vendor-neutral standard, and one can choose among several XML parsers, any one of which takes the work out of processing XML data.
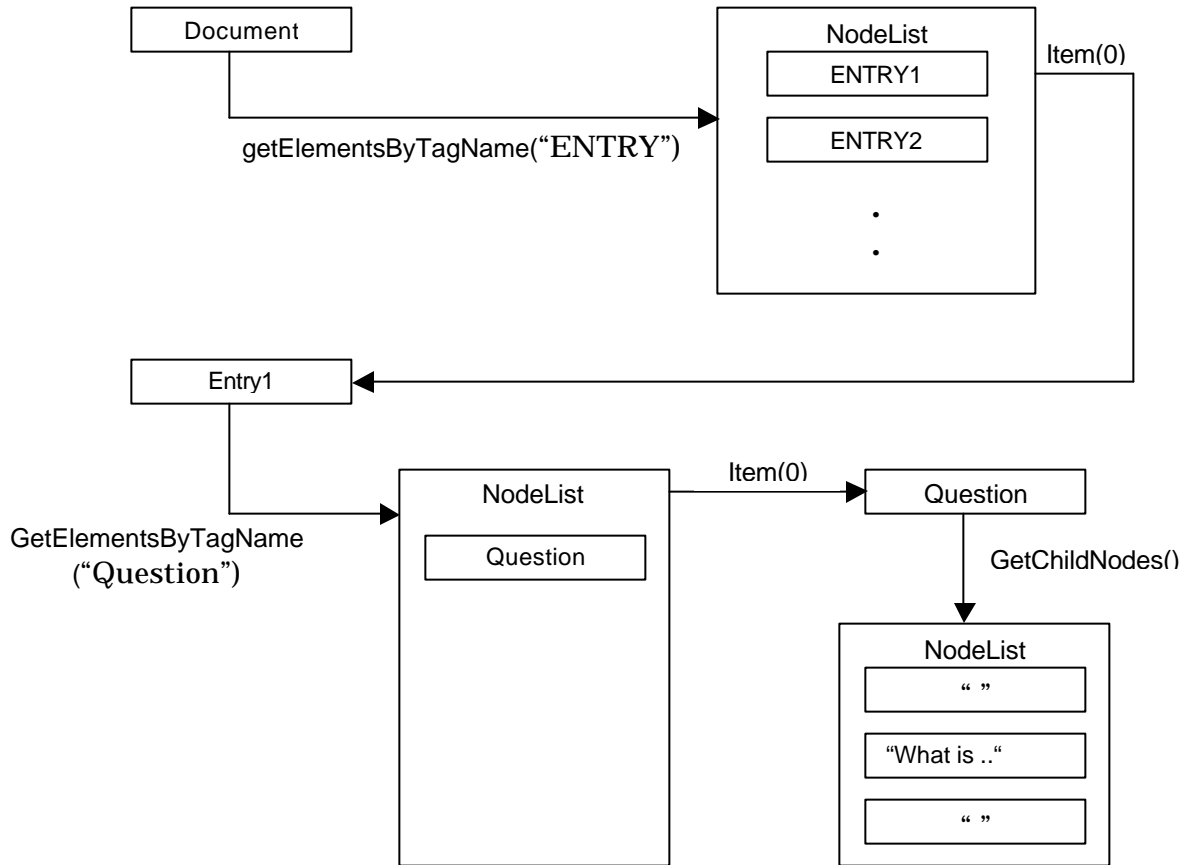
Figure 6: Information Extraction from a DOM object

▪ <u>Hierarchical</u> - Finally, XML documents benefit from their hierarchical structure. Hierarchical document structures are, in general, faster to access because one can drill down to the part one needs, like stepping through a table of contents. They are also easier to rearrange, because each piece is delimited.

b. The Program

The language acquisition process described earlier is operationalized in the Tutor program. The words to be acquired are presented to the learner in contexts selected in such a way as to ensure an efficient and, as it were, condensed acquisition process.

Java was used to code the Tutor, since the parent application, Kirrkirr, was thus implemented. Tutor needs to have a hierarchical object model for its XML worksheets and answer sheets hence it was considered appropriate to use Document Object Model (DOM). DOM is a set of platform and language neutral interfaces that allow programs to access and modify the content and structure of XML documents. It creates a tree of Nodes (based on the structure and information in the XML document) and provides access to the information by interacting with this tree of Nodes.

The information from the XML worksheets is converted into a DOM object, using a validating Java XML parser, and is then used to obtain information. The XML worksheet files follow a specified Data Type Definition (DTD), which is outlined in Figure 5. This DTD is used by the validating Parser to parse the XML file. Similarly, the answers typed in by the learner form another DOM object and are written to a file, whose name is specified by the learner.

The DOM object was instantiated using Sun's parser. First, an input source of the selected worksheet file was created. An XML Document builder object was then passed the input source as well as the Parser object and after parsing, the Document extracted. This Document contains a tree of Nodes and hence is simply a container of Nodes.

To extract information from the Document object, the Element (which are also Nodes) objects that have the tag name "ENTRY" were obtained from it using a method called *getElementsByTagName(String tagname)* which returns a NodeList of all the Elements with that tag name. The method *item(int index)* in NodeList object was used to obtain one ENTRY object. Once this reference to an ENTRY element was obtained, the Question, Description and Answer information from this ENTRY element were extracted. The NodeList object was thus iterated through to obtain information from each ENTRY Element (Node). Figure 6 presents a visual representation of the code used to get information out of the Document:

Similarly, a DOM object model is created when the learner loads a worksheet. As the learner enters every answer, "QuestionNo", "SuggestedAnswer", and "GivenAnswer" Nodes are created and appended to the DOM object. When the learner unloads the worksheet, this DOM object is printed out to a learner-specified file so that the teacher can assess and evaluate his/her performance.

Chapter - 4 The QuizMaster

Presenting the learner with questions is the only way to ensure that learning is occurring. The questions provide the instructor with feedback about how a student, or a group of students, is performing, and hence with the information about the effectiveness of the lesson itself (Steinberg, 1991). One of the most important characteristics of an advanced learning environment is its ability to evaluate the knowledge acquisition and retention rate of its users. Unfortunately, Kirrkirr was designed for browsing and information retrieval, and it lacked a facility for easy evaluation of user knowledge that an advanced educational environment required. This was the prime motivation for the development of QuizMaster, the testing tool. This section discusses some basic design issues that the author had to consider in the development of QuizMaster, and in that light leads a walk-through its design and implementation.

Tests that are administered at computer terminals, or on personal computers, are called computer-based tests (CBTs). Receptive-response items including multiple-choice, true false, and matching items are fairly easy to adapt to the computer-based testing medium (Brown, 1997). But, why should we bother to create computer-based language tests at all? Aren't they really just a sophisticated version of the paper-and-pencil tests that they will probably be modeled on? Brown (1992b) summed up and subdivided the advantages of using computers in language testing into two categories, testing considerations and human considerations.

Among the *testing considerations*, the following are some of the advantages of using computers in language testing:

- Computers are much more accurate at scoring selected-response tests than human beings are.
- Computers are more accurate at reporting scores.
- Computers can give immediate feedback in the form of a report of test scores, complete with a printout of basic testing statistics.
- The use of different tests for each student should minimize any practice effects, studying for the test, and cheating (Drasgow, Levine, & McLaughlin, 1987).
- Diagnostic feedback can be provided very quickly to each student on those items answered incorrectly if that is the purpose of the test. Such feedback can even be fairly descriptive if artificial intelligence is used (Baker, 1989, pp. 423-425, or Bunderson, Inouye, & Olsen, 1989, pp. 398-402).

Among the *human considerations,* the following are some advantages of using computers in language testing:

- The use of computers allows students to work at their own pace.
- Students may find that Computer based testing tools are less overwhelming (as compared to equivalent paper-and-pencil tests) because the questions are presented one at a time on the screen rather than in an intimidating test booklet with hundreds of test items.
- Many students like computers and even enjoy the testing process (Stevenson & Gross, 1991).

No doubt, cheating will arise, but such problems can be surmounted if a little thought and planning are used. In fact, computerized testing is increasingly being viewed as a practical alternative to paper-and-pencil testing (Kingsbury & Houser, 1993). Given the advantages of individual, time-independent language testing, computer-based testing will no doubt prove to be a positive development in assessment practice (Brown, 1997, p. 46).
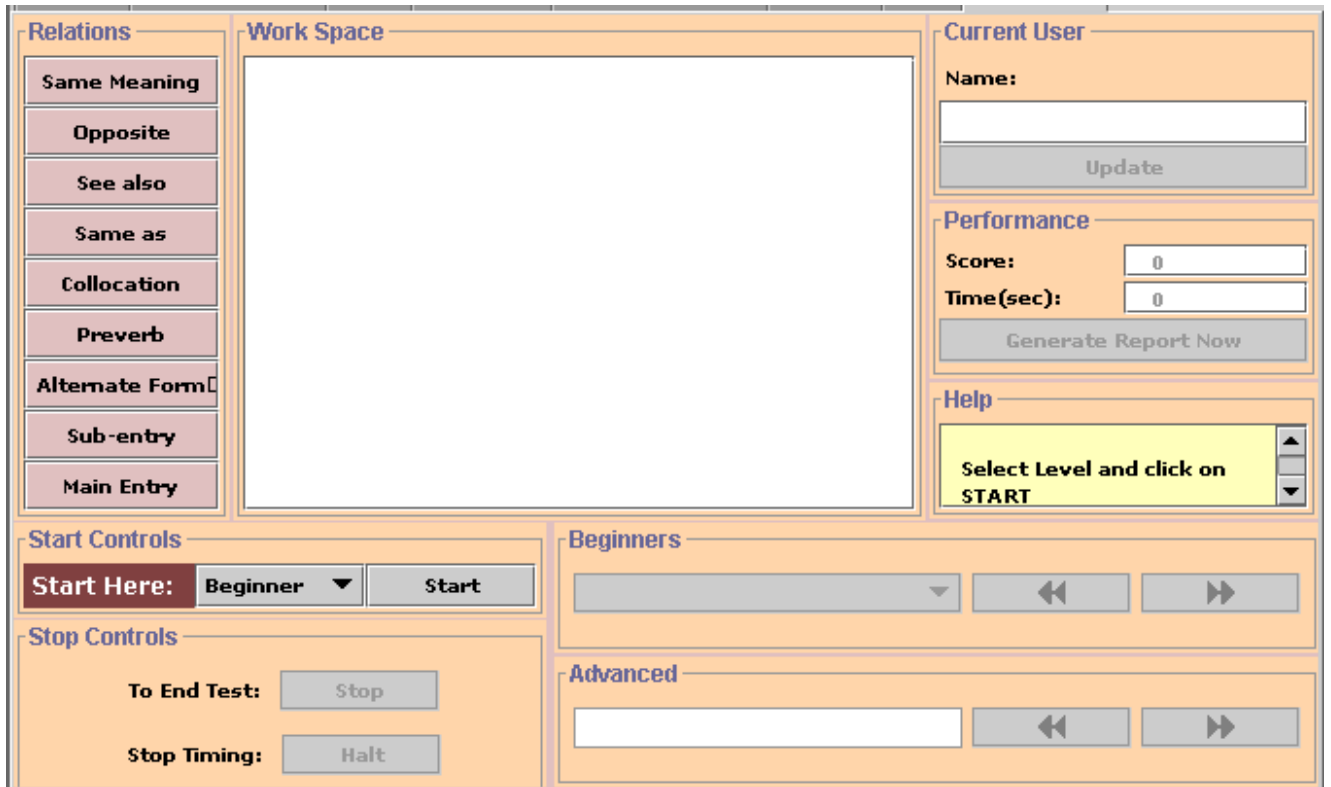
Relations

Same Meaning
Opposite
See also
Same as
Collocation
Preverb
Alternate Form
Sub-entry
Main Entry

Work Space

Current User

Name:

Update

Performance

Score: 0
Time(sec): 0

Generate Report Now

Help

Select Level and click on START

Start Controls

Start Here: Beginner ▼ Start

Stop Controls

To End Test: Stop

Stop Timing: Halt

Beginners

◄◄ ►►

Advanced

◄◄ ►►

Figure 7: The QuizMaster

4.1 Design Considerations

A number of test design questions need to be addressed by language testers before they can be fairly sure they are using a test responsibly (Brown, 1997). Hence, considerable thought was required to be put into the design of QuizMaster, as results generated by it would be held as a means for a learner to gauge his/her Warlpiri skills. This section examines a number of basic assessments, psychometric, technical, and administrative issues that needed to be taken into account when QuizMaster was designed. Some of the more important questions that need to be considered are following:

a. Should the test be standard length or vary across students?

An ideal testing tool should be capable of being tailored to each student's ability levels. The issue that arises then is whether the test should be the same length for each student or of different lengths tailored to each student's ability. Stocking (1987) indicates that the tailored strategy resulting in different lengths may produce biased scores for students who end up taking short versions. If the test is made to be relatively long for all students, even for those students who will be taking the "short" versions, no such biases may surface. However, if the test is short for all students, it may be preferable to use the same length test for all students.

b. How should the test items be sampled?

Traditional tests, in order to be theoretically valid, are often designed on the basis of clearly developed test specifications to sample different knowledge or skills that make up a particular domain. To develop questions or testing items that can adequately evaluate this knowledge or these skills by imbibing the idea from the test specifications requires careful design in to order to prevent duplication of effort.
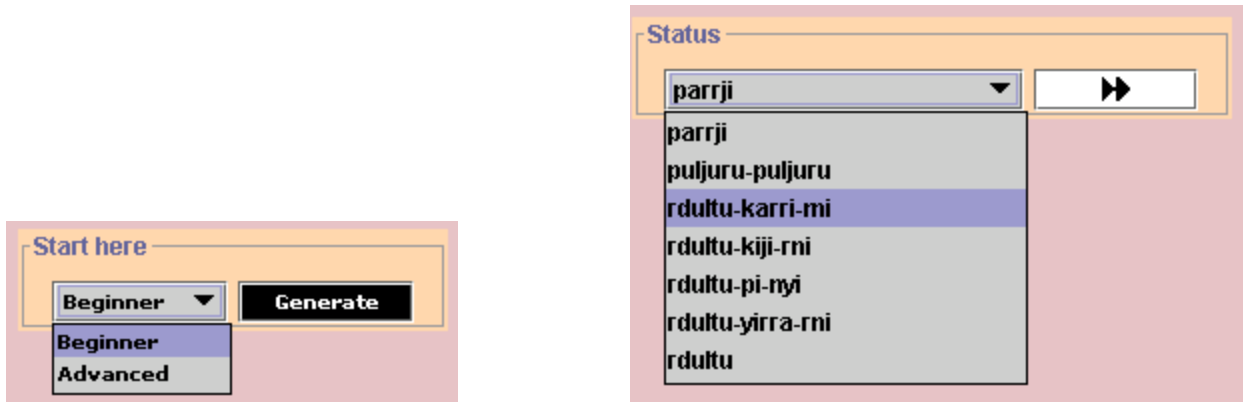
Figure 8: The two levels in QuizMaster (left) and choices of answers for Beginners level (right)

For instance, one strategy might be to develop a sampling algorithm that takes into account the issue of specification sampling by keeping track of which specifications have been sampled for a given student and selecting the remaining items in a way that best fulfills the rest of the test specifications. Such a scheme might even take into account the relative importance of the various specifications.

c. How should the scoring be done?

Scoring is another vital aspect of testing. A number of alternative methods have been explored in the educational literature including at least the following:

- Raw scores (a simple sum of the number of correct items)
- Weighted raw scores (in which some items count more than others)
- Scaled scores (e.g., the TOEFL, which is equated across forms; for an explanation of scaling and equating CATS, see Dorans, 1990; for studies using the techniques, see Mazzeo, Druesne, Raffeld, Checketts, & Muhlstein, 1991; O'Neill, Folk, & Li, 1993; Schaeffer, Steffen, & Golub-Smith, 1993)
- Any of the above (1-3) corrected for guessing (as discussed for CATs in Stocking, 1994)
- Any of the above (1-4) scores referenced to a conventional test (as described for CATs in Ward, 1988; Wainer et al., 1990), especially when a CAT is first introduced to replace an existing paper-and-pencil test

So far, there is insufficient research on which method works best for which purposes when interpreting the test scores. Hence, the author selected the method most suitable and easy-to-understand by Kirrkirr users.

d. How should CBT item omissions be dealt with?

On traditional tests, especially those scored with a correction for guessing, students can and do omit items if they are not able to answer them. However, on a CBT, problems arise if the students omit one or more items. How should testers deal with such omissions? Wainer et al. (1990, pp. 236-237) chose to not allow for omissions on their CBTs; the students simply could not go on to the next item until they had answered the one in front of them. Another strategy would be to allow omissions but assume that omitted items are wrong. Other problems may arise if students are allowed to omit items. Students can simply omit all items and get a good look at all of the items in the item bank. This would have important implications for item exposure.
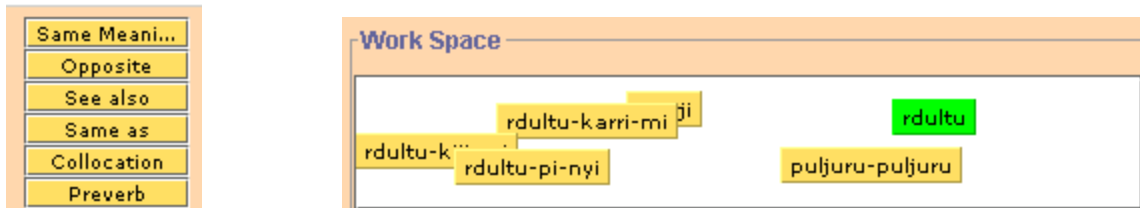
Figure 9: Relations tested (left); Words displayed with main word in green

e. How should decisions about cut-points on CBTs be made?

The literature on deciding cut-points is often referred to as standards setting (Jaeger, 1989, pp. 491-500). Reckase (1983) has suggested the use of the sequential probability ratio test (SPRT) for making pass-fail decisions in tests. Lewis and Sheehan (1990) suggested using Bayesian decision theory for making mastery/non-mastery decisions on a computerized sequential mastery test. The SPRT and other methods should all be considered in terms of how well they help with test construction and decision-making on CBTs.

f. Should item review opportunities be provided on a CAT?

In traditional paper-and-pencil testing situations, students who have time remaining at the end of the test can go back and review earlier items. In addition to the important logistical issues raised by Green (1990), which included system, hardware, human, and software issues, another important CBT logistical issue that needs to be addressed by language testers is whether to provide item review opportunities. Should this feature of traditional testing situations be mimicked in CBTs? This is an important consideration because this enhances the effectiveness of the test as compared to introducing logistical problems or large memory and computing resources required by the testing tool.

Although all the questions considered above can affect in positive and negative ways the effectiveness, appropriacy, and equitableness of the second language CBT tool, the author does not purport to highlight all the major issues related to CBT design or evaluation. Rather, she poses a number of specific questions that developers need to address, and CBT score users and examinees need to understand and carefully consider, when CBT becomes a heuristic of assessment. The answers to the questions posed are neither simple nor singular, but solutions must nevertheless be found if examinees and score users are to benefit from this inventive, efficient approach to second language assessment.

4.2   Design

The QuizMaster tests the vocabulary of its users and its contents are based on a systematic inventory of the vocabulary relevant for the target group.

Unlike a majority of other CBT tools, the QuizMaster does not ask contextual questions but works by presenting a word to the users and then expecting them to correctly input all the words related to it. This is because it was found that during the initial stages of instructed SLA, students learn a couple thousand, mainly high frequency words. Functional language proficiency, however, requires mastery of a considerably larger number of words. It is therefore necessary at the intermediate and advanced stages of language acquisition to learn a large vocabulary in a short period of time. There is not enough time to copy the natural (largely incidental) L1 word acquisition process. Incidental acquisition of the words is only possible up to a point, because, on account of their low frequency, they do not occur often enough in the second language learning material. Acquisition of new words from authentic second language reading texts by means of strategies such as contextual deduction is also not a solution for a number of reasons (Groot, 2000). There appeared to be no alternative to intentional learning of a great many new Warlpiri words in a relatively short period of time - hence the basic design concept of QuizMaster.

To cater to learners at different skill levels, they were divided into two major categories: Beginner and Advanced. So in QuizMaster, each examinee takes a unique test that is tailored to his or her ability level, which is broadly classified into the above-mentioned categories. Avoided are questions that have low information value about the user's proficiency. The result of this approach is higher precision across a wider range of ability levels (Carlson, 1994, p. 218). In fact, CBT was developed to eliminate the time-consuming and inefficient (and traditional) tests that presented easy questions to high-ability persons and excessively difficult questions to low-ability testees.

Clearly, there are various levels or dimensions to word knowledge (Nagy & Herman, 1987). Knowing a word may be seen in operational terms as a continuum ranging from vague recognition of its spelling (semantically, syntactically, stylistically) to correct and contextually appropriate productive use. Retrieval of a word from the mental lexicon for productive use requires a higher degree of accessibility or, in other words, a more solid integration in various networks than is needed for receptive use. For measuring this higher level of mastery, a test which asks testees to simply recognize a word and give its meaning is unsuitable; a test using the cloze technique, which measures testees' ability to produce the word themselves, is much more valid for that purpose. For a meaningful interpretation of the data, it is essential to give an accurate description of what one understands by the trait "knowing a word" and of what trait is intended to be measured by the testing method. Keeping this in mind, the "Beginner" level of QuizMaster required testees to select the correct word from a list of given options (one of which was definitely the correct answer) while the "Advanced" level of QuizMaster asked the testees to type out their answers in the answer text-box.

For both the levels, first a word is randomly selected from the Warlpiri dictionary. The testees are required to input words that they think have the following relation to one main word:

- "Same Meaning" (Synonym)
- "Opposite" (Antonym)
- "See-also" (any relation)
- Same as
- Collocation
- "Preverb"
- "Alternate Forms"
- "Subentry"
- Main Entry

These answers are then stored and processed to generate, at the end of the test, the suggested-versus-received-answer-file and to deliver a score to the testee. The file containing the answers entered by the user, called the report sheet, is created at the end to facilitate changes made to answers by the testees. The report sheet has personal information of the testee such as his name, age, and score and time taken depending on his/her preferences. This was included to personalize the QuizMaster. The score is also displayed in the end so as to not divert the attention of the testee from lexical retrieval and to reduce stress introduced by every performance evaluation. The score involves a positive unit point for a correct answer and zero points for a wrong answer. This method of marking was selected, as it was easy to understand and comprehend. The testees were thus marked out of a full mark of 9 points.

Careful UI considerations were involved in the design of QuizMaster. The layout of the buttons was the most crucial so that the learners don't get confused and/or intimidated at the sight of QuizMaster. Basic node-movement had to be included to add the interest factor and color schemes utilized to the maximum. An XML file was generated in the end so that the learner can have a record of his performance and also so that this file can be put to any use in the future that the future developers of Kirrkirr might perceive useful.
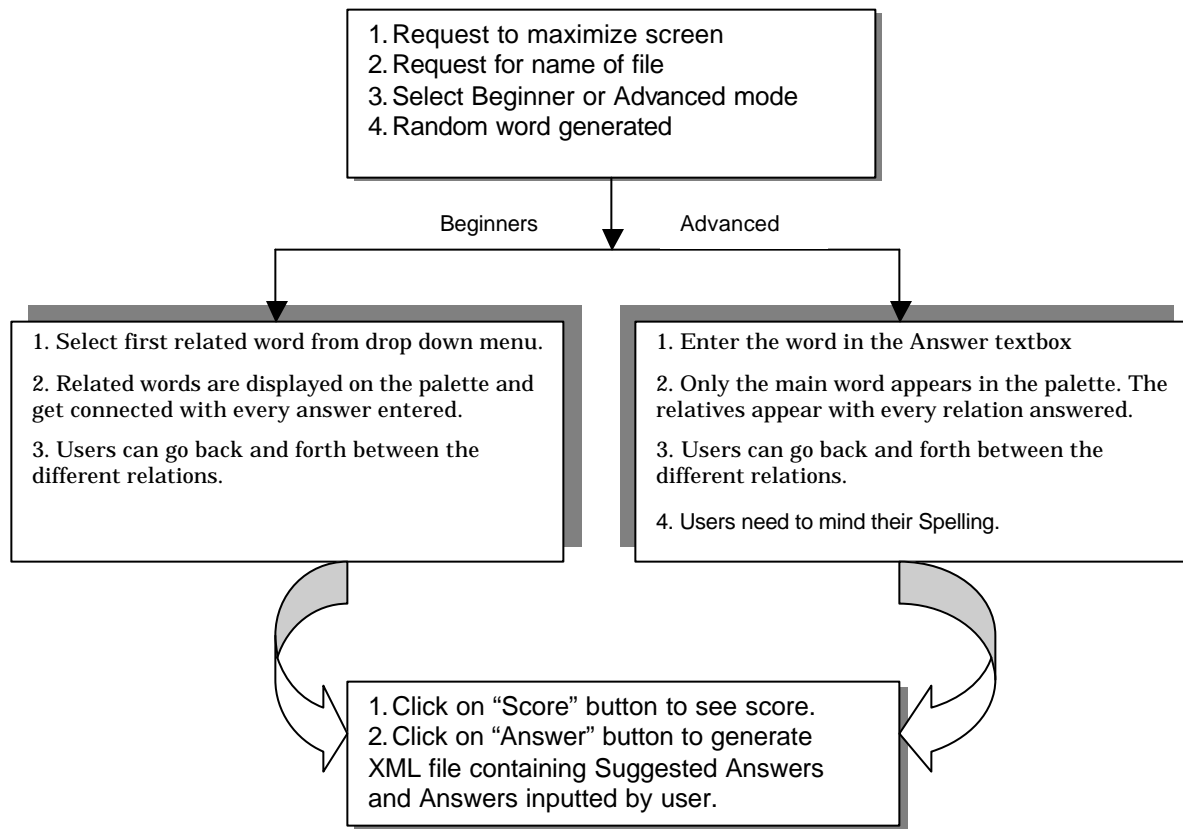
Figure 10: Flow of the testing methodology in QuizMaster

4.3   Implementation

The implementation of the QuizMaster involved major User Interface design considerations even while the "core-machine" was simpler to program than one would expect because of the advantages of Object Oriented Programming advantages presented by Java. The limitations that were needed to be wary of while implementing the Graphical User Interface were:

- Screen capacity - While most computers today have overcome the 80 characters by 25 lines restrictions of a few years ago, the amount of material that can be presented on a computer screen is still limited.

- Graphics capabilities of various computers (especially older ones) may be limited, and even those machines that do have graphics may be slow (especially the cheaper machines). Thus, QuizMaster implementation ruled out add-ons involving even basic graphs or animation as they may not be feasible at the moment in many language-teaching situations.

Once the design of the Test was figured out it was simple to implement as most of the functions involved the use of existing methods in other classes. Figure 10 summarizes the flow of the processes in the QuizMaster. The important function unique to QuizMaster was the random word generator. Random word generation was achieved by generating a random number using the Random class.

"MainWord = (String)parent.Headwords.get(myRandom.NextInt(parent.Headwords. ccapacity()));"

This command returns a string having a rank same as the generated random number in the words list of the Kirrkirr dictionary.

The rest of the programming primarily involved the successful integration and communication between methods of different classes to produce some utility for QuizMaster. After a main word is generated, all its related words are obtained from the information in the XML dictionary and stored as a vector. As the user inputs the answers, either by selecting from the drop down menu in case of beginners or by entering the words in the text box in case of advanced users, these answers are added on to another vector. In case the user wants to change any of his/her answers, he/she can do so and the changes are easily updated in the vector by altering the corresponding element. At the end of the test, when the user clicks on the "Answer" button, the two vectors' elements are used to create a DOM object that is printed out to a file. This generation of the XML file containing the suggested answers and the answers inputted by the users, involved once again, the method detailed in the previous section (refer to section 3.3b).

The implementation also involved extensive "reform" to the existing classes namely:

- FunPanel – the class that deals with the response and the display of the word nodes as illustrated in Figure 9.

- GraphPanel – the class that deals with the overall working of the funpanel. It holds the Funpanel and the FunOptionsPanel objects. The FunOptionsPanel deals with the settings that are used by the FunPanel.

- DictionaryCache – the class that deals with the information containing the words taken from the dictionary. It calls a parser and extracts this information.

- Dictfield – the class that holds the information about every word

- WordNode – the class that creates each of the word node that is displayed in the QuizMaster Panel.

These changes or reforms took advantage of the method overriding and the method overloading attributes of Java

Chapter - 5 Auxiliary Tools

In further attempts to move Kirrkirr towards being more effective CALL software, additions to it in the form of User Profiles and HTML file generation were made. While User Profiles make Kirrkirr more responsive to various learner preferences and needs, HTML generation aimed at isolating individual words or group of words that could be paid special attention to by the learner.

HTML generation includes an option whereby the learner/user, after having browsed through a few words, can generate an HTML file containing the words he/she has browsed. The learner can then focus on the acquisition of these words either by revising through the meanings (etc) contained in the file or even obtaining printouts.

User profiles create and maintain the settings for each user's work environment on the local computer where Kirrkirr is installed. A user profile is created for a user, if he/she desires, so that the next time the user loads his/her profile, Kirrkirr works in accordance to the user's personalized settings. User profiles provide several advantages to users that will be detailed later on in the chapter. Types of user profiles include:

- **Local user profile** - A local user profile is created the first time the user log on to a computer and is stored on a computer's local hard disk. Any changes made to the local user profile are specific to the computer on which the changes are made.

- **Roaming user profile** - A roaming user profile is created by the system administrator and is stored on a server. This profile is available every time the user logs on to any computer on the network. Changes made to the roaming user profile are updated on the server.

- **Mandatory user profile** - A mandatory user profile is a roaming profile that can be used to specify particular settings for individuals or an entire group of users. Only system administrators can make changes to mandatory user profiles.

The Kirrkirr application was equipped with the Local user profile and this section talks about the work that was involved in implementing the User Profiles and HTML generation and the objectives behind integrating them with Kirrkirr.

5.1   User Profiles

The Kirrkirr user profile defines customized overall environments, which include individual word display settings, network display settings and history settings. For an experienced learner, the provision of Profiles makes it possible to maximize his/her learning efficiency by having information presented in the manner that the learner is most ready to accept or prefers to see. This customization is very important to give the learner a sense of control over his/her learning pattern and hence a greater degree of satisfaction which can maintain the general enthusiasm in learning Warlpiri. The other benefits that profiles provide to users are as follows:

- Multiple users can use Kirrkirr, and each receives personalized settings when the corresponding profile is loaded. This helps decrease the cost of managing numerous computers
- When users load the profiles on Kirrkirr, they receive the settings, as they existed when they last logged off.
- Customization of Kirrkirr made by one user does not affect another user's settings.
- A default user profile that is appropriate for the user's tasks can be created.

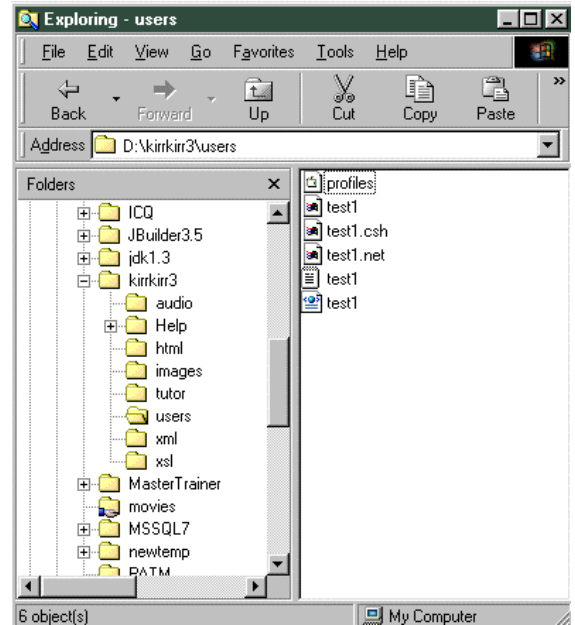| File extension	Contents
DAT	Index file, XML file, XSL filename and the state of each of the panels
TXT	History of the users activity
NET	Preferred network options.
XML	Notes Hash table
CSH	Cache

Figure 11: Files created for Profiles

a. Design

There exists a default profile that is loaded when Kirrkirr is first started. If a user wishes, his/her profile can be made the default. He/she can also turn on or off the option of automatically loading the default profile when Kirrkirr loads. This way, the user can be greeted by his preferred settings or choose to see a 'neutral' profile first.

Every profile keeps track of the variations in the settings of the following:

- XSL file – Kirrkirr uses an XSL file to display the meaning of every word in the Formatted Entry Panel. Depending on the style specified by the XSL file, the display of the content of the selected words varies. The name of the XSL file once specified by the user in the Options -> Preferences-> XSL Choice Panel is then saved in the profile and loaded when the profile is loaded. This way, the display style of the word can be changed according to the user preference, from for example "Just Meanings" for beginners to "All Details" in case of advanced users.

- Network Options – variations in the network options change the look of the words in the Network Panel. This includes the following:
  o Repulsion between words
  o Spring
  o Animation Speed
  o Edge width
  o Maximum Focus Nodes
  o Choosing what links to see between words
  o Edge Lengths
  o Random Movement
  o Display of Legend

- History – As the user browses through the words, every word he/she clicks on is added to a list maintained by Kirrkirr. The History of the words that the user had visited the last time he/she was using Kirrkirr is saved. This history is loaded when the user loads his profile so that the user can keep track of the words that he/she has already browsed through.
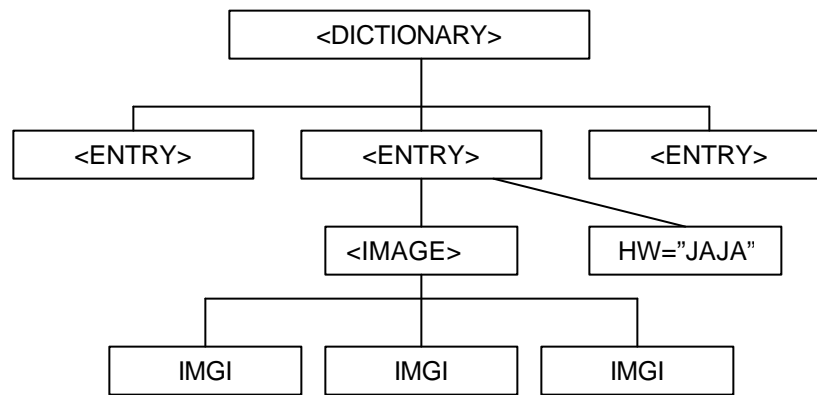
```
                    ┌─────────────────┐
                    │   <DICTIONARY>  │
                    └────────┬────────┘
          ┌──────────────────┼──────────────────┐
    ┌───────────┐      ┌───────────┐      ┌───────────┐
    │  <ENTRY>  │      │  <ENTRY>  │      │  <ENTRY>  │
    └───────────┘      └─────┬─────┘      └───────────┘
                    ┌────────┴──────────────┐
               ┌─────────┐           ┌──────────────┐
               │ <IMAGE> │           │  HW="JAJA"   │
               └────┬────┘           └──────────────┘
        ┌───────────┼───────────┐
   ┌─────────┐ ┌─────────┐ ┌─────────┐
   │  IMGI   │ │  IMGI   │ │  IMGI   │
   └─────────┘ └─────────┘ └─────────┘
```

Figure 12: Structure of the Kirrkirr Dictionary

b. Implementation

The Object-Oriented nature of Java was ideal for this feature and completely facilitated its successful implementation. An object of a single class ProfileManager handles all the profiles of Kirrkirr and defines the methods for loading, saving and specifying of profiles. It acts as a gateway which calls the objects of the Network Panel, Graph Panel, Notes Panel, History class, Formatted Entry Panel to save their contents in files specified by it for saving a profile or load their contents from files specified by it. It then calls their respective methods to update their settings. Thus the Profile Manager simplifies an otherwise tedious task.

All the User Profiles information is stored in the users folder of the Kirrkirr Home directory. This folder also contains a file called "Profiles.dat" where the list of the profiles maintained by the application is saved. With every new profile created, the name of the profile is added to the already existent list in "Profiles.dat".

When the profile is new, storing all the desired options in different files with the same name as the profile but different extensions primarily involves the saving of that profile. These files are stored in the Users folder of the home directory of Kirrkirr. The files created are summarized in Figure 11. For instance, for a profile named test1, the files created and stored in the users folder will be test1.dat, test1.txt, test1.net, test1.xml and test1.csh. (Refer to Figure 11).

Similarly, for loading any profile, these files are serially opened and the data from these files is used to update the settings of Kirrkirr. As the data obtained from the different panels is of different types, either String or Objects, it was practical to store it in separate files. This also facilitates easier enhancements to the Profiles feature by mimicking the object-oriented structure. Thus, data for different panels can simply be read from different files instead of having to deal with markers in a file for where the input for one panel ends and input for another panel begins. Every panel's class, for which the settings were stored, namely the Network, Formatted Entry and Notes Panels, was modified to be able to incorporate settings that were fed in. This was tedious, however much simplified due to the Object Oriented nature of Java.

5.2   HTML Generation

This feature provides the facility such that once the user has selected a few words, an HTML file with the definitions for all those words (using user chosen XSL to translate to HTML) can be generated. The advantages of this feature are

- The user can obtain printouts of the definitions of the words that he has learnt.

- The HTML file obtained can be used as a reference for second language acquisition even while away from the computer
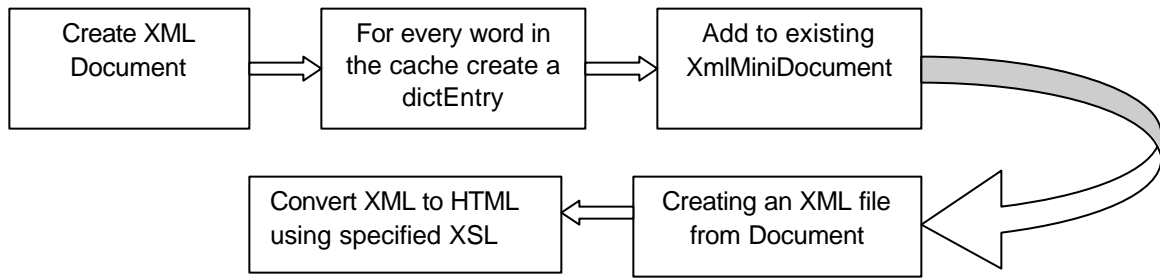
Figure 13: Process Flow for HTML Generation

- ▪ The HTML file can be used as a track record of the words the user has gotten familiar with. It can thus be a performance meter in the language acquisition process for every learner.

a. Design and Implementation

The implementation of this add-on was simplified to a certain extent by the existing structure of the Warlpiri Dictionary and the way in which Kirrkirr works. To suitably comprehend the implementation it is important to understand the functional and structural behavior of Kirrkirr. As indicated in Figure 12, every word in the dictionary is stored in the form of elements named "ENTRY".

As a word is clicked upon, the cache object obtains the information about the word from the dictionary ENTRY and creates a dictEntry object (an object to hold the definitions of a word). Using this dictEntry object the cache object then creates a XML Document object. This object now contains all the information about the specified word that is present in the dictionary element. By specifying different XSL files, the manner and extent to which this information is displayed can be changed. Thus, if the "Just Meanings" XSL file is chosen, then simply the meaning of the word is extracted from the Document object and displayed and if "ALL details" XSL file is selected, all the information present in the Document object is extracted and displayed to the user.

When the user clicks on the File -> Create HTML menu option, ONE HTML file with the learner specified name is created in the html folder of the Kirrkirr Home directory that contains the definitions of all the words that the user has browsed over so far and hence are saved in the current profile cache. The procedure that was followed in implementing this feature was as indicated in Figure 11.

For every word stored in the cache at the time of Generation of HTML command initiation, a dictEntry object is created. For the first word in the cache, an XML Document object is created. Then for every subsequent word in the cache, a corresponding dictEntry object is created. This dictEntry object is used to create ENTRY elements from the information from the dictionary and then added to the Document object by the method *addElement(),* detailed in Figure 14.

```
private void addElement(dictEntry de) {
try {
setup_bis(de);  // obtains information from the dictionary into the dictField
doc= myxmlMD.addElement(parent.xml_file, bis,doc); //add Element to the Document
finishup_bis(); // closes the dictionary file
} catch (Exception e) {..}
}
```

Figure 14: Code for Creation and Addition of an Element

This Document object is then written to a user-specified file by calling the Utility method, *Helper.writeDocToFile(doc, myXmlFile)* and the XML file then converted to HTML file using the XSL file specified by the learner in the Preferences using the method *XslDriver.makeHtml(myXmlFile, this.xslfile, myHtmlFile).* A sample of a generated HTML file is attached in the Appendix A4.

Chapter - 6 Evaluation: A Critical Part of Design

Evaluation of pedagogical materials refers to research intended to discover how successful the materials are for learners. For every tool developed with a specific purpose, it is meaningless to not analyze how far it succeeds in satisfying its purpose. Evaluation of performance is an inseparable part of development as it gives the developer feedback about the product and whether his aims and objectives of creation were fulfilled or not.

SLA research provides some clear guidance for the evaluation of instructional activities. For many years SLA researchers have observed that the process of classroom interaction reveals essential information about the quality of instruction (Long, 1980). As a consequence, analysis of L2 classroom discourse has become a mainstay in research into the efficiency of the instructional activities (Allwright, 1988; Allwright & Bailey, 1991; Chaudron, 1988; Day, 1986; Johnson, 1995; Seliger & Long, 1983; van Lier, 1988).

6.1   Research Methods for Evaluation

There are different methods of analysis of second language classroom discourse. All these methods for evaluation include process-oriented analyses of learners working on second language tasks. This can be done in two ways, one empirical and one purely theoretical.

One way of investigating task completion processes is to record learners' language during task completion. This empirical method involves, feedback from the learners and observation of their acquired language skills.

The second form of process-oriented analysis relies on introspective methods to gain evidence about the learners' language. In this method, SLA research facilitates the evaluation of instructional activities. Using the criteria researched earlier in the report, a checklist of desired task features can be developed. This checklist could then be used to compare with the features of the instructional software and the number if matches used to determine its effectiveness. This method assumes that the software will be used to its best capacity i.e. facilitation implies utility. In other words, it is assumed that the user is willing and takes advantage of every feature provided by the software and maximizes his learning experience, thus availability of more features adds to the usability of the software.

In this thesis, the second method was adopted to "measure" the effectiveness of Kirrkirr. This section talks in length about how such a methodology was used to reach a fair conclusion.

6.2   Questions for Evaluation of CALL

The types of process-oriented research methods used in classroom research can yield large quantities of data because they attempt to document the course of instructional process over time. It is therefore useful to be able to focus description and evaluation on specific design features. The design features to be short-listed should be the ones that play a role in SLA. Chapter 2 dealt with the process of SLA and narrowed down the suggested criteria for development of effective CALL solutions. Chapelle (1998) applied the design criteria to develop the following research questions, thus illustrating how evaluation procedures can be developed to parallel principles of design in CALL.

- Is there evidence the salient linguistic characteristics of the target language input have been attended to?
- Do learners get to see the modifications of linguistic input?
- Is "comprehensible output" from learners encouraged?
- Are there opportunities for learners to notice errors in their output?
- Can learners correct their linguistic output?
- To what extent do the learners interact with the computer to engage in modified interaction focusing on form and meaning?

- Is the User Interface design pleasing and "friendly"?
- Is Context AND Data provided?
- Is Control provided to the learners?
- Is the Media integrated with the software?

The above-mentioned evaluative research questions were used in this thesis to determine whether Kirrkirr with the additional tools is an efficacious remedy to the acquisition needs of Warlpiri students. Whether or not the instructional software has a positive answer or facilitates a positive answer to the questions mentioned, would determine its inclination to being an efficient tool of learning. The procedure can be equated to one, which involves checking the available features to those proven desired. With a positive 1-point for every "Present" answer and a negative 1-point for every "Absent" answer to the questions above, the total points that the software attains can be calculated and hence its comparative effectiveness evaluated.

6.3 Efficiency of the new Kirrkirr

The tools developed during the course of this project intended to improve the functionality of Kirrkirr as a CALL and CBT tool. It was hoped that the Tutor taught the language, the QuizMaster tested the knowledge gained by the learner, Profiles made Kirrkirr more appealing and easy to use and HTML generation implied enhanced support for lexical acquisition. As such, it would be logically wrong to evaluate them separately and hence the new Kirrkirr - with the additional modules that made it a unified CALL and CBT tool – was subjected to the testing process outlined earlier.

A question was singularly dealt with and used to obtain an approximate idea of the extent to which Kirrkirr with its Tutor and QuizMaster could provide a solution answering it. The results of the evaluation are summarized in Table 1.

	Features	Tutor	QuizMaster
a.	Salient linguistic characteristics attended to	✓	✓
b.	Visible modifications of linguistic input	✓	(-)
c.	Comprehensible output facilitated	✓	✓
d.	Error noticing in output possible	✓	✓
e.	Correction of linguistic output possible	✓	✓
f.	Learner interaction with the computer to engage in modified interaction	✓	(-)
g.	Appealing User Interface design	✓ (?)	✓ (?)
h.	Context AND Data	(-)	(-)
i.	Control provided to the users	✓	✓
j.	Media integrated with the software	(-)	(-)

✓ : Included   (?): Somewhat   (-): Negative or Absent

Figure 15: Summary Table of features

a. Is there evidence the salient linguistic characteristics of the target language input have been attended to?

This question deals with a CALL solution that successfully brings specific, target words to the notice of the learner by highlighting certain Warlpiri input as compared to other target language noise. Kirrkirr provides a positive answer to this question as it entails the following:

- The dictionary highlights every word being displayed in the various panels, so that the users have no doubt about the current word whose meaning they are viewing, or pronunciation they are listening to and are hence about to acquire

- Tutor provides the learners with the "Tip" box which brings to their attention whatever it is that the teacher wants the learners to imbibe

- QuizMaster displays the main word whose relatives the learner is supposed to point out, in green, drawing the learner's attention to the word.

b. Do learners get to see the modifications of linguistic input?

This aspect queries whether the instructional material provides learners with comprehensible input rather than just input. An affirmative would be an answer to this question as far as the new Kirrkirr is concerned.

- The tutor provides comprehensible input in the form of simplification and elaboration. This is done by the "Tip" button, which helps to provide any additional information, or meaning of new words or anything else that the teacher deems fit for the learner's level of knowledge. This is achieved by facilitating changes in the worksheet itself like change of questions and/or change of information to be provided.

- Quizmaster facilitates modification of linguistic input using simplification. The user can see the meanings of the "relational words" on the Help Panel. For instance, a learner who is not so fluent in English can check out the meaning of the word "Collocation" on the Help panel when it is his turn to enter a word that is a "collocation" to the main word. This way the learner is not assumed to be fluent in English.

c. Is "comprehensible output" from learners encouraged?

Kirrkirr very strongly supports comprehensible output so that the learners are pushed to use the target language constructively and to the best of their abilities. It achieves this by the following:

- Tutor entails the learner typing in the answers to questions. These answers are then outputted to an answer file, which is then evaluated by the instructor and feedback given to the learner about his/her current aptitude in Warlpiri.

- Quizmaster provides a scoring system that provides direct indication about the Warlpiri skills of the learner. In both its Beginner and Advanced options, QuizMaster involves the learner pointing out the words that he/she thinks are the relations of the main generated word and the scoring can be used to keep track of the learner's vocabulary.

d. Are there opportunities for learners to notice errors in their output?

The improved Kirrkirr provides a somewhat affirmative answer to this question.

- While using Tutor, the learner enters what he/she thinks are the answers and it involves the instructor printing out the answers. It, however, does not involve the facility whereby the learners can immediately notice errors in what they have entered and correct immediately. The learner does not notice errors until after the worksheet has been completed and the answer file obtained/printed by the teacher.

- The QuizMaster gives feedback about whether the answer is wrong or not by updating its Score. Hence it provides immediate opportunities to notice errors.

e. Can learners correct their linguistic output?

Kirrkirr works towards an answer to this question by relying heavily on the fact that it was designed as a complement to the linguistic instructor and not as the sole tutor of Warlpiri. It uses this to its maximum effectiveness by the following:

- Tutor facilitates detailed error-specific help by relying on the instructor to formulate worksheets that might focus on key areas to which the attention of the learner is to be drawn. The learners can also go through the printed out answer sheets and realize their faults and answer the worksheet questions again to correct their language.

- QuizMaster provides opportunities for learners to correct their linguistic output by letting them navigate to the particular "relation" and select the appropriate word. They can move back and forth between the words for which they intend to change the answers.

f. To what extent do the learners interact with the computer to engage in modified interaction focusing on form and meaning?

Kirrkirr partially supports the feature of modified interaction. While the QuizMaster does not consist of testing with a task goal, it is a completely different scenario with Tutor. Tutor, with its high customizability and variability facilitates "modified interaction". This can be done by the instructor by devising the worksheets is a way that might involve a specific task goal. Thus any teaching strategy that the instructor thinks will make teaching more effective, can be implemented in the Tutor.

g. Is the User Interface design pleasing and "friendly"?

The Graphical User Interface (GUI) is a matter of personal choice. This characteristic makes the question very difficult to answer in absolute terms. While some learners might like a certain stylistic approach and find a specific color combination appealing, others might not. Since this depends from individual to individual, this cannot be answered or used to gauge the effectiveness of the software using this methodology. In such a case, the author is expected to follow a set of guidelines of User Interface Design, a topic that has been extensively researched on. Kirrkirr is designed to satisfy the following general guidelines as proposed by Chua (1999):

- Adopt consistent design in the user interface

- Adhere to one standard user interface design

- Explain the rules of operation of the final application to the intended users

- Cater to environmental controls at both novices and expert levels

- Phrase labels consistently, in a short and concise manner, and or use jargon easily understood by the user

- Attempt to cluster related tasks or objects and not over clutter the interface with every possible combination of objects

- Navigation between screens and within screens is equally important

- Use color appropriately and provide for a secondary indicator if necessary

- Follow the contrast rule – dark text on light backgrounds and vice versa

- Use fonts sparingly and consistently

- Grey-out items, but do not remove them, the purpose with which users could form accurate mental models

- Do not create crowded screens whereby the user might be confused, typically seven items at any one item forms the threshold

- Use borders and whitespace to group logically related items on the screen

- If possible, attempt to provide modal windows that confirm users' intentions, especially for critical actions.

h. Is Context AND Data provided?

Kirrkirr does not provide this feature. While this feature is completely absent from the QuizMaster, the Tutor somewhat salvages the situation by falling back on its configurability feature. Using this, the instructor can impart helpful hints to the learner using the worksheet or the "Tip" button. Here, the Tutor relies heavily on the instructor for analysis such as tracking learner answers or looking for patterns or even trying to answer *why* a particular answer provided by the learner was wrong. They may also offer suggestions for further study or advise going on to a more advanced level or doing some extra work on the current or previous level.

i. Is Control provided to the users?

Kirrkirr as an application is a strong advocate of learners having control over their software. From being able to adjust visual, network and Notes settings, options such as loading of profiles can also be turned on or off. With the additional tools the scene is as follows:

- In Tutor, the learner can choose which worksheet he should work on and can decide to change worksheets anytime he/she wishes.

- While using QuizMaster, the learner can choose between the Beginner or Advanced levels and expect different set of words every time.

j. Is the Media integrated with the software?

The answer to this question would be negative as far as both Tutor and QuizMaster are concerned. Kirrkirr, however, is rich in Multimedia utility as it has a panel that deals solely with providing the sounds and pictures of the current word browsed by the learner.

6.4   Discussion

From the above evaluative exercise, it is clear that the new Kirrkirr has most of the features of an effective CALL tool. For a correct interpretation of the above data, however, a few preliminary remarks are called for (Groot, 2000). The analysis does not pretend to show the learning effect of each separate part of the Kirrkirr program but rather the overall effect of the Kirrkirr induced learning process. It is difficult, from a psycholinguistic perspective, to discriminate between the various stages in the word acquisition process. All one can logically say is that there must be a temporal order: noticing a word's properties must of necessity precede any storage, and consolidation can only follow when something has been stored. But even if it were desirable from a theoretical point of view, it is practically hardly possible to determine the relative contribution to the learning effect of the various stages in the learning process.

It is not unlikely that a lack of familiarity with the Kirrkirr method of word learning may negatively influence acquisition of Warlpiri. New learners might be conditioned for superficial learning and may find it difficult to switch to a different style. Hence even if Kirrkirr scored a full mark on each of the design criteria, it would be meaningless to say that it was equally effective. Sometimes drastic changes in the instructional method adversely affect the learning efficiency of the instructional tool. Kirrkirr design, and hence Tutor and QuizMaster design, relies on the fact that the application was developed to be a teaching aid to the linguistic teacher. It is at its maximum efficiency when used in conjunction with a human instructor who can perform the analyses for the student.

Chapter - 7 CONCLUSION AND FUTURE WORK

7.1   Future Work

While Kirrkirr currently deals with the SLA process through vocabulary acquisition, SLA through contextual means could also be pursued at the same time making the module Adaptive[1] to the learner. This may involve the development and add-on of an adaptive "Storyteller" that would narrate a story or engage the learner with role-plays, interviews, compositions, and even presentations and subject the learner to a number of item-response questions which could be based on what the learner participated in earlier. The Storyteller could then use a combination of item response theory and the concept of flexilevel tests (Lord, 1980) to create a test specifically designed for the individual student taking it.

Future developers on Kirrkirr could also strive to build an application in a "instructor-less" environment. In essence, if students can be guided to use the computer as a complement to a classroom instruction, the increase in practice time can help to more closely approach the advantages of total immersion learning. Features such as tracking learner answers and looking for patterns, responding not only with whether the answer was correct but also why it was right or wrong; Offering suggestions for further study - going on to a more advanced level or doing some extra work at the current or a previous level. The system could contain a Student Model that keeps a record of students' performance histories. The information about each learner in the Student Model would then determine the level of specificity of feedback messages, clues, and exercise difficulty. All these would work towards making Kirrkirr an Intelligent CALL tool.

Going one step further, it is recommended that it be made possible to network the different "Kirrkirr"s installed, with a simple communication software, so that learners can either chat or interact with their instructors and with other learners of Kirrkirr in the target language and hence reap the benefits of collaboration and competition with their peers. While this will also facilitate distance learning, an internet connection will be required but considering the spread of the net and increased broadband, this is a relatively smaller challenge to overcome.

It would be nice to have Profiles work in Kirrkirr in a way that a "Theme" works with a "Username" on the Windows OS. The Profile can be made to encompass the entire application. In addition to the current features, each Profile could lead to a different Look & Feel for Kirrkirr, different Language for the display of instructions and loading of different worksheets by the Tutor, automatic reminder of words that had gotten wrong in the last Quiz or even a different kind of story-loading in the Storyteller.

7.2   Conclusion

Today, computing power is no longer a limitation for implementation of learning tools. The problem has moved from how to implement to what tools to implement and why. This report provides a discussion of the bases of the development of learning tools for Kirrkirr. The author addresses a range of issues pertaining to CALL software development, from design, implementation to factors that make them effective and talks in detail about the efforts taken to transform Kirrkirr into a comprehensive

---

[1] In a Computer-adaptive language test items are selected and fitted according to the individual students involved and the test is ended when the student's ability level is located. The flexilevel procedures roughly determine the general ability level of the student within the first few test questions. Then, based on item response statistics, the computer selects items which are suitable for the student's particular level and administers those items in order to get a more finely tuned estimate of the student's ability level. This eliminates the need for students to answer numerous questions that are too difficult or too easy for them. They will be working on test items that are appropriate for their own individual ability levels.

integrated language acquisition tool. To this end, what actually makes linguistic software effective was first dealt with and then these criteria applied to design suitable tools for Kirrkirr.

The author does not discuss how collected information is used to assess the actual knowledge a student has gained, nor does she deal with the issue of how to adapt the system output and responses based on that information, this is because Kirrkirr is currently designed as an instructional aid to the L2 instructor and as an information provider. It leaves the analyses and adaptation to the human linguistic teacher and provides him/her with a convenient and interesting method of teaching and testing. The author regards teachers as facilitators and hence must, in many ways, know more than they would as directive givers of information. Facilitators must be aware of a variety of material available for improving students' language skill and hence capable of designing suitable worksheets etc.

Chapter 3 discusses Tutor, the CALL tool with a difference. Tutor seeks to be a highly configurable teaching tool serving both instructors and learners alike by allowing any style of teaching, be it presentation with slides, or contextual to syntactic questions and a record of learner performance. Chapter 4 introduces and details the development of QuizMaster, a CBT module that tests the learner's vocabulary at two levels, Beginners and Advanced. To enhance the functionality and effectiveness of Kirrkirr as a CALL tool, additional features were added in. The functionalities of Profiles and the ability to generate HTML using the specified XSL file were included.

It was hoped that these brought Kirrkirr one-step closer to being completely effective according to current research findings. However, research on the effectiveness of computers in language teaching is itself an ongoing process. With the rapid changes in computer technology as well, zeroing in on whether the CALL tool enhances learning is like aiming at a moving target. As both teachers and students become more familiar with the technology, they respond to it in different ways, which might lead to a new set of criteria that constitute effectiveness. Still, research can give some insight into how technology can and should be used, even if there are no definitive answers. The changes or lack thereof over time in what teachers and students do with and think about technology provide a perspective in viewing the role of CALL tools in education, and maybe even some ideas about activities that motivate students and encourage learning.

At this point, what is most clear as a result of ongoing research is that tools yielding information that is specific to a given student population and allowing teachers to track students' progress over time are deemed more useful than other tools. Tools also need to measure those aspects and outcomes of learning that would otherwise be unattainable without the use of technology. This thesis can be viewed as a first step towards achieving such a comprehensive, integrated tool for language learning.

APPENDIX

A1. WARLPIRI DICTIONARY WORKSHEET

YIRDI --------_____
[= name]

1. Look up words
Read the sentences,
Karrawari, karnalu ngarrirni - wapurnungku piya yijala -- piirn-pari.
Ngulaji ka karri wirringkajuku. Yinjirirlajuku karrawariji.
Answer the questions.
Nyiya KARRAWARI-ji?_____
Nyiya YINJIRI-ji?_____
[nyiya = what, ji is a focus marker of some sort]

2. Same as, or See also
This means that if you go to the other word you will find more information.
Look up PAPIRTA. You can see that it doesnt tell you what it means in English. So, now follow the Same as word.
What does PAPIRTA mean?_____

3. Alternant forms
You can see that papirta and paparda are just different ways of saying the same word. We call these ALTERNANT FORMS.
Look up YAPARALJI and find how many different ways people can say it._____

4. Same meaning, or   Synonyms
A SYNONYM of a word is another word that means the same. Kinyuwurru is a synonym of janmarda.
Look up KARARRPA and find a synonym.____

Look up YARLA. How many synonyms can you find for it?_____._____._____.

Ngapurla , watiya kirrirdimpayi kujaka manangkarrarla pardimi. Yirdijarra ka karri -- ngapurla, nyiya yirdi-kari?._____

5. Senses
Some words have more than one meaning. These meanings are called SENSES. Usually dictionaries give them numbers,1, 2, 3.
Look up WUNJU. Find out 3 different senses for it.
1.
2.
3.

6. Other information
You can find out different things about words in dictionaries.
Look up YARLIRLKARLIRLKI and answer the question.
Ngarni kalu YARLIRLKARLIRLKI-ji?._____.___
[= do they eat ..]

A2. Worksheet in XML form

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE BOOK [
<!ELEMENT BOOK (ENTRY+) >
<!ELEMENT ENTRY (Question+,Description?,Answer+)>
<!ELEMENT Question (#PCDATA)>
<!ELEMENT Description (#PCDATA)>
<!ELEMENT Answer (#PCDATA)>
]>

<BOOK>

<ENTRY>
        <Question>Look up words</Question>
        <Description>nyiya = what, ji is a focus marker of some sort</Description>
        <Answer></Answer>
</ENTRY>

<ENTRY>
        <Question>Read the sentences,
        1. Karrawari, karnalu ngarrirni - wapurnungku piya yijala -- piirn-pari.
        2. Ngulaji ka karri wirringkajuku. Yinjirirlajuku karrawariji.
        </Question>
        <Description>nyiya = what, ji is a focus marker of some sort</Description>
        <Answer>answer 1 to be entered here</Answer>
</ENTRY>

<ENTRY>
        <Question>Answer the questions.</Question>
        <Description>nyiya = what, ji is a focus marker of some sort</Description>
        <Answer>answer 2 to be entered here</Answer>
</ENTRY>

<ENTRY>
        <Question>Nyiya KARRAWARI-ji?</Question>
        <Description>nyiya = what, ji is a focus marker of some sort</Description>
        <Answer>answer 3 to be entered here</Answer>
</ENTRY>

<ENTRY>
        <Question>Nyiya YINJIRI-ji?</Question>
        <Description>nyiya = what, ji is a focus marker of some sort</Description>
        <Answer>answer 4 to be entered here</Answer>
</ENTRY>

<ENTRY>
        <Question>Same as, or See also  </Question>
        <Description>desc</Description>
        <Answer>answer 5 to be entered here</Answer>
</ENTRY>

<ENTRY>
        <Question>This means that if you go to the other word you will find more information</Question>
```

```
        <Description>desc</Description>
        <Answer>answer 6 to be entered here</Answer>
</ENTRY>

<ENTRY>
        <Question>Look up PAPIRTA.  You can see that it doesnt tell you what it means in English.
        So, now follow the Same as word.
        </Question>
        <Description>desc</Description>
        <Answer>answer 7 to be entered here</Answer>
</ENTRY>

<ENTRY>
        <Question>What does PAPIRTA mean?</Question>
        <Description>desc</Description>
        <Answer>answer 8 to be entered here</Answer>
</ENTRY>

<ENTRY>
        <Question>Alternant forms</Question>
        <Description>desc</Description>
        <Answer>answer 9 to be entered here</Answer>
</ENTRY>

<ENTRY>
        <Question>You can see that  papirta and paparda are just different ways of saying
        the same word.  We call these ALTERNANT FORMS.
        </Question>
        <Description>desc</Description>
        <Answer>answer 10 to be entered here</Answer>
</ENTRY>

<ENTRY>
        <Question>Look up YAPARALJI and find how many different ways people can say it</Question>
        <Description>desc</Description>
        <Answer>answer 11 to be entered here</Answer>
</ENTRY>

<ENTRY>
        <Question> Same meaning or Synonyms</Question>
        <Description>desc</Description>
        <Answer>answer 12 to be entered here</Answer>
</ENTRY>

<ENTRY>
        <Question>A SYNONYM of a word is another word that means the same. Kinyuwurru is a
        synonym of janmarda.
        </Question>
        <Description>desc</Description>
        <Answer>answer 13 to be entered here</Answer>
</ENTRY>

<ENTRY>
        <Question>Look up KARARRPA and find a synonym</Question>
        <Description>desc</Description>
        <Answer>answer 14 to be entered here</Answer>
</ENTRY>
```

```
<ENTRY>
        <Question>Look up YARLA. How many synonyms can you find for it?</Question>
        <Description>desc</Description>
        <Answer>answer 15 to be entered here</Answer>
</ENTRY>

<ENTRY>
        <Question>Ngapurla , watiya kirrirdimpayi kujaka manangkarrarla pardimi. Yirdijarra
        ka karri -- ngapurla,  nyiya yirdi-kari?</Question>
        <Description>desc</Description>
        <Answer>answer 16 to be entered here</Answer>
</ENTRY>

<ENTRY>
        <Question>Senses</Question>
        <Description>desc</Description>
        <Answer>answer 17 to be entered here</Answer>
</ENTRY>

<ENTRY>
        <Question>Some words have more than one meaning.  These meanings are called SENSES.
        Usually dictionaries give them numbers,1, 2, 3 </Question>
        <Description>desc</Description>
        <Answer>answer 18 to be entered here</Answer>
</ENTRY>

<ENTRY>
        <Question>Look up WUNJU.  Find out 3 different senses for it.</Question>
        <Description>desc</Description>
        <Answer>answer 19 to be entered here</Answer>
</ENTRY>

<ENTRY>
        <Question>Enter First Sense</Question>
        <Description>desc</Description>
        <Answer>answer 20 to be entered here</Answer>
</ENTRY>

<ENTRY>
        <Question>Enter Second Sense</Question>
        <Description>desc</Description>
        <Answer>answer 21 to be entered here</Answer>
</ENTRY>

<ENTRY>
        <Question>Enter Third Sense</Question>
        <Description>desc</Description>
        <Answer>answer 22 to be entered here</Answer>
</ENTRY>

<ENTRY>
        <Question> Other information</Question>
        <Description>desc</Description>
        <Answer>answer 23 to be entered here</Answer>
</ENTRY>
```

```
<ENTRY>
        <Question>You can find out different things about words in dictionaries.</Question>
        <Description>desc</Description>
        <Answer>answer 24 to be entered here</Answer>
</ENTRY>

<ENTRY>
        <Question>Look up YARLIRLKARLIRLKI and answer the question.</Question>
        <Description>desc</Description>
        <Answer>answer 25 to be entered here</Answer>
</ENTRY>

<ENTRY>
        <Question>Ngarni kalu YARLIRLKARLIRLKI-ji?</Question>
        <Description>= do they eat ..</Description>
        <Answer>answer 26 to be entered here</Answer>
</ENTRY>

</BOOK>
```
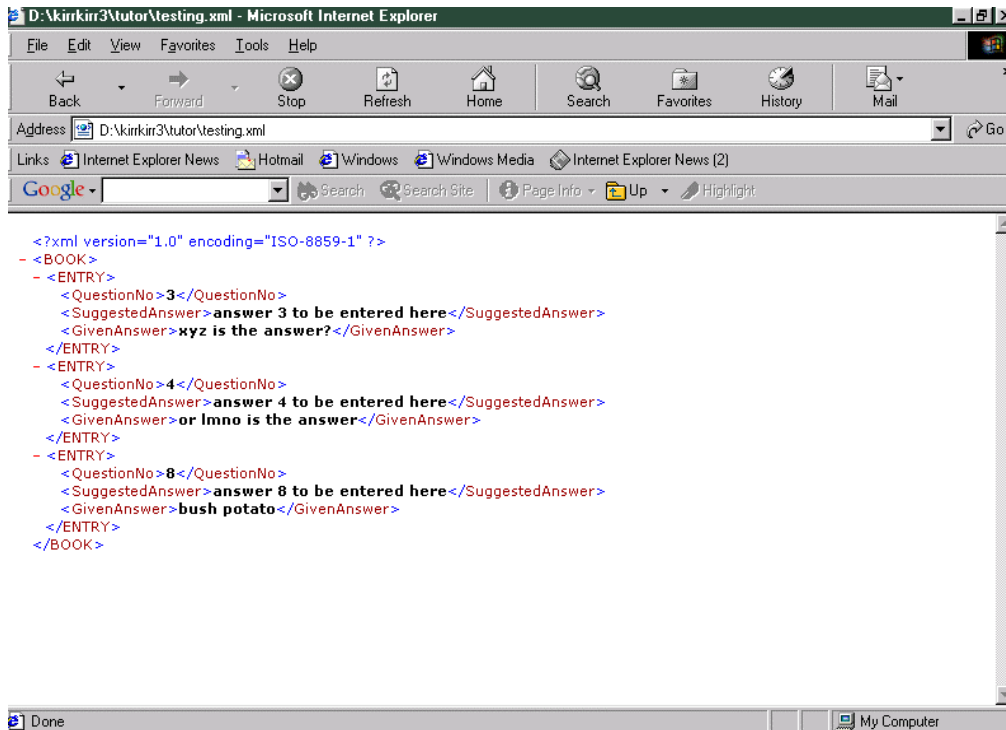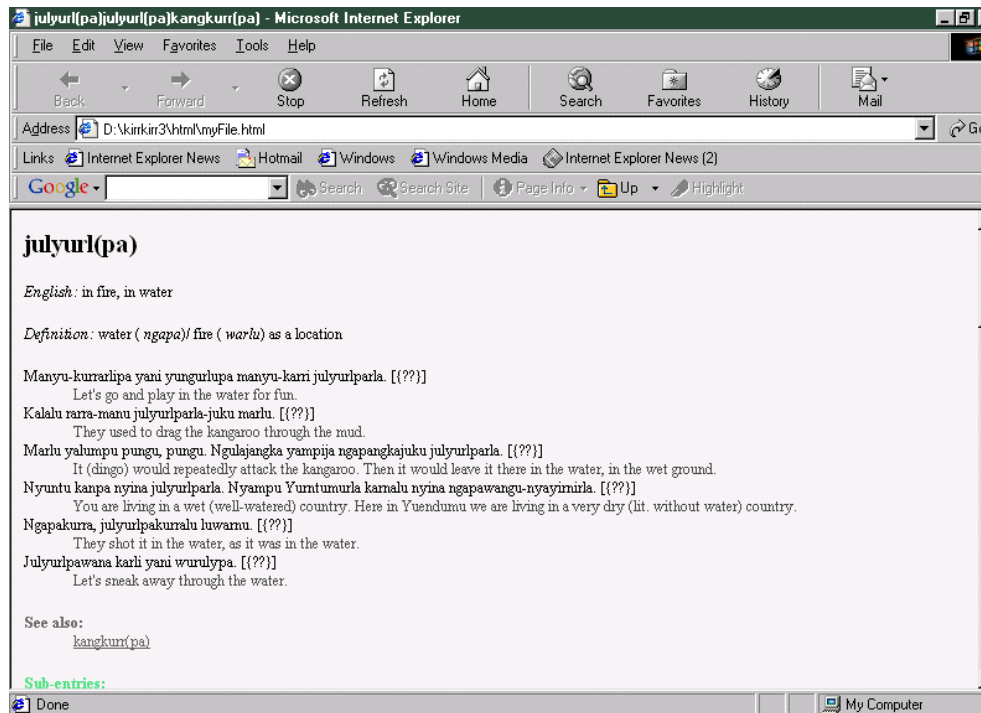
A3. Answersheet generated by Tutor

A4. HTML file generated by the "Generate HTML" option

Bibliography

[Allwright]
Allwright, D. (1988). *Observation in the language classroom.* London: Longman.

[Allwright & Bailey]
Allwright, D., & Bailey, K. (1991*). Focus on the language classroom: An introduction to classroom research for language teachers*. Cambridge: Cambridge University Press.

[Baker]
Baker, F. (1989). *Computer technology in test construction and processing. In R. L. Linn Educational measurement* (3rd ed., pp. 409-428). London: Collier Macmillan.

[Brown]
Brown, J., (1997). Computers in Language Testing: Present Research and Some Future Directions. *Language Learning & Technology*, Vol. 1, No. 1, July 1997, pp. 44-59

Brown, J. (1992b). Using computers in language testing. Cross Currents, 19, 92-99.

[Bunderson & Inouye & Olsen]
Bunderson, C., Inouye, D., & Olsen, J. (1989). The four generations of computerized educational measurement. In R. L. Linn (Ed.), Educational measurement (3rd ed., pp. 367-407). London: Collier Macmillan.

[Carslon]
Carslon, R. (1994). Computer-adaptive testing: A shift in the evaluation paradigm. *Journal of Educational Technology Systems*, 22, 213-224.

[Chapelle]
Chapelle, C. (1998). Multimedia CALL: Lessons to be learned from research on instructed SLA. *Language Learning & Technology*. Vol. 2, No. 1, July 1998, pp. 22-34

Chapelle, C. (1997). Call in the year 2000: Still in search of Research Paradigms? *Language Learning & Technology*. Vol 1, No 1, 19-43.

[Chaudron]
Chaudron, C. (1988). *Second language classrooms: Research on teaching and learning*. Cambridge: Cambridge University Press.

[Chua]
Chua, E. (1999). Tools for Information Modelling, Final Year Report, Nanyang Technological University.

[Day]
Day, R. (Eds.). (1986). *Talking to learn: Conversation in second language acquisition*. Rowley, MA: Newbury House.

[Dole, Duffy, Roehler & Pearson]
Dole, J., Duffy, G., Roehler, L., & Pearson, P. (1991). Moving from the old to the new: research on reading comprehension instruction. Review of Educational Research, 61, 239-264.

[Dorans]
Dorans, N. (1990). Scaling and equating. In H. Wainer, N. J. Dorans, R. Flaugher, B. F. Green, R. J. Mislevy, L. Steinberg, & D. Thissen (Eds.), Computerized adaptive testing: A primer (pp. 137-160). Hillsdale, NJ: Lawrence Erlbaum.

[Doughty]
Doughty, C. (1991). Second language instruction does make a difference: Evidence from an empirical study of SL relativization. Studies in Second Language Acquisition, 13, 431-469.

[Drasgow, Levine & McLaughlin]
Drasgow, F., Levine, M., & McLaughlin, M. (1987). Detecting inappropriate test scores with optimal and practical appropriateness indices. Applied Psychological Measurement, 11, 59-80.

[Dunkel]
Dunkel, P. (Ed.), Considerations in developing or using second/foreign language proficiency computer-adaptive tests, *Language Learning & Technology*, Vol. 2, No. 2, January 1999, pp. 77-93

[Gass]
Gass, S. (1997). *Input, interaction, and the second language learner*. Mahwah, NJ: Lawrence Erlbaum Associates Publishers.

[Gay]
Gay, G. (1987). Interaction of learner control and prior understanding in computer-assisted video instruction. Journal of Educational Psychology, 78, 225-227.

[Green]
Green, B. (1990). *System design and operations.* In H. Wainer, N. Dorans, R. Flaugher, B. Green, R. Mislevy, L. Steinberg, & D. Thissen (Eds.), Computerized adaptive testing: A primer (pp. 23-40). Hillsdale, NJ: Lawrence Erlbaum.

[Groot]
Groot, P. (2000). Computer Assisted Second Language Vocabulary Acquisition. *Language Learning & Technology* Journal: Vol. 4, No. 1, pp60-81.

[Healey]
Healey, D. (1993). Learner choices in self-directed second language learning. Ann Arbor, MI: UMI Dissertation Services.

[Jaeger]
Jaeger, R. (1989). Certification of student competence. In R. L. Linn (Ed.), Educational measurement (3rd ed., pp. 485-514). London: Collier Macmillan.

[Jansz]
Jansz, K. (1998). Intelligent processing, storage and visualization of dictionary information University of Sydney Computer Science Honors thesis.

[Jansz, Manning & Indurkhya]
Jansz, K., Manning, C. & Indurkhya, N. (1999). Kirrkirr: Interactive Visualisation and multimedia from a Structured Warlpiri Dictionary. Proceedings of AusWeb99, the Fifth Australian World Wide Web Conference, pp. 302-316.

[Johanesen & Tennyson]
Johanesen, K. & Tennyson, R. (1983). Effect of adaptive advisement on perception in learner-controlled, computer-based instruction using a rule-learning task. Educational Communication and Technology, 31, 226-236.

[Johnson]
Johnson, K. (1995). *Understanding communication in second language classrooms*. Cambridge: Cambridge University Press.]

[Kingsbury & Houser]
Kingsbury, G., & Houser, R. (1993). Assessing the utility of item response models: Computer adaptive testing. Educational Measurement: Issues and Practice, 12, 21-27.

[Krashen]
Krashen, S. (1982). *Principles and practice in second language acquisition*. Oxford: Pergamon.

[Larsen-Freeman & Long]
Larsen-Freeman, D., & Long, M. (1991). *An introduction to second language acquisition research*. London: Longman.

[Lewis & Sheehan]
Lewis, C., Sheehan, K. (1990). Using Bayesian decision theory to design a computerized adaptive mastery test. Applied Psychological Measurement, 14, 367-386.

[Long]
Long, M. (1996). The role of linguistic environment in second language acquisition. In W. C. Ritchie, & T. K. Bhatia, (Eds.), Handbook of second language acquisition (pp. 413-468). San Diego: Academic Press.

Long, M. (1988). Instructed interlanguage development. In L. Beebe (Ed.), Issues in second language acquisition: Multiple perspectives (pp. 115-141). New York: Newbury House.

Long, M. (1980). Inside the "black box": Methodological issues in classroom research on language learning. Language Learning, 30, 1-42.

[Lord]
Lord, F (1980). Applications of item response theory to practical testing problems. Hillsdale, NJ: Lawrence Erlbaum.

[Mazzeo, Druesne, Raffeld, Checketts & Muhlstein]
Mazzeo, J., Druesne, B., Raffeld, P., Checketts, K., & Muhlstein, A. (1991). Comparability of computer and paper-and-pencil scores for two CLEP general examinations (Report #91-5). New York: College Entrance Examination Board.

[Mondria]
Mondria, J. (1996). Vocabulaireverwerving in het vreemde-talenonderwijs. De effecten van context en raden op de retentie [Vocabulary acquisition in foreign language teaching. The effects of context and guessing on retention]. Doctoral dissertation. The Netherlands: Groningen University Press.

[Nagy & Herman]
Nagy, W., & Herman, P. (1987). Breadth and depth of vocabulary knowledge: Implications for acquisition and instruction. In M.G. McKeown & M. Curtis (Eds.), The nature of vocabulary acquisitio (pp. 19-35). Hillsdale, NJ: Erlbaum.

[Nobuyoshi & Ellis]
Nobuyoshi, J., & Ellis, R. (1993). Focused communication tasks. English Language Teaching Journal, 47, 203-210.

[O'Neill, Folk, & Li]
O'Neill, K., Folk, V., & Li, M.-Y. (1993). Report on the pretest calibration study for the computer-based academic skills assessments of The Praxis Series: Professional assessments for beginning teachers. Princeton, NJ: Educational Testing Service.

[Oxford]
Oxford, R. (1990). Language learning strategies: What every teacher should know. New York: Newbury House Publishers.

[Pica]
Pica, T. (1994). Research on negotiation: What does it reveal about second-language learning conditions, processes, and outcomes? Language Learning, 44(3), 493-527.

[Reckase]
Reckase, M. (1983). A procedure for decision making using tailored testing. In D. J. Weiss (Ed.), New horizons in testing: Latent trait test theory and computerized adaptive testing (pp. 237-255). New York: Academic Press.

[Robinson]
Robinson, G. (1989). The CLCCS CALL study: Methods, error feedback, attitudes, and achievement. In W.F. Smith (Ed.), Modern technology in foreign language education: Applications and projects. Lincolnwood, IL: National Textbook Company.

[Rubin]
Rubin, J. (1987). Learner strategies: theoretical assumption, research, history, and typology. In A. Wenden and J. Rubin (Eds.), Learner strategies in language learning (pp. 15-30). Englewood Cliffs, NJ: Prentice-Hall.

[Schaeffer]
Schaeffer, R. (1981). Meaningful practice on the computer: Is it possible? *Foreign Language Annals,* 14 (2), 133-137.

[Schaeffer, Steffen & Golub-Smith]
Schaeffer, G., Steffen, M., & Golub-Smith, M. (1993). *Introduction of a computer adaptive GRE General Test* (Research Report # 93-57). Princeton, NJ: Educational Testing Service.

[Schmidt & Frota]
Schmidt, R., & Frota, S. (1986). Developing basic conversational ability in a second language: A case study of an adult learner of Portuguese. In R. Day (Ed.)*, Talking to learn: Conversation in second language acquisition.* Rowley, MA: Newbury House.

[Seliger & Long]
Seliger, H., & Long, M. (Eds.). (1983). *Classroom oriented research in second language acquisition.* Rowley, MA: Newbury House Publishers.

[Simpson]
Simpson, J. (2000). A Warlpiri worksheet. Unpublished private communication.

[Smith]
Smith, M. (1993). Input enhancement in instructed SLA: Theoretical bases. *Studies in Second Language Acquisition, 15*, 165-179.

Smith, M. (1991). Speaking to many minds. *Second Language Research, 7*, 118-132.

[Sng]
Sng, J. (2000). Optimising Kirrkirr: An electronic dictionary. Final Year Project, School of Applied Science, Nanyang Technological University, Singapore. Technical report AS00-CE-F076.

[Steinberg]
Steinberg, E. (1991). *Computer-Assisted Instruction: A Synthesis of Theory, Practice, and Technology*, Hillsdale, NJ: Lawrence Erlbaum Associates, 1991.

[Stevenson, & Gross]
Stevenson, J., & Gross, S. (1991). Use of a computerized adaptive testing model for ESOL/bilingual entry/exit decision making. In P. Dunkel (Ed.), *Computer-assisted language learning and testing: Research issues and practice* (pp. 223-235). New York: Newbury House.

[Stocking]
Stocking, M. (1987). To simulated feasibility studies in computerized adaptive testing. *Applied Psychology: An International Review, 35,* 263-277.

[Swain]
Swain, M. (1985). Communicative competence: Some roles of comprehensible input and comprehensible output in its development. In Gass, S. & Madden, C. (Eds.)*, Input in second language acquisition* (pp. 235-253). Rowley, MA: Newbury House Publishers.

[Swain & Lapkin]
Swain, M., & Lapkin, S. (1995). Problems in output and the cognitive processes they generate: A step towards second language learning. *Applied Linguistics, 16,* 371-391.

[van Lier]
van Lier, L. (1988). *The classroom and the learner*. London: Longman.

[Wainer, Dorans, Green, Mislevy, Steinberg, & Thissen]
Wainer, H., Dorans, N., Green, B., Mislevy, R., Steinberg, L., & Thissen, D. (1990). Future challenges. In H. Wainer, N. Dorans, R. Flaugher, B. Green, R. J. Mislevy, L. Steinberg, & D. Thissen (Eds.), *Computerized adaptive testing: A prime*r (pp. 233-272). Hillsdale, NJ: Lawrence Erlbaum.

[Ward]
Ward, W. (1988). The College Board Computerized Placement Tests: An application of computerized adaptive testing. *Machine-Mediated Learning, 2,* 217-282.

[Warschauer]
Warschauer, M. (1996b). Computer-assisted language learning: an introduction. In S. Fotos (Ed.), *Multimedia Language Teaching* , 3-20. Tokyo: Logos.

[Warschauer & Healey]
Warschauer, M., & Healey, D. (1998). Computers and language learning: An overview. *Language Teaching*, 31, 57-71

[Van Dijk, & Kintsch]
Van Dijk, T. & Kintsch, W. (1983). *Strategies of discourse comprehension*. Orlando, FL: Academic Press.