SLLS: An Online Conversational Spoken Language Learning System

by

Tien-Lok Jonathan Lau

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2003

Author .....
Department of Electrical Engineering and Computer Science
May 9, 2003

Certified by .....
Stephanie Seneff
Principal Research Scientist
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

SLLS: An Online Conversational Spoken Language Learning System

by

Tien-Lok Jonathan Lau

Submitted to the Department of Electrical Engineering and Computer Science
on May 9, 2003, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Computer Science and Engineering

Abstract

The Spoken Language Learning System (SLLS) is intended to be an engaging, educational, and extensible spoken language learning system showcasing the multilingual capabilities of the Spoken Language Systems Group's (SLS) systems. The motivation behind SLLS is to satisfy both the demand for spoken language learning in an increasingly multi-cultural society and the desire for continued development of the multilingual systems at SLS. SLLS is an integration of an Internet presence with augmentations to SLS's Mandarin systems built within the Galaxy architecture, focusing on the situation of an English speaker learning Mandarin. We offer language learners the ability to listen to spoken phrases and simulated conversations online, engage in interactive dynamic conversations over the telephone, and review audio and visual feedback of their conversations. We also provide a wide array of administration and maintenance features online for teachers and administrators to facilitate continued system development and user interaction, such as lesson plan creation, vocabulary management, and a requests forum. User studies have shown that there is an appreciation for the potential of the system and that the core operation is intuitive and entertaining. The studies have also helped to illuminate the vast array of future work necessary to further polish the language learning experience and reduce the administrative burden. The focus of this thesis is the creation of the first iteration of SLLS; we believe we have taken the first step down the long but hopeful path towards helping people speak a foreign language.

Thesis Supervisor: Stephanie Seneff
Title: Principal Research Scientist

Acknowledgments

First and foremost, I would like to thank Stephanie Seneff, my thesis advisor, for the opportunity to work with her on this system. With little knowledge of speech and linguistics technology, I came to Stephanie with the modest skills that I possessed. She was able to tap those skills, and together we were able to formulate and develop this exciting system that hopefully will continue past my tenure at SLS. Her comments, suggestions, and critiques acted as a guiding hand to my work, and our discussions were always a boiling pot of new ideas waiting to be developed! I cannot emphasize enough how thankful I am for the confidence Stephanie had in me and how enjoyable it has been to work with her this past year.

Lucky for Stephanie, she was not the only person I was able to rely on! The supportive environment at SLS was also instrumental in the success of this thesis. I would especially like to thank Chao Wang, Scott Cyphers, Min Tang, Mitch Peabody, Michelle Spina, and Richard Hu. Chao not only supplied the voice for the system, but also shared the burden of solving certain complexities of the SLS systems. She was sorely missed during her pregnancy, and the energy and vigor she had when she returned was essential to SLLS. The breadth of Scott's knowledge is amazing to behold, and he was the resource for all my system design and technology decisions. Min also lent his voice to the system, but more importantly, was a source of information for issues large and small, from the latest news to the workings of LATEX. Mitch will be taking over the system once I'm gone, so I would just like to wish him the best of luck. His perseverance in the battle with obscure bugs to produce the word scoring is most admirable. Michelle so kindly introduced me to various tools that helped me process new phrases. Richard and I worked together to figure out how Frame Relay and Galaxy worked before branching off on our separate projects. The headway he made really jump started my work, and enabled me to progress faster than expected.

I would also like to thank all my friends for all the good times we've had, especially throughout this year - Catherine, Jordan, Kelvin, Patrick, Tim, Ronald, Nick, Edmund, Kenny, Peter, Wen, Joe and numerous others. I've never had housemates

before, and it was surprisingly fun. Hopefully when we all start working, the good times will continue! Good luck to all of you in your future endeavors.

Lastly, I would like to thank my family, who have always been my source of motivation and strength. My life has been full of peaks and valleys, and my family has always been there for me. Although my family is complicated, I was always provided a warm and caring environment, which has allowed me to become what I am today. Thank you all! Thanks mom! I'm done!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Since 1989, the Spoken Language Systems Group (SLS) at the Massachusetts Institute of Technology has been conducting research in the development of conversational systems. The motivation behind this research is the belief that one of the best ways to increase the naturalness of human-computer interaction is to mimic interpersonal interaction through the use of spoken conversations. Tasks that lend themselves to the adoption of computers are typically data intensive, structured, time-consuming and able to be automated. The systems developed by SLS thus far have focused on tasks that fit this description - urban navigation, air travel planning, weather forecasting and reminders. Due to the cultural independence of these tasks, the logical step for the research was taken to extend them into the multilingual domain, allowing global access to these tasks through multiple spoken languages. As a result of these efforts, there are now a number of powerful multilingual systems in development at SLS, making it possible to create a language learning system envisioned in [19]. The focus of this thesis is to take the first step towards that vision, extending the existing multilingual SLS technologies to create the first iteration of an interactive online spoken language learning system.

1.1 Motivation

Our increasingly intertwined global society has enabled a vast array of communication mechanisms, from instant messaging to video conferencing, connecting more people around the world than ever before. Yet even with all this technological infrastructure in place, cross-cultural communication remains remarkably rare because of the language barrier. As the technological hurdles to global communication are gradually overcome, the human inability to overcome this barrier has steadily become the limiting factor in global communication. Translators are required in droves to enable cross-cultural interaction, which is limiting and impractical for the average person. There have been attempts at providing real-time computer translation over the telephone, most notably [1] and [17], but these technologies are far from ready for usage. For the average person today, to communicate naturally with a foreigner requires learning their language.

The main components to language learning are reading, writing and speaking. In classroom style instruction, reading and writing skills are usually more heavily emphasized because they can be easily tested individually. Speaking skills are often honed through group readings of text and little else. In smaller classes, there may be individual spoken exams, but these are both time-consuming and infrequent. As a result, students are more confident about their reading and writing skills, but even after extensive study, are fearful of conversing in a foreign language because they have limited opportunities to interact in practical settings without fear of embarrassment. However, speaking ability is by far the most practical language skill, especially when visiting a foreign country for the first time. There is no better way to gain the respect and understanding of strangers in a foreign country than by conversing with them in their native tongue. Unfortunately, it is also the most difficult aspect in picking up a new language, especially given the subtle variations in tone, pitch and accent that accompany fluency in a language. This difficulty is amplified by our inability to evaluate our own pronunciation as beginners. Hence, although there is a definite demand for spoken language education beyond the classroom, this demand can not

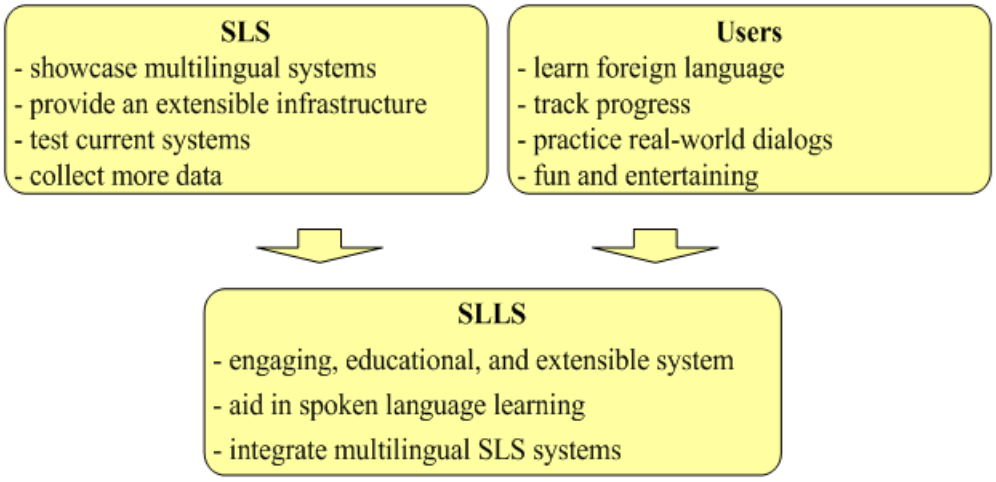


Figure 1-1: Summary of the Motivations

be adequately satisfied by simplistic systems that do not have practical interactions with the user.

At the same time, the Spoken Language Systems Group (SLS) at the Laboratory for Computer Science has been developing a number of Mandarin systems that provide speech interfaces to tasks such as translating, reminders and weather reporting through the Yishu, Lieshou [2] and Muxing systems [14] respectively. These systems are still in their infancy, and require an impetus for continued extension and development. There is also a desire to create an integrated interface for users to sample the slew of SLS multilingual offerings without having extensive language knowledge and to demonstrate the capabilities of SLS's systems.

All these factors present an exciting opportunity for us to marry the capabilities of SLS with the real-world need of spoken language learning. Many others have tried using computers in the spoken language learning process, but they have yet to succeed because they do not engage the user in a realistic way. Some of these systems are overly simplistic, attempting to transfer the instructional experience directly. Others are overly complicated and require substantial upfront costs on the part of the user, whether these costs are time or monetary. We are in the unique position of having the ability to develop a language learning system that can engage the user by leveraging the existing research at SLS.

1.2 Goals

The goals for this thesis project are to create a spoken language learning system that will:

1. Allow any users with a phone line and Internet access to engage in dynamic conversations
2. Provide audio and visual feedback to help users improve their language skills
3. Provide teachers with the means to control their students' interaction with the system and monitor students' progress
4. Provide administrative functionality for maintenance of the system and for future extensions
5. Bring together the multilingual systems at SLS
6. Demonstrate the feasibility of using SLS technologies for language learning

We believe we have succeeded in achieving these goals, and a detailed evaluation is given in Chapter 7, *Evaluation*.

1.3 Approach

We have fused an online Internet browsing experience with interactive conversations over the telephone to create the Spoken Language Learning System (SLLS). Our initial development efforts were concentrated on the specific situation of a native speaker of English learning Mandarin. We have broken down the language learning process into three stages - preparation, conversation, review. Preparation includes providing practice phrases for users to listen to, as well as generating a simulated conversation that users can follow and review. The simulated conversation not only serves as a practice to the user, but it also gives the user a sense of the types of phrases that the system expects.

After the user feels adequately prepared, the user and system can then engage in a Mandarin conversation over the telephone. The telephone conversation follows a lesson plan scripted by a teacher or administrator beforehand, but provides variability by randomly selecting different words from an equivalence class. Upon completion of the conversation, the user can then review their conversation online and listen to the various phrases spoken by both the user and the system. The user is provided with visual queues to words and phrases that the system was unable to recognize as well as access to their previous conversations.

On the administration side, teachers can log in to the site to check on their students' progress and assign new lesson plans. A default set of lesson plans are offered initially, and new ones would be created by the teachers through interaction with the administrators. Administrators can examine the user logs, view requests for new vocabulary and sentence constructs, add new capabilities, and perform user maintenance online, providing a unified interface for all typical administration tasks.

Scenarios for the student, teacher and administrator interactions with the system are described in detail in Chapter 3.

To ease the development of future generations of the SLLS, these core features will be supplemented by the design and implementation of a highly extensible framework for administrators and developers to maintain and manage the system. The completion of the most common tasks required by administrators will be possible directly through the web site, streamlining the development process.

Finally, the spoken language components will be built through augmentations to the existing Phrasebook, Orion [11], and Jupiter [18] systems within the Galaxy architecture that are described in detail in the following chapters.

1.4 Outline

In Chapter 2, we discuss the various spoken language learning systems currently available and the technologies of the SLS background to contextualize this research. We then proceed to outline the envisioned user scenarios to illuminate system operation.

In Chapter 4, we describe the augmentations that were made to the SLS systems to provide the conversational system, and this is followed by a chapter on new developments for the SLLS infrastructure. Chapter 6 is our evaluation of the system in accomplishing the goals we set out to achieve, and in Chapter 7 we offer possible future extensions for the next version of SLLS and summarize our work.

Chapter 2

Background

In this chapter, we present the background behind the conception of SLLS to provide some context for our work. We first present the prior developments in computer aided spoken language learning systems, which are typically shrink-wrapped software packages stuffed with features. We then proceed to discuss typical online spoken language learning systems that are at the other end of the spectrum, simplistic and impoverished. Finally, we outline the SLS technologies that were available when we began work on SLLS to highlight the wealth of technologies we were able to work with.

2.1 Computer Aided Spoken Language Learning

There has been a substantial growth in computer aided spoken language learning that has followed the increasing prevalence of computers in our daily life. There are countless systems that have been developed through research and also for commercialization. The quality of these systems varies tremendously, ranging from programs that play back sentences to complicated systems with lesson plans, recording and scoring [7]. However, there are a number of problems with these systems. First, they easily become stale once users have gone through the predetermined interactions and users are often forced to learn about topics that are impractical or uninteresting, and forced to go at a pace designated by the system developer. Secondly, these systems

are often limited by the computing power of the users' system, which is often difficult to predict. Most systems do not perform dynamic recognition and synthesis as yet because the average user does not have that type of computing power. Finally, there is a disconnect between the teachers and students, since in these systems, the computer plays the role of the teacher, and hence it is difficult to integrate these systems into a classroom setting.

We describe a few of the marquee systems below in the area of computer aided spoken language learning to demonstrate the current capabilities.

2.1.1 Rosetta Stone

A resource provided by a company of the same name to the U.S. State Department, NASA, and in over 9,000 classrooms worldwide, Rosetta Stone specializes in software for foreign language acquisition. Their system uses a method they call "Dynamic Immersion" that has the user linking speech with images, separating the written language learning process from the listening. For spoken language learning, they have a system that records the user's voice and supports play back for comparison with the voice of the native speaker. User utterances are then graded on a meter scale and provided to the user as a grade report. One of the most interesting things about Rosetta Stone is that they not only offer a language learning system, but they offer a whole curriculum for schools to integrate into their foreign language classes. The capabilities of this system are way beyond those of almost all of the other foreign language learning systems, commercial or otherwise, and it provides a great example of technology in the classroom.

At this juncture, we see this as a possible future model for SLLS as a real world application, and this is discussed further in Chapter 7 *Future Work*.

2.1.2 Fluency

The Fluency system developed at the Language Technologies Institute of Carnegie Mellon University uses the CMU SPHINX II speech recognizer to pinpoint pronun-

ciation errors and then gives suggestions as to how to correct them through audio and visual feedback [13]. Figure 2-1 illustrates the Fluency interface. After speaking a specified phrase, users can hear themselves; hear a native speaker; read how to pronounce a sound; see a side headcut and a front view of the lips; and hear the word in isolated form. Empirical studies on the effectiveness of this system have demonstrated that, through consistent use of the system over a period of time, users' pronunciations have improved, which shows that research on pronunciation learning is promising. The Fluency system is a work-in-progress, and serves as an excellent comparison for SLLS [5].

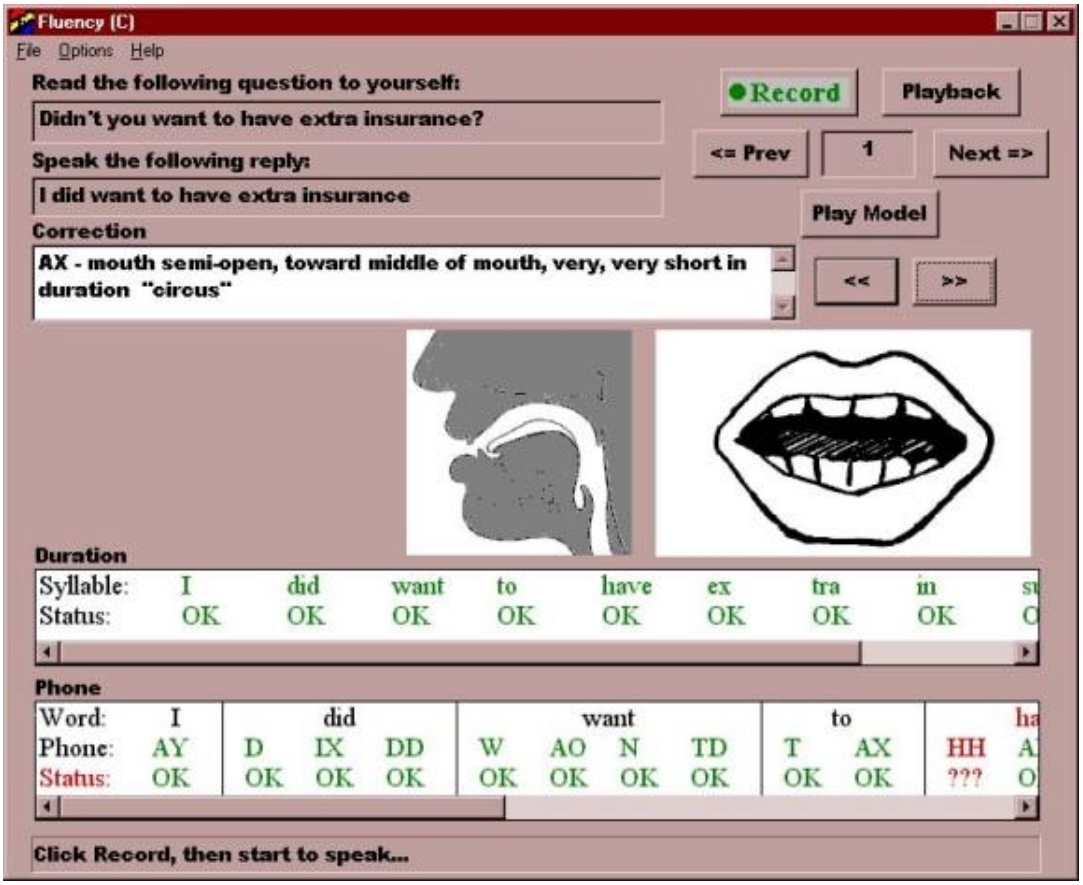


Figure 2-1: The Fluency Interface

2.1.3 PhonePass

PhonePass is a spoken English diagnostic test using speech recognition technology developed by Dr. Jared Bernstein and Dr. Brent Townshend. Users call into the system and their spoken responses are diagnosed on the exact words used, as well as the pace, fluency, and pronunciation of those words in phrases in sentences. The specially designed recognizer converts these phrases to linguistic units and these units are then compared to statistical models based on native speakers. Users are given scores on listening vocabulary, repeat accuracy, pronunciation, reading fluency, and repeat fluency. To ensure the reliability and accuracy of the system, the developers performed extensive testing, comparing the results of PhonePass with those of human graders. Their findings have shown that the performance of PhonePass is only marginally worse than its human counterpart and hence the system has been widely used in Japan at universities. Businesses have also begun to use it to screen for potential job candidates, for example during the Korean World Cup interviews for volunteers [4].

Although an excellent diagnostic tool, PhonePass does not provide any spoken language learning capability. However, it is an example of how computers can reliably and accurately evaluate human spoken language, demonstrating the feasibility of a fully automated spoken language learning system with human level feedback.

2.2 Online Spoken Language Learning

There are numerous online language learning web sites that attempt to help users learn a foreign language. Unfortunately, these are generally impoverished systems that only provide instructional information combined with limited playback capability. Many of these try to follow conversational lesson plans, with emphasis on grammar structure and the acquisition of new vocabulary. Examples of these systems are [10] and [15].

We believe that these systems act more as resources for information rather than as significant spoken language learning systems, but because they are available online,

allow their information to be easily accessible. Furthermore, although not demonstrated by the current systems, online systems have the potential to have much more sophisticated processing on the server end. Light-weight clients with access to the Internet could have features that require complex systems such as recognition and dynamic synthesis by placing these systems on remote servers, whereby the clients only act as an interface for the user. With the gradual adoption of mobile technology, this thin-client heavy-server approach will become increasingly prevalent, and we believe that this model is suitable for a language learning system.

2.3 Spoken Language Systems Group's Technologies

The sole reason we were able to embark on this ambitious project is because of the wealth of SLS technologies we were able to draw upon. The extensible architecture of Galaxy, the ability for external applications to use Frame Relay, the multilingual capabilities of Phrasebook, Jupiter and Orion, and the natural synthesis from Envoice all provided the springboard for us to create SLLS. What follows is a short description of each of these systems as they existed when we embarked on development of SLLS.

2.3.1 Galaxy

Galaxy is the architecture created by SLS for developing conversational systems. The first version of Galaxy was used to build spoken language systems that accessed online information using spoken dialogue [6]. It was subsequently enhanced to allow for more flexibility in building conversational systems [12]. The Galaxy architecture uses a central hub to mediate interaction among various Human Language Technologies (HLT) in a hub-spokes model depicted in Figure 2-2. The hub is programmed with hub scripts, a rule-based scripting language. The hub provides communication links among the various servers and specifies the commands they should execute and the order they should occur. Servers in the Galaxy architecture use *semantic frames* as

the meaning representation to represent any data they jointly process. The *Frame Relay*, described in the next section, bridges between the Galaxy hub and external systems [8].

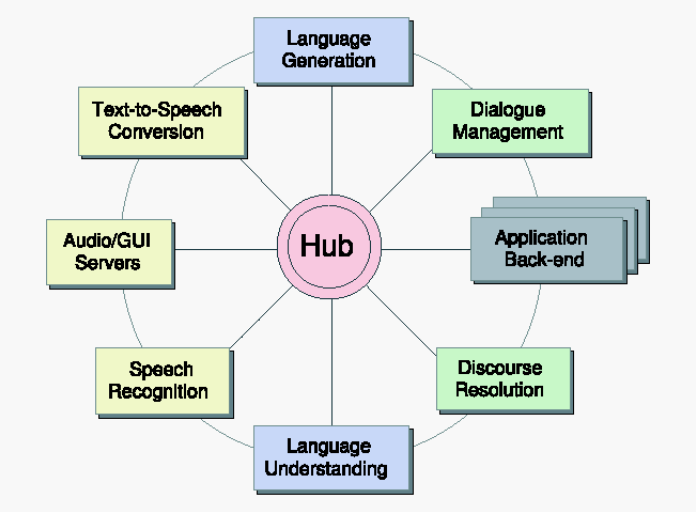


Figure 2-2: The Galaxy system architecture

The typical servers involved in a conversational system listed in the order that they are normally accessed during a dialogue turn are: the audio/GUI servers, speech recognition, language understanding, context resolution, application back-end, dialogue management, language generation and text-to-speech conversion. When a user speaks, the sound is captured by the audio/GUI servers and converted into digital form. This digital form is then processed by the speech recognition server, producing an N-best list of likely sentences, which the language understanding server would use to extract the meaning. The context resolution server will then determine the context of this utterance and attempt to resolve any ambiguities. If any information retrieval is necessary, for example weather forecasts or driving directions, the application back-end will query a database based on the parameters from the query frame. With the retrieved information, the dialogue management server can now suggest a template for generating a reply, and this template is then used to generate a phrase that is understandable to the user by the language generation server. Finally, the phrase is converted from text to speech by the text-to-speech conversion server. [3]

In addition, the Galaxy hub maintains the notion of a session that is initiated every time it detects that a new user is interacting with a component of the Galaxy architecture. Session tracking is vital for scalability because it allows Galaxy to simultaneously accommodate multiple users through a single hub instantiation. Distribution of system resources among multiple sessions is also mediated by hub scripts. User utterances are recorded to files and a detailed log file is maintained for each session.

To accommodate the multilingual capability in Galaxy, the components are required to be as language transparent as possible. This means that they should be independent of the input or output language, which not only increases modularity, but also simplifies the process of developing multilingual systems. Where language-dependency is essential, this dependency is abstracted to external tables and models. To port the system to new languages therefore only requires alterations to these external sources [19]. An illustration of a multilingual Galaxy configuration is depicted in Figure 2-3.

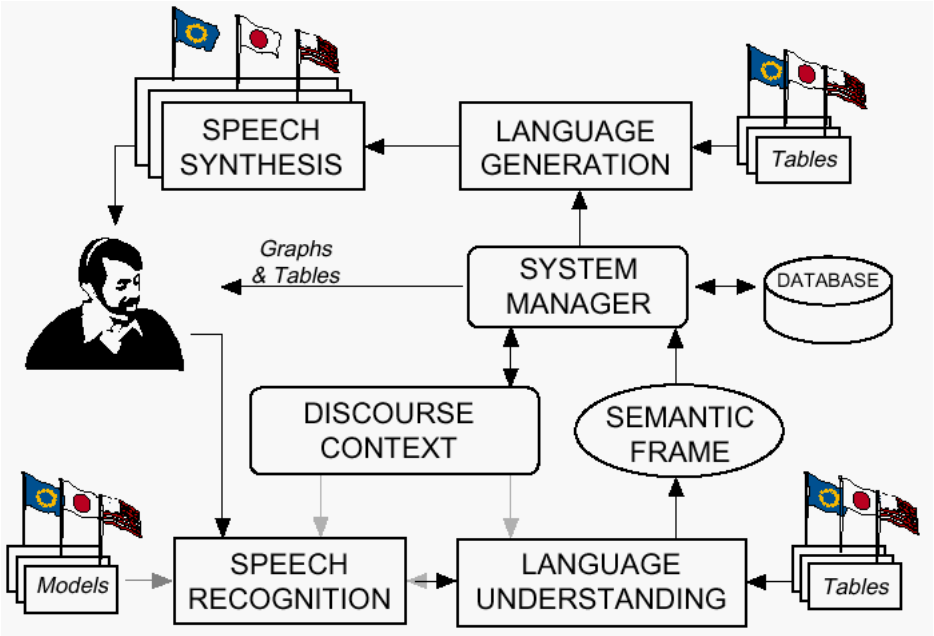


Figure 2-3: The architecture of Galaxy configured for multilingual conversations

2.3.2 Frame Relay

The Frame Relay is a recent addition to the Galaxy software suite that allows Galaxy and non-Galaxy components to communicate with each other through semantic frames [8]. Messages in the frame relay consist of an envelope and body. The envelope provides fields that specify the destination and other routing information, and the body contains the rest of the message. Messages can direct the hub to execute specific portions of the hub script and indicate special commands that control Galaxy session settings. For example, telephone calls can be initiated, audio settings can be configured, and parameters can be set.

2.3.3 Phrasebook and Yishu

Phrasebook is a speech translation system for common phrases typically used when travelling to a foreign location, such as ordering food and asking for directions. The Galaxy system was extended to provide this translation capability by adding these phrases to both the English and foreign language training corpus for the recognizer, incorporating the newly recorded speech into the synthesis library, and creating a hub script to manage the translation process and user interaction.

One of the key features of Phrasebook is the ability to detect the language spoken dynamically and generate a translation in the other language. This is possible by performing conjoined recognition on the spoken phrase with both languages and then simultaneously choosing the highest scoring hypothesis and associated language. Although this is at times less accurate than single language recognition, the user experience is vastly improved through this seamless language adaptation. Unfortunately, when recognition performance is poor, the system may detect the incorrect language and the user will hear something in his original language, which may lead them to believe that the translation system is malfunctioning. Therefore, it is necessary for Phrasebook to differ from the other spoken language systems at SLS by generating an intermediate paraphrase of the hypothesized utterance in the *same* language before generating the translation. The paraphrase provides another degree of feedback to

the user, informing the user immediately as to whether the system understood them. Figure 2-4 depicts a sample user interaction with Phrasebook and Yishu.

User:	Is there a restaurant nearby?
System Paraphrase:	Is there a restaurant in the vicinity?
System Translation:	附近有餐館嗎？ (fu4 jin4 you3 can1 guan3 ma5?)

User:	我肚子好餓。 (wo3 du4 zi3 hao3 e4.)
System Paraphrase:	我好餓。 (wo3 hao3 e4.)
System Translation:	I am hungry.

Figure 2-4: An example of a user using Phrasebook and Yishu.

The current languages supported by Phrasebook include English, Mandarin, and Spanish. Yishu is the Mandarin version of Phrasebook that we have used in the development of SLLS.

Phrasebook and Yishu provide the logical backbone to a spoken language learning system because translation is such an essential part of learning a new language. As described in Chapter 4, *Augmentations to SLS Technology*, these systems are the primary component of the user-system conversation.

2.3.4 Jupiter and Muxing

Jupiter is a telephone-based conversational weather system that provides weather information for cities around the world through spoken English queries. Users call up the system, and can ask typical questions about the weather such as "what is the temperature in Boston tonight", "how much snow will there be in New York tomorrow", and "what is the weather like in Shanghai". Muxing is Jupiter's Mandarin counterpart, built using the same basic infrastructure but tailored to Chinese.

Jupiter, like Phrasebook and Orion, required extensions to Galaxy for recognition

and synthesis coverage, and also is an example of using the back-end application to retrieve data from an external source. The key-value information extracted from the user's speech are used as terms in the query to the weather database. [14, 18]

Recently, seamless language detection has been incorporated into Jupiter, allowing users to speak English or Mandarin to the system any time during the conversation and receive a reply in the respective language. This functionality brings Jupiter, with respect to seamless language switching, in parallel with the Phrasebook system.

2.3.5 Orion and Lieshou

Orion is a conversational system that allows users to schedule tasks to be completed at a later time by the system. Some of these tasks are calling the user with reminders or calling the user to provide weather information. Orion is a departure from some of the other systems at SLS because of its two stage interaction model, namely the task enrollment and task execution stages. Whereas the other SLS systems for the most part disregard who the user is in the conversation, it is necessary for users to register with Orion and verify who they are before scheduling a task so that Orion is able to contact them at the scheduled time [11].

Lieshou, Orion's Mandarin counterpart, is a recent development at SLS and allows for the exciting potential of incorporating the Orion system into SLLS [2]. Figure 2-5 illustrates a user interacting with Lieshou to schedule a wake up call. First the user has a conversation with Lieshou to register the task. Then Lieshou triggers Muxing to call the user at the appropriate time to complete the task.

2.3.6 Envoice

Envoice [16] is a concatenative synthesis system developed by members of SLS. By carefully designing system responses to ensure consistent intonation contours, Envoice was able to achieve natural sounding speech synthesis with word- and phrase-level concatenation. An efficient search algorithm was devised to perform unit selection given symbolic information by encapsulating class level concatenation and substitu-

Lieshou	User
你好! 我是獵手, 你的自動任務代理. 請告訴我你的用戶名. 如果你還沒注冊, 請說新用戶. (Hello! I am Lieshou, your automatic task manager. Please tell me your user name. If you have not registered, please say "new user".)	褚千慧. (Chian Chuu)
歡迎褚千慧. 你想安排甚麼任務? (Welcome Chian Chuu. What task would you like to schedule?)	請在明天早上七點鐘打電話叫醒我. (Please call me tomorrow morning at 7 to wake me up.)
我打電話時你想讓我告訴你波士頓的天氣嗎? (Do you want me to tell you the weather in Boston when I call you?)	好. (Sure.)
我應該打給甚麼電話號碼? (What number should I call you at?)	請打到家里. (Please call me at home.)
褚千慧你已經提供所有需要的信息。 我會在十一月二十六號星期二上午七點鐘打電話到九二二五八七三四告訴你波士頓的天氣。對嗎 ? (Chian Chuu, you have already provided all the necessary information. I will call you on Tuesday November 26th at 7 am at 92258734 with the weather in Boston. Is this correct?)	對，謝謝 (Yes, thank you.)
我把你的任務要求用電子郵件傳給你了. 您還有問題嗎? (I have sent you an email with your requested task. Is there anything else?)	沒有了, 再見! (Nope. Good bye!)
The next day at 7 am…	
褚千慧, 早上好. 這是木星打電話叫醒你. 波士頓今天, 部分晴天, 最高氣溫七十 華士度左右, 今晚, 多雲, 可能有陣雨, 最低氣溫六十華士 度左右, 百分之五十可能有雨, 您還想知道甚麼? (Chian Chuu, good morning. This is Muxing calling you to wake you up. The weather in Boston is partially clear, highest temperature of around 70 degrees Fahrenheit. Tonight, cloudy with the possibility of some rain. Lowest temperature of around 60 degrees Fahrenheit. 50% chance of rain. Is there anything else you would like to know?)	沒有, 謝謝. (No, thank you)

Figure 2-5: An example of a user interacting with Lieshou. [2]

tion costs, greatly reducing the time used in synthesis.

Envoice is currently deployed in several SLS systems, with language development efforts underway for English, Mandarin and Japanese.

Chapter 3

Usage Scenarios

In this chapter we detail the anticipated experience of students, administrators, and teachers using SLLS to illuminate the functionality of the system.

3.1 Student

SLLS is an open system that welcomes anyone who has the interest in learning a spoken language to log on and begin learning. Here we describe the interaction of a fictitious student, Catherine, and SLLS to showcase how a typical user would work with the system.

Catherine has some rudimentary knowledge of Chinese, having studied it for a semester in college, but now would like to refresh her spoken Mandarin. She is preparing for a trip to Taiwan over Chinese New Year to visit her family. She would like to make sure that she remembers how to talk about relatives, how to ask about common daily things, and how to order food at restaurants.

To begin, Catherine needs to register with SLLS through the web site. She selects Registration from the SLLS web page and proceeds to enter her information. The most important fields here are the phone numbers where she can be reached, and her email and password for logging in later on. However, if necessary, these can all be changed at a later time by navigating to the **Profile** link on the side bar when she is logged in, as depicted in Figure 3-2.

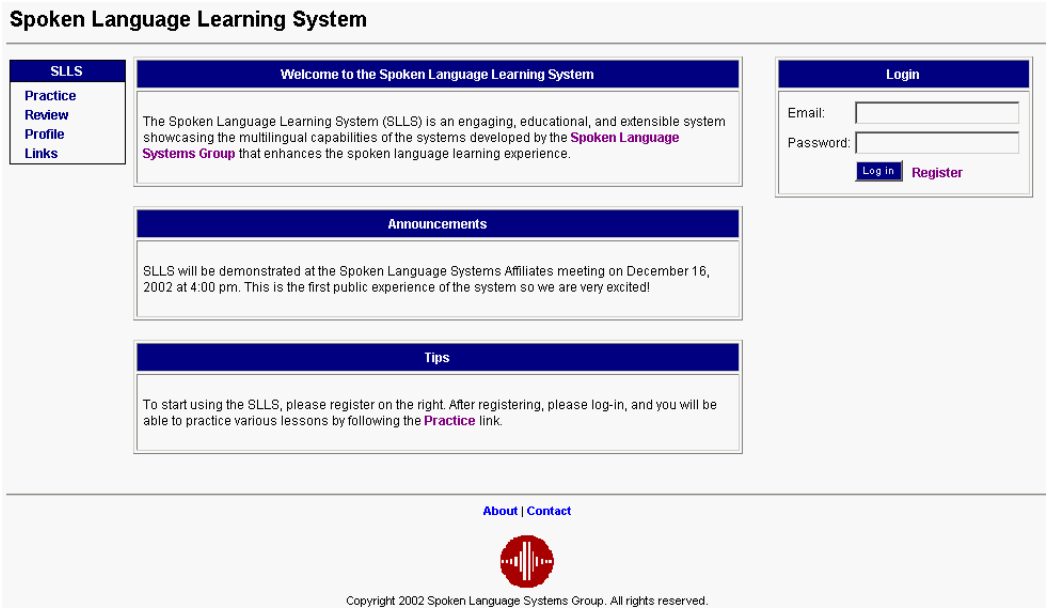
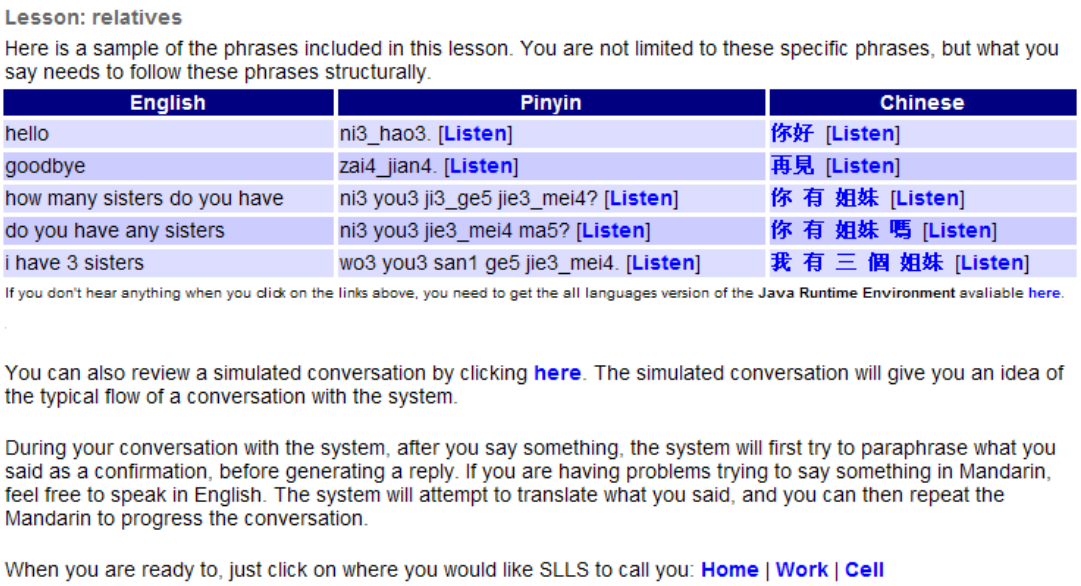


Figure 3-1: SLLS Web site Starting Point

Figure 3-2: SLLS Web site Registration and Profile Editing

Upon registration completion, she is automatically logged in and can navigate to **Practice** where she can view a list of lessons. There are a number of available lessons that have been created by other students and educators that she can choose from, or she can choose to create her own (see *Teacher* for details on lesson creation). Catherine notices that there is already a lesson on relatives created by Dr. Chao, a teacher, and so selects that lesson to practice.



Lesson: relatives

Here is a sample of the phrases included in this lesson. You are not limited to these specific phrases, but what you say needs to follow these phrases structurally.

English	Pinyin	Chinese
hello	ni3_hao3. [Listen]	你好 [Listen]
goodbye	zai4_jian4. [Listen]	再見 [Listen]
how many sisters do you have	ni3 you3 ji3_ge5 jie3_mei4? [Listen]	你 有 姐妹 [Listen]
do you have any sisters	ni3 you3 jie3_mei4 ma5? [Listen]	你 有 姐妹 嗎 [Listen]
i have 3 sisters	wo3 you3 san1 ge5 jie3_mei4. [Listen]	我 有 三 個 姐妹 [Listen]

If you don't hear anything when you click on the links above, you need to get the all languages version of the **Java Runtime Environment** avaliable here.

You can also review a simulated conversation by clicking **here**. The simulated conversation will give you an idea of the typical flow of a conversation with the system.

During your conversation with the system, after you say something, the system will first try to paraphrase what you said as a confirmation, before generating a reply. If you are having problems trying to say something in Mandarin, feel free to speak in English. The system will attempt to translate what you said, and you can then repeat the Mandarin to progress the conversation.

When you are ready to, just click on where you would like SLLS to call you: **Home** | **Work** | **Cell**

Figure 3-3: A lesson about relatives that Catherine can practice

At this point, Catherine can view the phrases that are in the lesson and listen to sample phrases, as illustrated in Figure 3-3. After listening to "how many sisters do you have" and "I have two sisters" in Mandarin a few times, she can also see a simulated conversation of the system interacting with itself. She glances through the simulated conversation to get a feel for how the system operates and the typical conversation flow, and then asks the system to call her on her cell phone by clicking the **Cell** link. She answers her cellular phone when it rings, and proceeds to have a conversation with SLLS. While on the phone, she also has visual feedback from the system through the web site where she is shown what the system has recognized and is trying to say, as shown in Figure 3-5. A sample conversation with the system may proceed as in Figure 3-4.

This conversation highlights a few key features of the current version of the SLLS system. First, although initially SLLS directs the conversation by prompting Catherine with questions, she can at any time choose not to answer the questions and instead take the initiative by asking the system questions. This is shown above when SLLS asks Catherine "do you have any brothers?" in line 2, and instead of answering, Catherine asks the system "do you have any sisters?" as a reply in line 3. In line

	Catherine	SLLS
1.	你好。	你有幾個姐妹？ (How many sisters do you have?)
2.	我有一個姐妹。 (I have one sister.)	你有兄弟嗎？ (Do you have any brothers?)
3.	你有姐妹嗎？ (Do you have any sisters?)	我有兩個姐妹。 (I have two sisters.)
4.	你有。。。 (Do you have…)	對不起，我沒聽懂你說的是什麼。 (I'm sorry, I didn't understand you)
5.	Do you have any brothers?	你有兄弟嗎？ (Do you have any brothers?)
6.	你有兄弟嗎？ (Do you have any brothers?)	我有三個兄弟。 (I have three brothers.)
7.	再見。 (Goodbye)	再見。 (Goodbye)

Figure 3-4: A sample conversation between Catherine and SLLS

4, Catherine fumbled a little with what she was going to say, so the system provides a polite message so that she can try again. Not remembering how to actually ask about brothers, Catherine says "do you have any brothers?" in English in line 5, to which the system then replies with the Mandarin translation. She then asks the system "do you have any brothers?" in Mandarin and she continues the conversation to completion.

Finally, Catherine can review her conversation online, listening to the words and phrases she spoke, and see which are the phrases that she needs work on for her next conversation. Figure 3-6 illustrates the review interface for one of Catherine's conversations. Catherine can click on any of the words to listen to individual words, or listen to the whole phrase. Words that are better spoken are in navy blue, while words that are not well spoken are in red. (The technology to implement the judgement of quality is under development at SLS and beyond the scope of this thesis.) Later on, she can revisit the conversation to view comments and feedback posted by her teacher on her conversation so she can also have expert human advice as well as knowing what the system thought about her conversation.

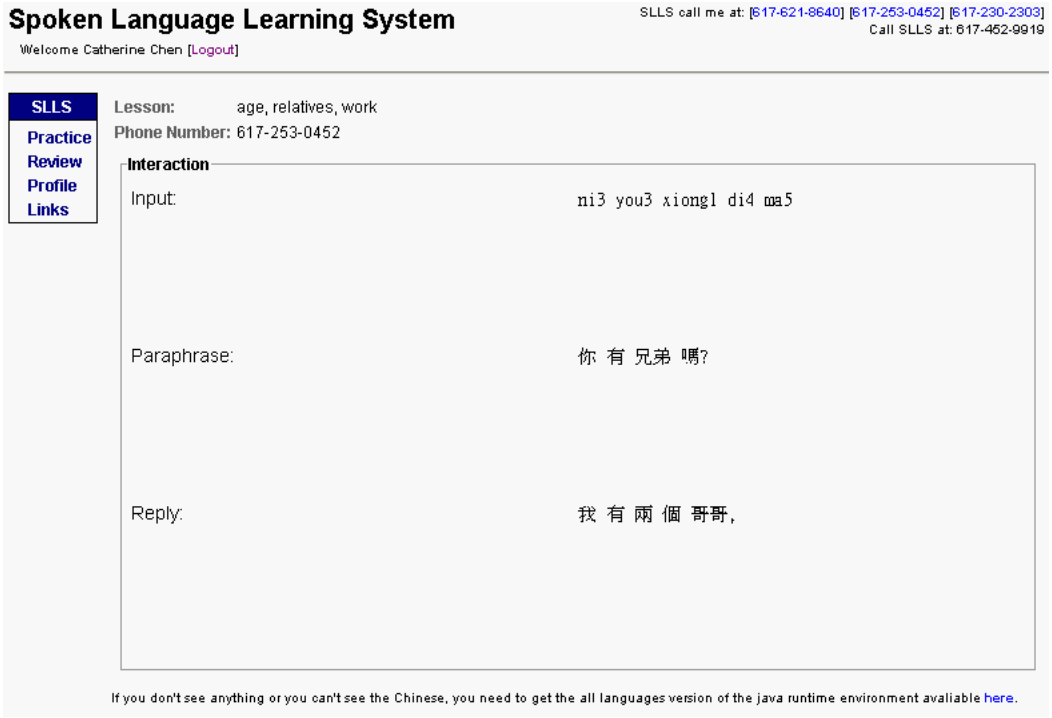


Figure 3-5: Visual feedback from SLLS during the conversation

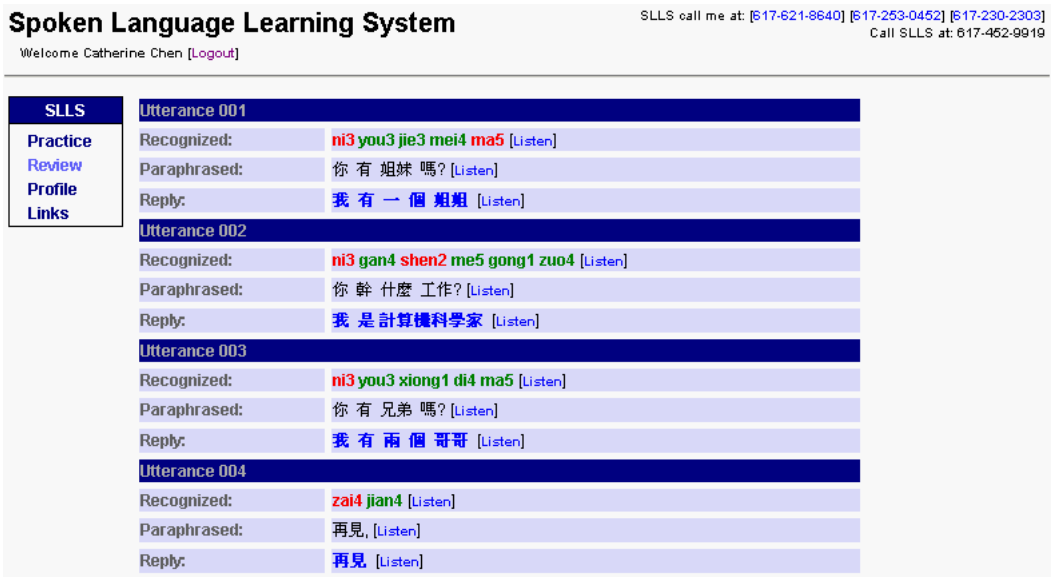


Figure 3-6: The review interface.

3.2 Administrator

Stephanie is an SLLS administrator in charge of maintaining the system and expanding its capabilities. She also wants to know how the system is being used so that she

can better improve it. SLLS provides a great deal of functionality through the Web interface so that common administrative tasks can easily be completed remotely. For example, on the **User Management** web page, Stephanie can add, edit and delete users, as well as manage their groups and permissions. Also, like teachers, she can review the conversations the students have had.

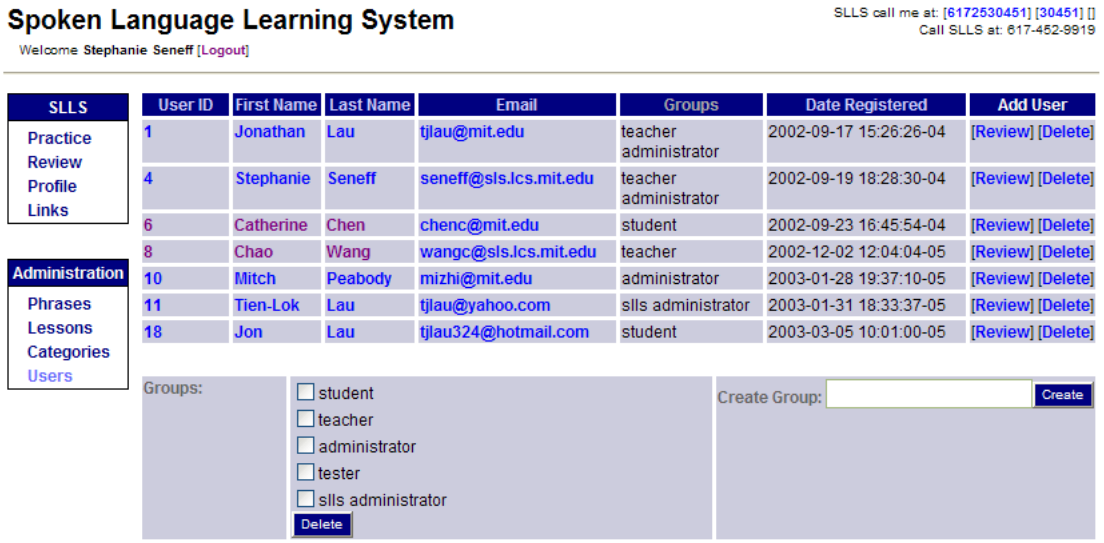


Figure 3-7: User Management Interface

To add new phrases to the system, Stephanie navigates to the **Phrase Management** web page where she inputs the English phrase as well as the key-value parsing of the phrase into the system. Currently, this key-value parsing is a simplified representation of the phrase and is obtained when the system developer augments the recognition and synthesis components to incorporate the new vocabulary. As we discuss in Chapter 7 *Future Work*, this process will hopefully become more transparent to the user as development of the system continues. Once the phrase is entered, it can be added to other lessons and used by students in conversations.

Stephanie can also check on requests made by users, such as adding vocabulary and bug fixes. She can also let her users know the status of those requests by changing the status on the web page and entering a comment. If there are a number of similar requests, she can select all of them and reply to them at the same time. By placing these requests on the web, we reduce the burden of having to deal with email overload

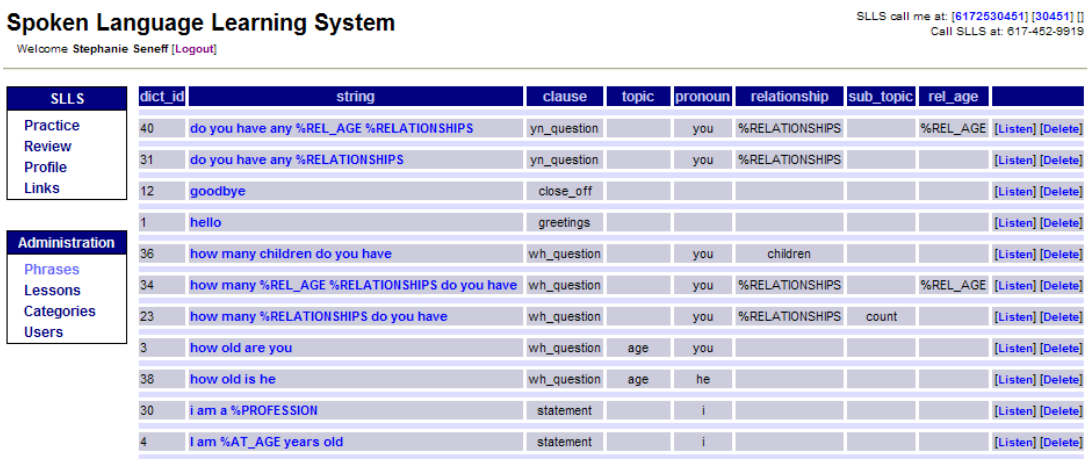


Figure 3-8: Phrase Management Interface

on the part of the administrator, and at the same time, users can have a central repository so they can check if they are making a duplicate request from that of another user. As depicted in Figure 3-9, currently there are two pending bug fixes and one lesson request that has been deferred. Stephanie is about to change the status of the first bug fix to completed because she has just fixed that bug. The user who submitted the bug, Chao Wang in this case, can check the **Requests** page later to see the status change.

3.3 Teacher

Dr. Chao is a Chinese teacher who wants to use SLLS to have her students practice conversations about relatives. She logs on to SLLS to create a lesson for her class where she can either edit a previously created lesson, or create an altogether new lesson. She chooses to create a brand new lesson. She is then shown all the phrase patterns currently available and selects a subset pertaining to relatives. She enters a lesson name and clicks **Create**. Next, for each phrase in the lesson, she specifies what the system response should be if a user says that phrase. For example, Dr. Chao specifies that if the system hears "how many %RELATIONSHIPS do you have," it will respond with "I have %COUNT %RELATIONSHIPS", where a number will

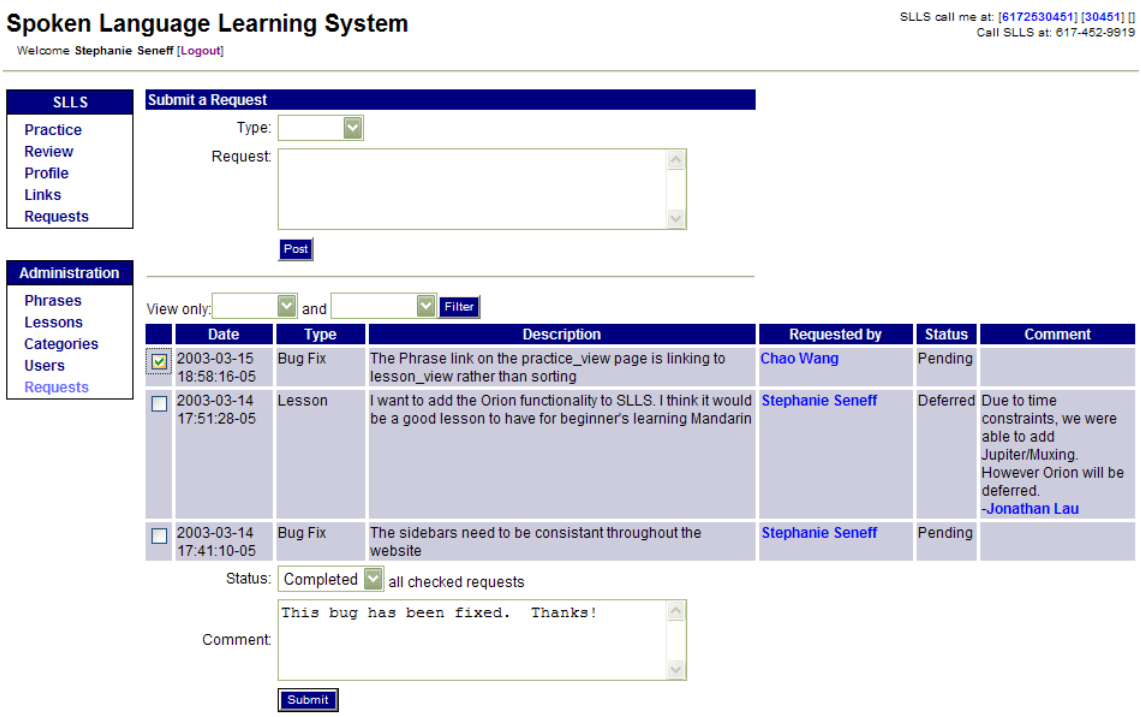


Figure 3-9: Viewing and answering requests through the web site

be specified for the *%COUNT* variable and a relationship such as "brothers" will be specified for the *%RELATIONSHIPS* variable in a process described in detail in Chapter 5 *SLLS Developments*. She also specifies that, whenever the systems hears "I have %COUNT %RELATIONSHIPS" it will choose between asking about another set of relatives, or saying "goodbye". At the same time, Dr. Chao can also add more phrases to the lesson or remove phrases from the lesson. Figure 3-10 is the interface that Dr. Chao would be working with at this point. Once she is happy with the lesson flow, she can try it out by going to Practice and going through the lesson as a student would, iterating the process to perfect the dialogue.

After her students have had a chance to go through the conversation, Dr. Chao can review her students' interactions and leave feedback for them on the web site, helping them improve. In Figure 3-11, Dr. Chao is reviewing a student's conversation. She sees that another teacher has already provided some feedback, and is adding an additional comment to provide the student with more guidance.

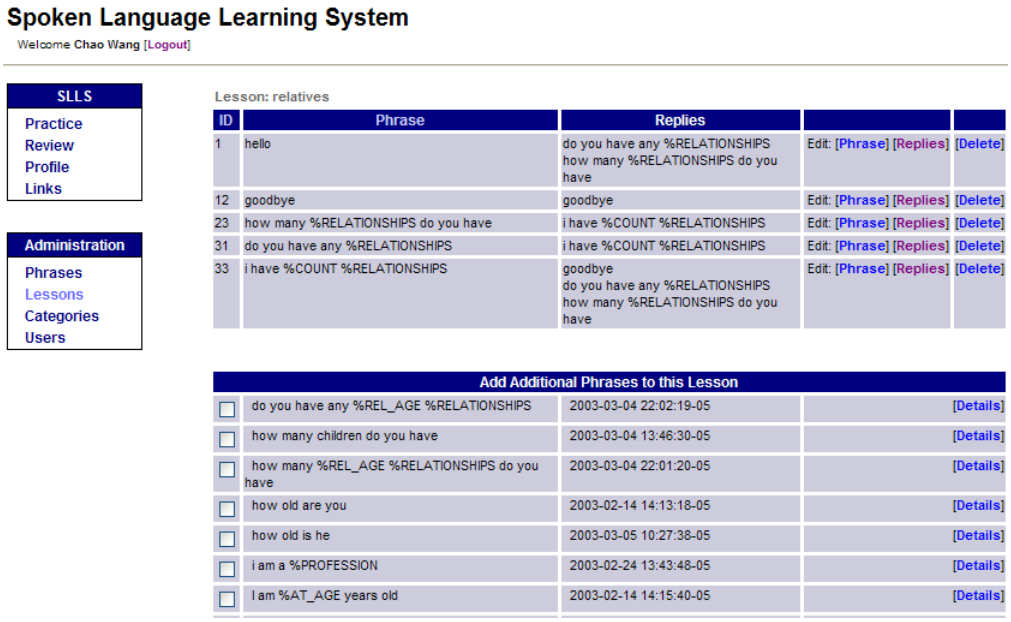


Figure 3-10: Dr. Chao editing her *relatives* lesson

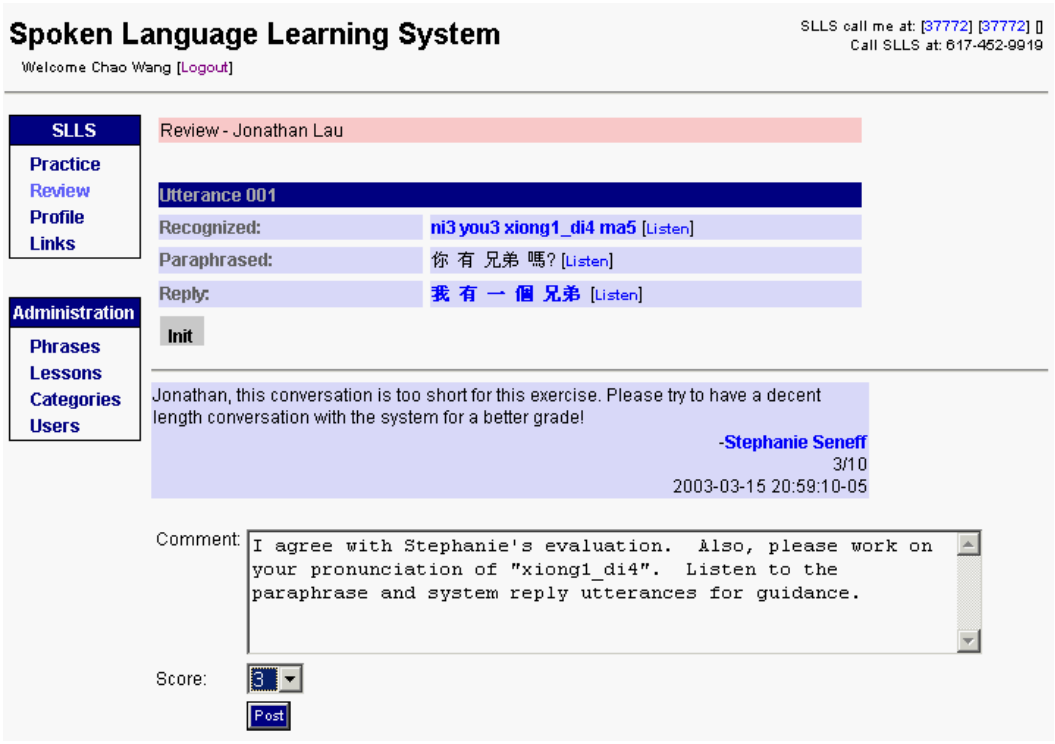


Figure 3-11: Dr. Chao giving some feedback to a user's conversation

Chapter 4

Augmentations to SLS Technologies

One of the main goals of SLLS is to integrate and augment the multilingual offerings of SLS and hence we tried to leverage existing systems whenever possible. Not only is this more efficient because of code reuse, but it also encourages testing and refinement of these technologies. In this chapter we detail how various systems were used and modified to produce the features that were required for SLLS. Depending on how developed the systems were to begin with, some of the systems, such as Jupiter, only required minor additions in order to fit into the SLLS architecture while other systems, such as Phrasebook, required slightly more work.

4.1 Conversant Phrasebook

Starting from the spoken language translation system Phrasebook and its Mandarin counterpart Yishu, we wanted to create a conversational system that would be able to interact with the user dynamically in lesson plans we created. However, we believed that the translation capability was an extremely powerful feature that could be incorporated into the conversational system to provide translation on demand. Therefore, the first thing we had to do was to determine the flow of the user-system interaction. As mentioned when we introduced Phrasebook above, the system produces a

paraphrase before actually performing the translation to provide additional feedback to the user. Keeping this structure, the language flow for Conversant Phrasebook is outlined in Table 4.1. Conversant Phrasebook detects the language the user is speaking and then determines if it will just translate the phrase or make a request to the back-end application to generate a response.

	Conversation	**Translation**
User:	Mandarin	English
Paraphrase:	Mandarin	English
Response:	Mandarin	Mandarin

Table 4.1: Language flow for Conversant Phrasebook

The next step was to determine how to generate responses to user input. As a conversational system, it is important for the responses to be logical replies to the user while attempting to progress the conversation. Since most of the current SLS systems act as information retrieval agents, their conversations are typically more limited in scope and they only need to prompt the user to generate a query in a form the system recognizes. However, as a learning system, we thought it was necessary for the system to be able to both ask and answer questions, which is more similar to normal human conversations. Furthermore, we wanted to ensure that it would be easy for future users to add and modify the responses of the system and this meant that the representation had to be simplistic, flexible and understandable. After much debate, we decided that using key-value pairs and English phrases would be the best representation to fulfill these criteria. When Conversant Phrasebook detects that the user is speaking Mandarin, it will send a request to the SLLS application with the key-value frame for that utterance generated by recognizing and parsing the utterance. The SLLS application will then determine a reply for the utterance through a process described in detail in the *Conversation* section of Chapter 6, in the end, returning an English string to Galaxy. The English string is then passed through the various Galaxy servers to translate it into Mandarin before it is finally synthesized for the user.

It was also necessary to have interfaces for Conversant Phrasebook to SLLS for

calling the user upon receiving a frame from SLLS and sending a frame to SLLS with the location of the log file upon completion of the conversation. To provide visual feedback during the conversation, Conversant Phrasebook also sends the SLLS updates of the recognized, paraphrase and response strings to be displayed to the user. These were all accomplished by passing the Frame Relay server a frame with the relevant information, which is described in detail in Chapter 5.

Finally, it was necessary to train the recognizer with the phrases for our lessons and to ensure that all the vocabulary was covered by the synthesizer for both English and Mandarin.

4.2 Learning Jupiter

Upon the successful creation of our first lesson using Conversant Phrasebook, we decided to ensure that we were able to deliver on the promise of integrating other SLS multilingual systems. Due to timing constraints, we decided to incorporate only one of these systems at the present time. We evaluated the Orion task delegation system and Jupiter weather system on the criteria of usefulness and available functionality, and decided that the Jupiter system was slightly superior at this current stage.

To ensure a consistent user experience, the first augmentation to the Jupiter system was to have it follow the language flow of Conversant Phrasebook described in Table 4.1. This was accomplished by building on the bilingual Jupiter system described in Chapter 2 and adding the spoken paraphrase capability that is absent in the typical Jupiter system. The same interfaces to SLLS that were added to Phrasebook were then needed in Jupiter for it to work with the language learning framework. At this point we already had a working system, but we found that the Mandarin synthesis using Envoice was extremely poor, given that Muxing was in an early development phase. Therefore, as with Conversant Phrasebook, it was necessary for us to work within the Envoice system to improve synthesis quality.

4.3 Envoice

It is absolutely essential to have extremely high quality synthesis in language learning tasks. To ensure that we have control over the continued development of the synthesis system, we decided to use the Envoice system developed at SLS. However, the development of Envoice is beyond the scope of this thesis - please refer to [16] for further details. In this section, we would like to highlight the augmentations that were made to Envoice that are important to the current and future versions of SLLS. First, for users to listen to the synthesized responses at a word level, Envoice had to provide the timing information to SLLS. Since this timing information is already used for the concatenative synthesis, it was not difficult to provide this feature. Second, as mentioned in both the Conversant Phrasebook and Learning Jupiter sections, the coverage of Envoice was extended to cover the vocabulary of those two systems. Chao Wang and Min Tang were kind enough to volunteer their voices for recording sessions, after which their voices were transcribed, aligned and then incorporated into Envoice. Part of the result of this coverage expansion is the availability of two distinct voices for Envoice, and, because of this, Envoice was extended to allow for dynamic switching of the system voice. The implications of this are two fold for SLLS. For the simulated conversations, there is the potential to use the voices to distinguish between the simulated user and the simulated system, and, for conversations, the voices can be used to differentiate between the role of the translation agent and the conversation agent.

4.4 Simulated Conversation

To generate the simulated conversations, we could have either taken the route of modifying the conversational hub scripts or we could have started from the aptly named Batch Mode script that is used to perform batch processing on Galaxy. Batch Mode typically uses utterances from a file as input to a conversational system. The system then responds, and the whole transcript is saved in a log file. Since we did not want

to deal with the issue of simulating a phone call into the system, and we believed that the Batch Mode interaction was easier to augment due to its simplicity, we decided that we would use Batch Mode as the backbone for the Simulated Conversation.

Initially, a welcome frame containing the lesson ID and an initial utterance is passed to the Simulated Conversation system from the SLLS web site via the frame relay to launch a simulated conversation. This frame is passed through the typical chain of Galaxy servers, resulting in the key-value frame that is then passed to the SLLS application. The SLLS application returns an English reply string to Galaxy, which is then processed through the synthesizer and delivered via a local audio server rather than the telephone audio server. The local audio server is a server that creates a wave file and plays the file on the machine that it is running on. The English string is then passed back through the hub script as an input string where it is paraphrased and synthesized by local audio before finally being passed in key-value form to the SLLS application again for a reply. When the SLLS has no more replies for the conversation, this process terminates and a frame is sent to the SLLS application to begin the post-processing of the simulated dialog, which is described in detail in the next chapter. Figure 4-1 illustrates the simulated conversation creation process.

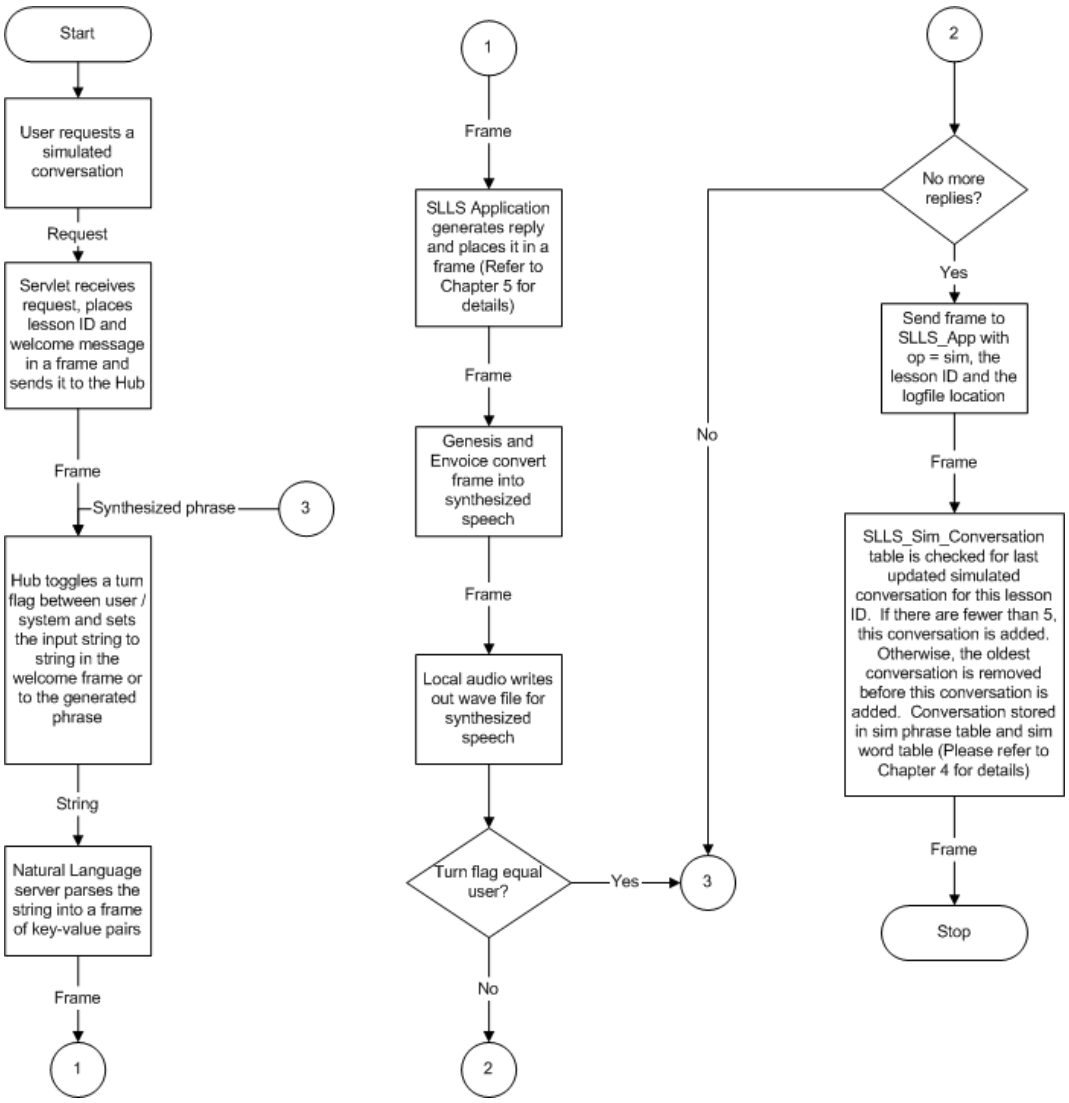


Figure 4-1: Step-by-step description of SLLS during a simulated conversation

Chapter 5

SLLS Developments

The SLLS web site is built using Java Server Pages, Java Applets and Java Servlets. The information for the web site is stored in a PostgreSQL database that is described in detail in the next section. The use of a database ensures persistence of information and allows for multiple users to access the system at the same time. It also allows for much more pre-processing and limits the amount of work that has to be done on the fly for the user every time they log in, since most of the information will already be held in the database. Figure 5-1 describes the high level organization of the SLLS system. In the database model diagrams in the following chapter, PK refers to Primary Key, while FK refers to Foreign Key. Primary keys are unique identifiers used as a means to access instances of the table rows in the database, so that every entry in the table can be referenced just by looking up the PK. Foreign keys are used in our database to ensure referential integrity so that when users delete elements on the web site, all entries related to that element are also removed.

5.1 Administration

In Chapter 3, we outlined scenarios for an administrator and a teacher using the system. Although these features may not seem as important as the three phases of operation described in the previous sections, they are in fact central to the management of SLLS and enable the continued development of the system. The main ad-

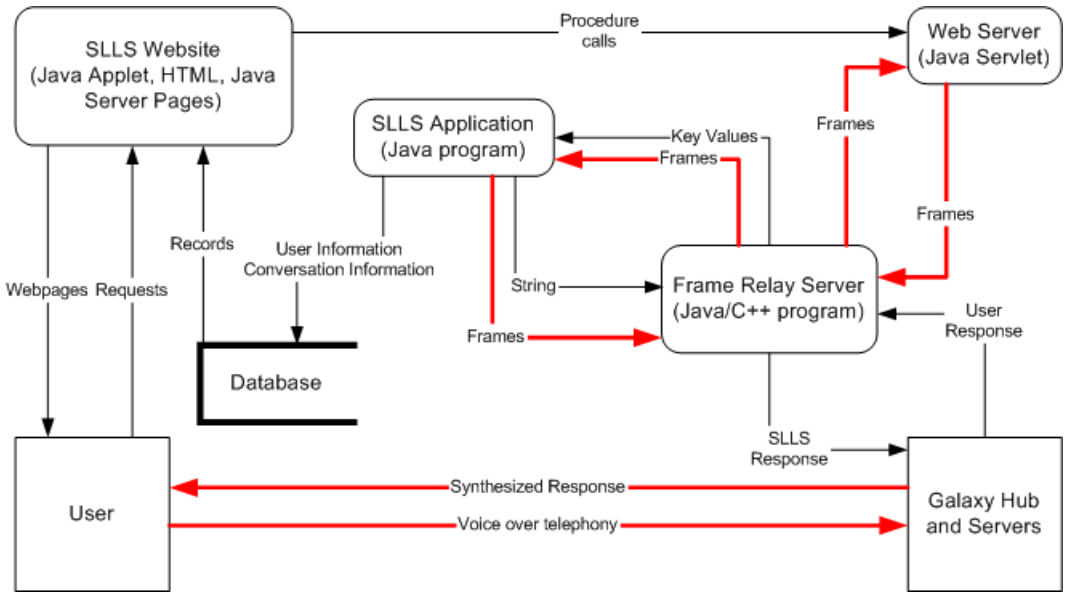


Figure 5-1: SLLS Overview

ministration tasks are: user management, lesson management, phrase management, and category management. The feedback, as mentioned in **Review**, and requests features are paramount to building a sense of community among the users, allowing them to feel connected to and invested in the system.

5.1.1 User Management

Administrators have the ability to add, edit and remove users, add and remove groups, and change the groups a user belongs to. As mentioned previously, all users are placed in the student group by default, and so, for a user to become a teacher or an administrator requires the approval of an administrator.

Each web page on SLLS checks the user's permissions by first checking that the user has logged in, and then taking the user's unique id and checking the groups the user belongs to. If the user is in the administrator or teacher groups, then they are allowed to view the pages associated with these roles. Otherwise, they are only allowed to view the pages for students.

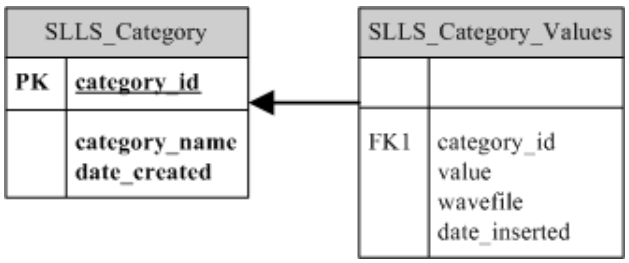


Figure 5-2: Category Tables

5.1.2 Category Management

Categories are an equivalence class of words that can be used interchangeably in a phrase. For example, the category PROFESSION could have elements such as doctor, lawyer, programmer, farmer, college student, etc. When specified by the phrase, SLLS randomly selects an item from these equivalence classes as a representative for the category for the generated spoken output to provide variation. Through this mechanism, we are able to provide a great deal of variability even with a limited number of phrases, increasing the entertainment and educational value of the system.

Users have the ability to add a new category, delete categories, add elements to categories, delete elements from categories, and attach wave files to the elements in the categories. New categories are added to the SLLS_Category table. New elements in the category are then added to the SLLS_Category_Value table with the category id of the category.

5.1.3 Phrase Management

To add new phrases to SLLS requires knowing what the key-value parse of the phrase is, as well as any categories that the phrase will reference. Once these two are known, adding a phrase in SLLS only requires filling in a form on the web site. The form will place the necessary parameters into the database, and that phrase can then be used in any lesson. The key-value parse of the phrase is obtained by running the phrase through the utterance understanding process. This unfortunately still remains a highly manual process because it is necessary to ensure that the recognizer is able

to understand the phrase. We discuss this problem in Chapter 7.

There are two data models that can be used for the conversation dictionary with different benefits and limitations. The most simple and direct approach would be to place all the dictionary items in an SLLS_Dictionary table with columns for every field. By placing all the fields in one table, all database accesses are simplified. This is also more efficient since it would only require one database hit for a data retrieval and insertion. However, the limitation with this model is that it is not as clean for additional fields to be added since it would require the addition of another column in the table. Alternatively, it is possible to make our database more easily extensible through the use of the *skinny data* model. In this model, only the most basic fields of the dictionary are kept in the dictionary table, with any additional attributes kept separately in an SLLS_Attributes table. The attributes table would only have four columns: one to provide a unique identifier for each row, one for the dictionary id that the attribute belongs to, and then a key-value pairing for this attribute. Although by modelling our database in this way, we make it much easier on the database end to add new fields, it actually requires more database hits and more complexity in the code. For each dictionary match, we will have to hit the database as many times as there are attributes for that item plus the database hit on the SLLS_Dictionary table. Moreover, the code required to handle this will be much more complicated, and, to someone just introduced to it, almost unintelligible. The SQL statement itself would require the joining of four different select statements. Therefore, under the assumption that adding new fields will not be a common occurrence, we have chosen to represent the dictionary using one simple table.

The list of categories can be found by navigating to the category page. To use them in a phrase merely requires placing a % sign before the category name. For example, the phrase "How many older brothers do you have?" can be generalized using categories into "How many %REL_AGE %RELATIONSHIPS do you have?", with %REL_AGE and %RELATIONSHIPS being replaced by a random selection from the respective REL_AGE and RELATIONSHIPS category. Hence it is now possible for the system to say "How many older sisters do you have?" and "How

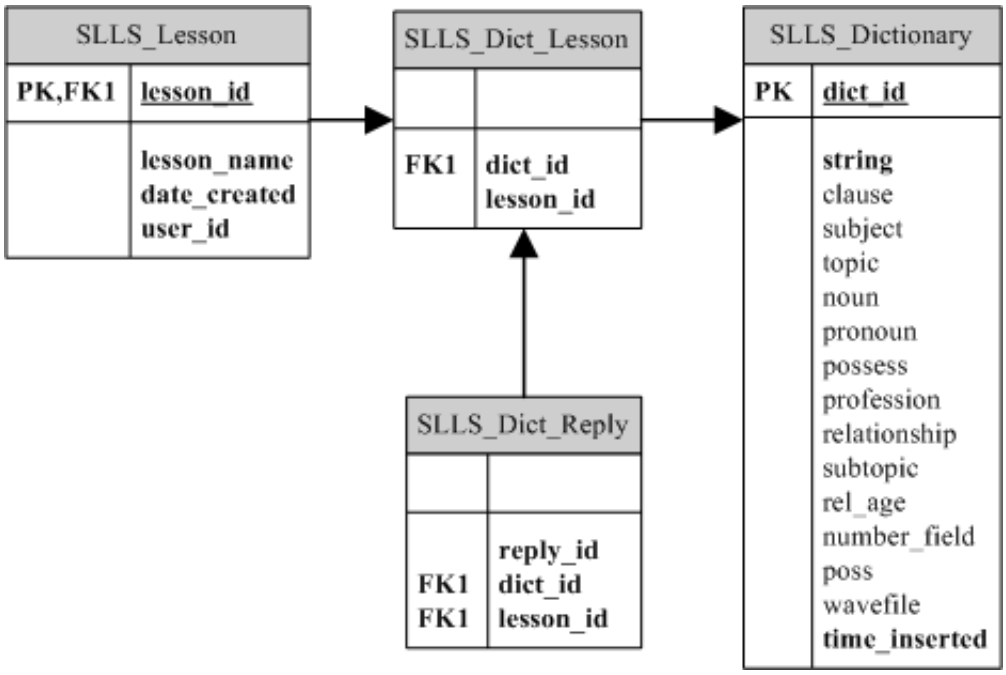


Figure 5-3: Lesson Tables

many younger brothers do you have?" as well as the phrase originally intended by the user.

Users can also edit, and delete the phrases.

5.1.4 Lesson Management

As outlined in Chapter 3, SLLS allows teachers to create their own lessons from the available phrases, providing them with more control over their students' interactions.

When a new lesson is created, an entry is inserted into the SLLS_Lesson table. The lesson ID associated with that entry is then combined with the dictionary ID's of the phrases that are included in the lesson and entered into the SLLS_Dict_Lesson table. For each reply that the user specifies to a phrase, an entry is created in the SLLS_Dict_Reply table with the lesson ID, the dictionary ID of the phrase, and the dictionary ID of the reply. Figure 5-3 is the data model representation of the database for lessons. When phrases are removed from the lesson, or replies altered, the entries in the database are deleted.

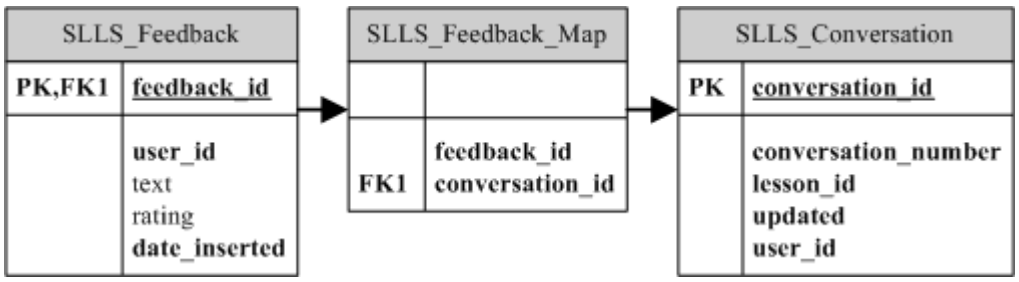


Figure 5-4: Feedback Tables

Users can also edit and remove the lessons at any time.

5.1.5 Feedback

Administrators and teachers have a custom view of the review interface to allow them to browse through user conversations and post feedback. When the feedback is posted, a new entry is created in the SLLS_Feedback table to hold the contents of the feedback. Then the feedback is associated with the specific conversation by adding an entry in the SLLS_Feedback_Map table. The retrieval and display of the feedback is described under *Review*.

5.1.6 Requests

One of the benefits of having a single web interface for students, teachers and administrators is that it is possible to create a centralized resource that facilitates their interaction. The Requests feature allows registered users to ask for features and vocabulary in a central bulletin board, to which administrators can post replies. The users can all see the status of all the requests, and can sort and filter them accordingly. This reduces the redundancy of duplicate requests from users via email, as well as managing the email threads to ensure that users have feedback.

Submitting requests merely requires users to fill in a form on the web site. The values entered in the form are then stored in the SLLS_Requests table. To reply to these requests, the administrators select the requests they wish to respond to by clicking on the check box, selecting the status of the requests, and entering a note.

SLLS_Requests	
PK	request_id
	request_user type text date_requested status fulfilled_user fulfilled_date fulfilled_comment

Figure 5-5: Requests Tables

The entries in the SLLS_Requests table are then updated with those values for the selected requests.

To display these requests with all the pertinent information, we need to join the SLLS_Requests table with the SLLS_Users table twice, once to retrieve the information of the submitter, and once for the replier. In this way, we are able to have only one table for storing requests while still allowing the system to keep track of the user who submitted the request and the administrator who replied to it.

Filtering of the requests is very important because we anticipate that there will be many of these requests which would potentially make it very difficult for users to browse. The filtering is accomplished by passing a "WHERE xyz =" parameter to the SQL query, with xyz as the type or the status of the request.

5.2 Registration

Originally, we had incorporated Lieshou's (Chinese Orion) spoken language registration system into SLLS. Users would have Lieshou call them and attempt to register over the telephone through a Mandarin conversation. Upon completion, they would be given a user identification number that they would then use to complete the registration process on the web site. The *c_first_wave*, *c_last_wave* and *email_wav* fields in the database would be used to store the wave files of the user's first name, last name

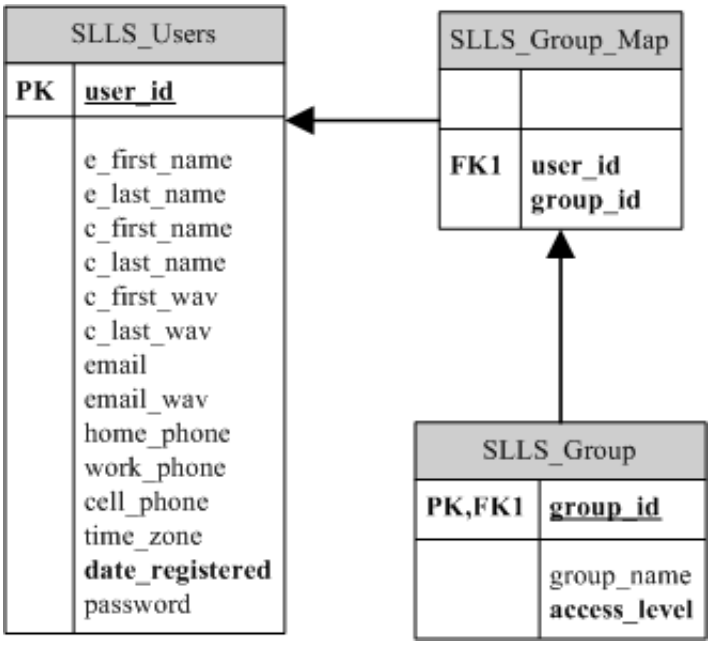


Figure 5-6: User Tables

and email respectively. Although this was a very entertaining and novel approach to registration, we found that, for the average language learner, this was overly cumbersome and in fact quite a disincentive to use the system. Therefore, although this registration method is still available, the primary method to register with SLLS is through the registration form online. When users complete the registration form, they are placed in the student group, giving them permission to all the student features. A new entry is placed in the SLLS_Users table with all the information they entered in the registration form. An entry is also created in the SLLS_Group_Map mapping the user to the student group. The administrators can then change this later as described previously in the *User Management* section. Figure 5-6 shows the database model for user information.

5.3 Preparation

During the preparation stage, we try to introduce the user to the operation of SLLS and to provide them with practice. Since SLLS operates with a unique online-

telephony combination and a conversational system that is unfamiliar to most people, it is imperative that we provide an adequate introduction to the system to ease the user experience. To facilitate this, we have provided the ability to listen to practice phrases and for users to review a simulated conversation with the system.

5.3.1 Playing Wave Files Online

The first step in allowing users to practice phrases of a conversation is to provide them with the facility to listen to the phrases online. There are a number of web sites that currently do this by providing sound files for users to download or stream and have these files played through a third party application such as Real Audio or Microsoft Media Player. However, we thought that it would be better for the user if we could play the sound files natively, removing the need for an external media player.

The main problems with playing files natively is that there are permission and cross-platform compatibility issues. To ensure that rogue web sites do not place viruses and other harmful computer programs on the user's computer, there are strict restrictions placed on writing files on the user's local machine, thereby limiting the ways for us to implement the native sound playing. Microsoft actually offers an *ActiveX* control to play sound online, but unfortunately this is not supported by all the different Internet browsers. Therefore, we resorted to using a combination of technologies that allowed us to offer a cross-platform server-based native listening feature that could be used throughout SLLS.

When the web page is first rendered, all the places where the user can click for a sound will be affiliated with a Javascript command with parameters pulled from the database. For the phrases, the parameter is just the wave file name to be played, while for individual words, the parameters are the wave file name and the timing information generated by *Envoice*. A Java applet is initiated in the background, waiting for the user to click on the Javascript. The click will evoke the Javascript which in turn will call a method in the applet. If the user wants to listen to the whole phrase, the applet will load the wave file directly and play it for the user. On

the other hand, if the user wants to listen to a word, instead of playing the wave file immediately, the applet passes these parameters to a servlet to perform cropping. In order to leverage the applet's capability to play entire wave files, we simulate playing segments of the wave file by having the servlet write another wave file from the start boundary time to the end boundary time. The applet is then passed this wave file and proceeds to play it back to the user. Through this process, we are able to provide a native word- and phrase- level playback mechanism for users to listen to wave files online, which is used by the *Dynamic Practice*, *Simulated Conversations*, and *System Feedback*.

5.3.2 Simulated Conversations

Simulated conversations in SLLS integrate the frameworks developed for conversations and reviewing with the Simulated Conversation Galaxy system. As described in the previous chapter, simulated conversations are created by passing a seed to the Simulated Conversation system, which uses the mechanism described in detail in the *Conversation* section below to generate a reply. When there are no more replies, the simulated conversation is stored in the database in a hierarchical structure similar to user conversations (Figure 5-7). These conversations are then displayed to the user through the same mechanism as the user conversations in the *Review* phase with playback of wave files provided by the process described above in *Playing Wave Files Online*.

Although for the most part, the simulated conversations are analogous to user conversations, except that the system plays both roles, there is one key issue that complicates the simulated conversations, which is the need to have fresh simulations without introducing a delay in the user experience. As mentioned before, variability is achieved through equivalence classes of categories, and hence proceeding through a conversation will probabilistically produce different results each time. Therefore, to generate varying conversations, we only need to run the Simulated Conversation system every time. Unfortunately, because the operation of the Simulated Conversation system is similar to having a real conversation with the system, there is a long delay

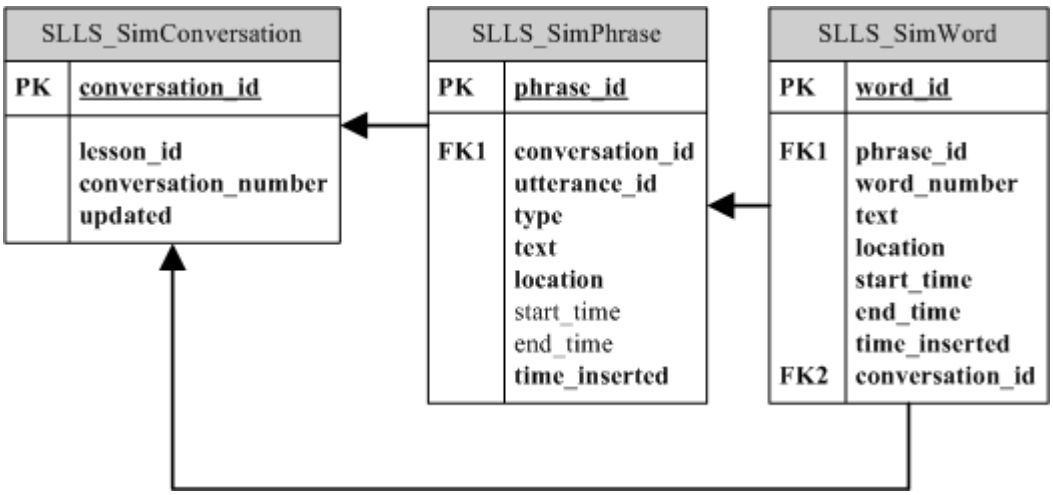


Figure 5-7: Simulated Conversation Tables

during the actual simulation for the conversation to complete. This is an unsatisfactory experience for the user waiting for a web page to load. To remedy this situation, we have created a cache of a number of simulated conversations that are stored in the database for each lesson. This cache, which is updated every time a user opts to view a simulated conversation, is depicted in Figure 5-7. While the user is shown the newest simulated conversation in the database for that lesson, a new simulated conversation for that lesson is requested. When the simulation is complete, this new conversation replaces the oldest conversation in the cache, creating a repository of newly generated conversations without the user experiencing any delay. The next time a user requests a simulated conversation, the process is repeated.

5.3.3 Dynamic Practice

There were a number of approaches we considered for the framework to provide practice, such as allowing multiple waveforms per phrase, allowing a single waveform per phrase, allowing multiple instances of the phrase each affiliated with a separate waveform, and dynamically generating practice phrases and waveforms. However, given the desire to reduce the burden of administrators and to make the interface as simplistic and manageable as possible, as well as the capabilities of the *Simulated*

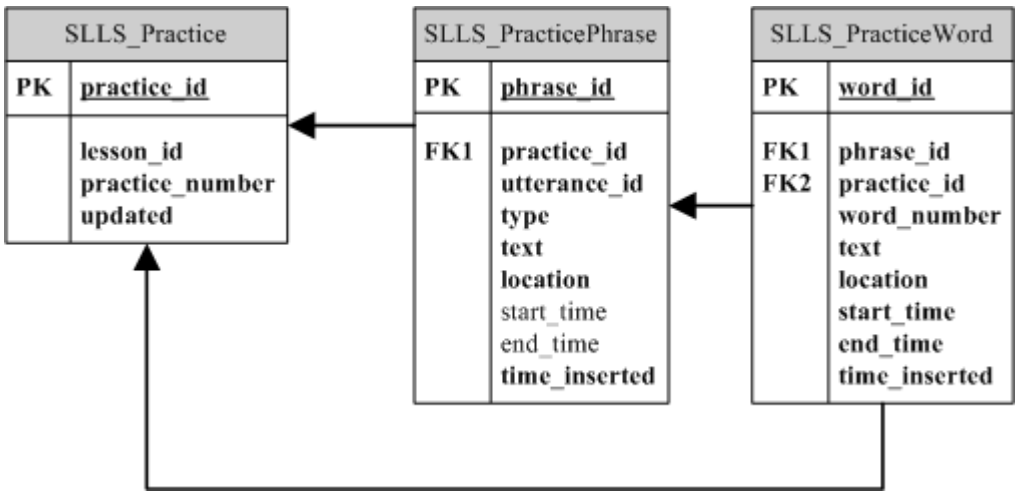


Figure 5-8: Practice Tables

Conversations, we have chosen to have dynamically generated practice.

A mechanism similar to the caching employed by the *Simulated Conversation* is used to produce the dynamic practice. When a user requests practice phrases, a frame is sent to the Simulated Conversation system with the lesson ID and instructions to generate a practice set, which is then passed to the SLLS application. The application retrieves all the phrases in that lesson and proceeds to replace the equivalence class placeholders with an element from the respective equivalence class, producing a list of English phrases. These phrases are then sent back to the Simulated Conversation system one by one for translation and synthesis. Once all the phrases have been synthesized, the phrases and the wave files are stored in the practice database tables as depicted in Figure 5-8. Meanwhile, the user is shown the newest set of practice phrases from the practice cache. When a user requests practice again, the process is repeated, providing variability to the user without burdening the administrators.

5.4 Interaction

Once the user is comfortable with the operation of the system and the phrases that will be used in the lesson, they will proceed into the interaction phase of the system operation. During this phase, the user will have either Conversant Phrasebook or

Learning Jupiter call them at their specified phone number and then engage in a Mandarin conversation with the system. With the Conversant Phrasebook system, the recognition and synthesis are handled by the augmented Phrasebook, while the SLLS application handles the dialogue management. The lesson the user has selected is key to the response generation since, as mentioned in the Lesson Management section, the administrator or teacher selects the possible replies to a user input. When the conversation is complete, the application will then process the log file to store the interaction, setting the stage for the Review phase.

5.4.1 Initiation

When the user clicks on one of the links for the system to call them, they are directed to a page with a Java applet. The applet sends a request to a Java servlet with the session ID, user ID, lesson name, telephone number and the action parameter for post-processing. The servlet then places these parameters in a frame and sends it off to the hub via the Frame Relay. The hub saves these parameters, and then requests the telephony server to call the user at the phone number.

5.4.2 Conversation

There are two SLLS developments for conversations: the online real-time textual display of the conversation proceedings for visual feedback and the dialogue management for reply generation. In this section, we outline how these two tasks were achieved.

Real-time Textual Interface

During the conversation with the system, it is beneficial for purposes of visual feedback and language learning to display the textual representation of the spoken phrases. Users can then combine what they say and hear with what they see on the screen, enforcing the language learning process. This seemingly simple task of displaying the status of the conversation is complicated by the request-response model of Internet protocols and the asynchronous operation of Galaxy. Had it been possible to produce

an HTTP request from the web page directly to Galaxy and have Galaxy respond with an HTTP reply, this process would be greatly simplified. Unfortunately, this capability was not available, and hence a more convoluted scheme was developed to produce the desired result.

After the initializing applet requests SLLS to call the user, it will begin polling the servlet in two second intervals for the input string, paraphrase string and reply string using the session ID as the key to uniquely identify the user's browser. During the conversation, whenever an input string, paraphrase string or reply string is generated, the hub sends, via the Frame Relay, a frame to the servlet with the string and the saved session ID from the initiation. The servlet then stores these parameters into hashtables. Each time the servlet is polled, it looks in its hashtables to see if data for that session ID exists, and if so, returns that data to the applet. In essence, to overcome the constraint of having incompatible system operation, we created an intermediate cache to act as the facilitator for the two sides and allow data to traverse through the system.

Dialogue Management

When a frame is passed to the SLLS application requesting a reply from Conversant Phrasebook, all the key-value parameters are extracted from the frame and used to look up the corresponding entry in the SLLS_Dictionary table, which returns a dictionary ID. Combining this dictionary ID with the lesson ID, we query the SLLS_Dict_Reply table to find all the replies that were specified by the teachers and administrators for this particular phrase in this particular lesson. If there are more than one reply, a reply will be selected at random from the choices. At this point, we have the reply as an English string with the possibility of categories embedded in it for variation. This string is parsed for the categories and a random element from the category is selected and substituted for the category name, resulting in an understandable English sentence that is returned to Conversant Phrasebook.

For example, if the input phrase is "what do you do for a living", the key-value frame would have the parameters $clause : wh\_question; pronoun : you; topic : profession$.

Using these parameters, the dictionary ID is looked up in the SLLS_Dictionary table and found to be 20. The lesson ID for the conversation is also extracted from the frame, which in this case is 3. Then from the SLLS_Dict_Reply table, entries that have a dictionary ID for 20 and a lesson ID for 3 are selected. In this case, there are two entries, with dictionary ID's 15 and 24. Entry 15 is randomly selected, which, after looking up in the SLLS_Dictionary table, turns out to be "I am a %PROFESSION." This string is parsed, and the %PROFESSION tag indicates that the elements in the PROFESSION category should be retrieved from the SLLS_Category_Values table. Again if there are multiple entries, one is randomly selected to replace the category name, for example "doctor". Finally, the reply "I am a doctor" is returned to Conversant Phrasebook for translation and synthesis.

We have designed the dialogue management system to be as flexible as possible and tried to incorporate variability at multiple points to decrease the repetitiveness of using the system. Teachers and administrators have the power to sculpt the lesson plans as they see fit, and any changes they make can be immediately experienced in conversations. As the vocabulary of SLLS gradually increases, with almost no effort, the variability of the conversations will also increase.

Figure 5-9 summarizes the entire conversation process.

5.4.3 Post-Processing

When the telephony server detects that the user has hung up the phone, a frame is sent to the SLLS application via the Frame Relay with the location of the log file, the user's ID, the lesson, and the type of conversation it was. Based on the type of conversation, for the moment either Conversant Phrasebook or Learning Jupiter, the SLLS application proceeds to parse the log file for various parameters and values to populate the database. To store all the information in the database, a hierarchical structure of conversation, phrase and word is used, as depicted in Figure 5-10, that is analogous to the structure for simulated conversations. First, a new entry is placed in the SLLS_Conversation table for the whole conversation and the assigned conversation ID is saved. Then, for each phrase, an entry is placed in the SLLS_Phrase table with

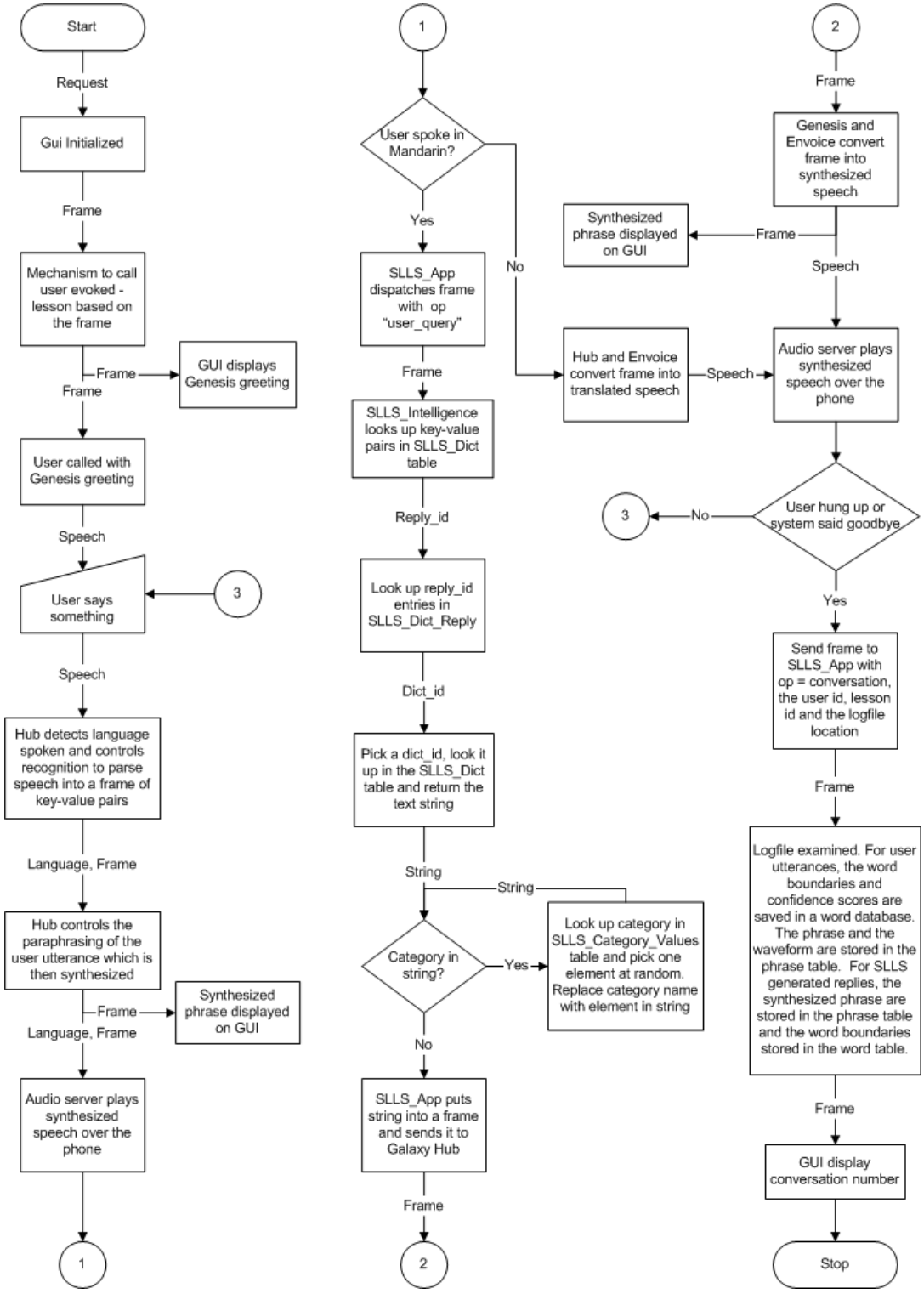


Figure 5-9: Step-by-step description of SLLS during a conversation

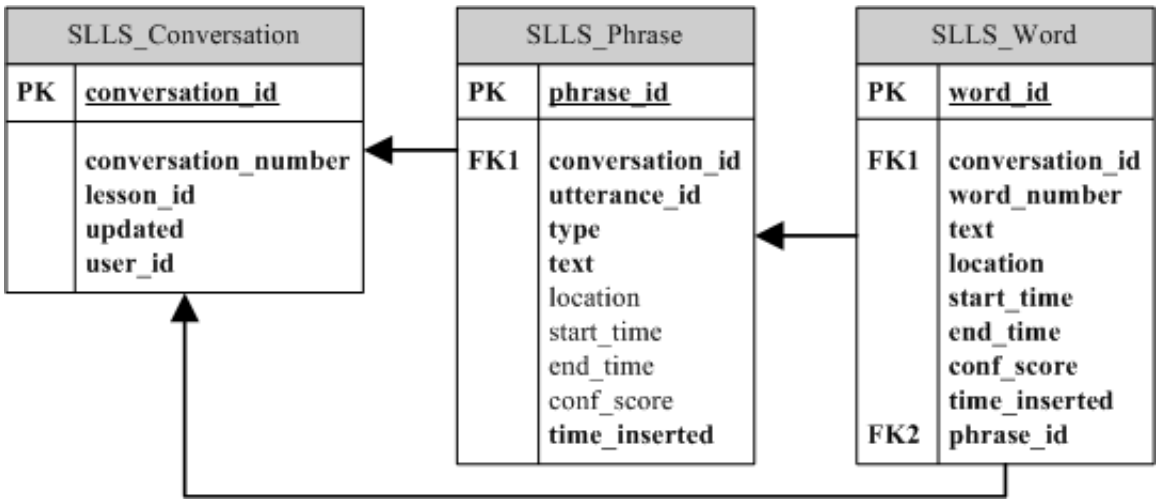


Figure 5-10: The Database Hierarchy for Storing Conversations

the conversation ID and all the phrase parameters, and the phrase ID is saved. Finally, for all the phrases that have word level breakdowns (namely the input phrase and the reply phrase), for each word in the phrase, a new entry in the SLLS_Word table is created with the phrase ID. This structure will be vital to the reconstruction of the conversation in the *Review* stage described later. Once the log file parsing is complete, the SLLS application sends a frame with the number of the conversation for the user to the servlet, and the polling applet will display this information to the user.

5.5 Review

Review is the last of the three phases of user operation. In this stage, users will have audio and visual feedback regarding their conversation, allowing them to assess their strengths and weakness, and highlighting areas for improvement. What follows is a description of how the review interface is generated.

5.5.1 System Feedback

When the user navigates to the review web page, the conversation ID is used to retrieve the conversation parameters from the SLLS_Conversation table. Then, using the conversation ID, the phrases for the conversation are retrieved from the SLLS_Phrase table. These phrases are sorted by type (input, paraphrase, reply, translation) and utterance ID, so that it follows the order of the actual conversation. For the input phrases, the phrase ID is used to retrieve the words of the phrase from the SLLS_Word table. The words are sorted by word number, and then, for each word, a Javascript link with the start and end boundaries of the word in the wave file is generated. The scores of the words are used to determine the color of the text - navy blue symbolizes a high score, and red symbolizes a low score, with varying degrees in between. At the end of each phrase, the system creates a link to play the whole waveform at once. A similar process is used to display the paraphrase and the replies, although these do not have scores associated with them. In this manner, we are able to present the textual proceedings of the conversation to the user.

To provide the capability for users to listen to the proceedings, we employ the same mechanism as described in the *Playing Wave Files Online*.

5.5.2 Human Feedback

At the end of the Review interface is an area for teachers and administrators to leave feedback for the user. To display this feedback, we join the SLLS_Feedback_Map, the SLLS_Feedback, and the SLLS_Users tables and select only the feedback entries in the SLLS_Feedback table that pertain to the particular conversation. Joining these three tables allow us to display the name and email of the user who posted the feedback, as well as the feedback itself, and this model allows multiple feedback from multiple users. This human feedback is especially important early on when the system feedback is still under development. Furthermore, it allows teachers to interact with students on a more personal level, increasing the feeling of connectedness for users.

Chapter 6

Evaluation

The optimal way to evaluate a spoken language learning system is by having a wide spectrum of people use the site and measure their progress and satisfaction through some external metric. This can be accomplished by leveraging the language classes at MIT and designing lesson plans in SLLS tailored to fit the introductory classes. We have spoken to various teachers in the Chinese department here at MIT who are interested in learning more about the system. Unfortunately, given the time constraints with this thesis, we were unable to undertake such a large scale project at this time. Instead, we will evaluate SLLS on its success in fulfilling the goals outlined in Chapter 1 by assessing user satisfaction through a survey of a small group of users.

Three individuals were asked to test the operation of SLLS. They all have no experience with spoken language systems, moderate experience with Internet technologies and have varying degrees of Mandarin proficiency. Table 6.1 summarizes the user profiles. We selected these people because we believe that they will be representative of the users of SLLS going forward. The general populace has little experience with conversational systems, but because of the widespread usage of the Internet, most people have some exposure to it. We had an administrator create two lessons for them, one using Conversant Phrasebook and the other using Learning Jupiter. We then set them loose on the web site with the goal of partaking in those two lessons. Below we discuss how successful the users were in the various tasks, and offer their comments, complaints and suggestions.

User	**1**	**2**	**3**
Mandarin Proficiency:	None	Beginner	Conversational
Internet Proficiency:	Knowledgable	Adept	Knowledgeable
Conversational Systems Experience:	None	None	None

Table 6.1: Profile of the three testers of SLLS

6.1 Experience

As a user without any experience with Mandarin, User 1 thought the system was very unforgiving. Although he was able to listen to the practice and the simulated conversation, he still felt unprepared for the actual conversation. The translation capability during the conversation was instrumental in reducing his frustration and allowing him to at least progress through a limited conversation. Yet even when he was able to proceed with the conversation, he was unable to consistently comprehend the system responses, again due to lack of familiarity. This was particularly prevalent in the conversation with Learning Jupiter, since, even though he was able to repeat the translation from the system, he was unable to comprehend the weather information, hence whether the weather was live or not was of no consequence to him. Further interactions with the system only increased his level of frustration, although he felt some degree of accomplishment when the system was able to understand him, and he was able to make educated guesses as to the meaning of the replies. Moreover, he felt that the reviewing process was very useful, and it was very helpful for him to be able to hear his own voice and compare that with the system. It was not very encouraging for him to see red text (signifying low confidence) when reviewing the conversations, but as he progressed, he had some visual queues that showed that he was improving.

For User 2, having some knowledge of Mandarin greatly increased her level of comfort with the system. After engaging in three conversations, she had a grasp of the system capabilities and hence enjoyed relatively smooth experience. Rather than focusing on getting the system to operate correctly, User 2 found herself engaging in numerous conversations with the system in order to improve the confidence scores on the review page. Much of her time was spent listening to the words that had a

low score and comparing her pronunciation with that of the system paraphrase. One approach she used was to repeat the translated phrase from the system during the conversation. For each phrase, she would first say the phrase in English, listen to the translated response from the system, and then emulate the Mandarin to proceed with the conversation. Although she did show some improvement over time, it was difficult for her to make the pronunciation changes indicated by listening to the paraphrase. She felt that she needed some more advice, either generated by the system or through a teacher to help her further diagnose her pronunciation. Unfortunately, even though the functionality for teachers to post feedback about student conversations was in place, during the trials we did not have teachers on hand to provide the feedback. Finally, she found the conversation with Learning Jupiter especially exciting because it was live information that she could verify. However, the synthesized replies were not quite as eloquent as those from Conversant Phrasebook, and she did not feel that she would be comfortable describing the weather to someone, especially since the Learning Jupiter system only allows users to query for weather and not speak about it to the system.

The main problem User 3 had with the system was that his expectations were too high. The preparation phase was able to provide him a good idea of what the system would say and what it expected in return, and so, for the first few conversations, he was able to progress smoothly. However, having gotten used to the system understanding him, he started pushing the limits of the system by straying from the phrases in the practice phase. Figure 6-1 depicts one of his conversations with SLLS. Even though the phrases he spoke were similar in nature to the phrases in the lesson plan, because they were structurally different, the system was unable to comprehend his speech. This was quite a disappointment for him, as he expected that the system would have been able to respond to more realistic user speech than is currently supported. For example, when asked what sports he likes, he responded with a list of sports, which is a typical answer to such a question. The system however, is unequipped to handle such a phrase, and hence was unable to respond. Given that the goal of SLLS is to create a learning environment that simulates real world

	User 3	SLLS
1.	你好。 (Hello)	你幾歲? (How old are you?)
2.	我二十二歲。 (I am twenty-two years old.)	你有妹妹嗎? (Do you have any younger sisters?)
3.	沒有。 (No.)	對不起，我沒聽懂你說的是什麼。 (I'm sorry, I didn't understand you.)
4.	我沒有妹妹。 (I don't have any younger sisters.)	你幹什麼工作? (What do you do for a living?)
5.	我不喜歡工作。 (I don't like to work.)	對不起，我沒聽懂你說的是什麼。 (I'm sorry, I didn't understand you.)
6.	我是大學生。 (I am a college student.)	你喜歡哪些運動? (What sports do you like?)
7.	我喜歡游泳，籃球和網球。 (I like swimming, basketball and tennis.)	對不起，我沒聽懂你說的是什麼。 (I'm sorry, I didn't understand you.)
8.	我喜歡籃球。 (I like basketball.)	再見。 (Goodbye)

Figure 6-1: The proceedings of one of the conversations User 3 had with SLLS

conversations, he felt that expanding the "knowledge" of the system is paramount in making the system practical. Another component he would like to see is for the system to have some semblance of memory. Currently, the system incorporates randomness in its responses to provide variability to the user. However, for prolonged conversations, there is the opportunity for the system to repeat itself, or even say contradictory things. For example, he had one conversation where he keep asking the system how many brothers it had. The system responded the first time with "I have three brothers", the next time with "I have one brother", and then went back to "I have three brothers" again. He appreciated the variability the system provided, but thought that perhaps this might be confusing to beginners using the system, while at the same time making the system seem "lost". For the most part however, User 3 thought that the system was "exciting, and could really become something very useful for beginners".

6.2 Analysis

Although feedback from the users indicated a number of issues that should be addressed in the near term to improve SLLS for the next round of testing, it was overall quite positive. At the end of testing, the three users were still very excited about the future of the system, and felt that their spoken Mandarin ability had improved even in the limited interaction they had. In this section, we discuss the issues and outline possible steps that could be taken to remedy them.

6.2.1 Users With No Mandarin Experience

Users with no Mandarin experience are one of the groups of people SLLS is trying to help. Unfortunately, as shown by User 1's frustrating experience with the system, there is still some work that needs to be done before SLLS is equipped to serve these users. It is probably impossible for any system to truly make spoken language learning effortless, given that a considerable amount of responsibility for the preparation lies in the learner himself. However, one thing that could perhaps help these users is an interaction mode with the user that prompts repetition. The system could have a list of phrases and then speak the phrases first in English, and then in Mandarin. The user would then have to repeat the Mandarin. If the system is unable to recognize the user's speech, it would repeat the phrase again. This process would run until the system ran out of phrases or until the user hangs up. The system would then process the log file and generate a review interface that has the scoring indicators and allows the user to hear both the system and user utterances. This mode would be similar to a language tutor having you repeat sentences until he feels that you have the correct pronunciation, and would definitely help prepare users with no experience for conversations with SLLS.

6.2.2 Learning Jupiter Limitation

The ability to incorporate Jupiter into SLLS was one of the main goals of this thesis. User feedback has been very positive in terms of having access to a live information

system, and it is definitely a triumph to integrate SLS's multilingual offerings. However, User 2 pointed out a limitation with Learning Jupiter that is not present in Conversant Phrasebook. Whereas users can partake in both the question and the answer roles in the conversation with Conversant Phrasebook, in Learning Jupiter, users are limited to asking questions about the weather. This limits the usefulness of the lesson for the user since they are unable to practice giving the weather. A solution to this is to have two different lessons on weather, one with Learning Jupiter and one with Conversant Phrasebook. The lesson with Conversant Phrasebook would randomly generate weather information when asked, and would be able to randomly generate questions regarding weather to prompt the user for weather forecasts. Then for live information, users will engage in the lesson with Learning Jupiter. Another interesting configuration, where a user interacts with two different agents, is also feasible. One voice would provide weather information and a second one would ask about it. The user would then feel that they were communicating the information from one computer agent to another, testing both the users listening comprehension and speech.

6.2.3 Develop Intelligence

For users who have no experience with conversational systems, once the conversation starts going smoothly, it is very natural for them to imagine that the system is "intelligent" and begin to speak to it like a person. As User 3 has shown, users will undoubtedly utter phrases that are either too simple to too complex for the system during the conversation, and the fact that the system is unable to respond results in a disappointing user experience. User 3 has also shown that providing the system with some commonsense may help users' understanding, since in every day interactions, commonsense underlies all interpersonal communication. With commonsense, the system would not constantly change the number of brothers it has, nor would it say that it is a 3 year old doctor. Unfortunately, like the problem of effortless language learning, incorporating true commonsense into a system at this point in time is also quite impossible. What can be done however is to couple a simple form of memory

that will allow SLLS to remember what it said with a number of constraint rules that will limit groups of non-sensical phrases. Although not a perfect mechanism for providing intelligence in a system, for the purposes of a spoken language learning system, it should be more than adequate.

Chapter 7

Future Work

We have established an infrastructure and demonstrated the feasibility and potential of the SLLS. We see this first iteration of SLLS as a launch pad for a whole gamut of tools and features to empower users and developers of multilingual systems by providing a practical medium for learning. Some of the tasks ahead include incorporating more of the systems at SLS, developing a collection of lesson plans, improving the performance of the underlying language technology components, launching a data collection effort in Mandarin classes, developing software toolkits for non-experts, developing appropriate grading schema, and integrating multi-modal interaction, including graphical interfaces and audio/visual synthesis.

7.1 Extending Language Learning Experience

SLLS currently focuses entirely on the spoken language learning experience to supplement the overall language learning process. As the system becomes more developed, it will be helpful to users for SLLS to incorporate more in-depth practice and preparation to become a one-stop language learning destination. We could take a cue from the online language learning systems described in Chapter 2 and provide grammar tutorials, vocabulary lists, and reading and writing assistance. When SLLS launches in language classes, this information can be obtained through interactions with the teachers and the textbooks to better tailor the system to the class. However, before

such ambitious projects are undertaken, it is imperative for the underlying technology to improve to ensure a satisfactory user experience. Although this first version of SLLS is functional, our user testing has shown that a great deal of work needs to be done for real use.

7.1.1 Develop New Lesson Plans

To actually make it so that SLLS will be usable in a classroom setting, we will have to continue to develop new lessons tailored to the teachers. Based on their requirements, we will then augment the system to cover those phrases, and allow the teachers to sculpt their own lesson plans. The *Request* feature described in Chapter 4 was created to facilitate this exchange. Some other obvious future lessons would be based on the other multilingual systems at SLS such as Orion. We are currently in discussion with various people at the Foreign Language Department at MIT for possible joint development efforts to bring SLLS into the classroom.

7.1.2 Improve Performance of Underlying Technology

The two key limitations of the current SLLS are the recognition of the user's utterances and the synthesis of system generated responses. Although we have been able to leverage the best practices of SLS in developing SLLS, our evaluation has shown that improvements in the recognition and synthesis components are still necessary for the real use of SLLS. Below we outline the main hurdles to these two components for future work.

Recognition

As SLLS is a spoken language learning system, our target audience is non-native speakers of a foreign language who will have trouble with pronunciation and grammar, as well as tone, pitch and accent. However, speech recognition systems are typically imperfect even for native speakers of the language, let alone for the task of recognizing non-native speech. Without the ability to recognize non-native speech, SLLS will be

extremely limited in its usefulness, so it will be vital to continue research in this area. One such improvement that has been developed at SLS is the scoping down of the recognition vocabulary while loosening the recognition constraints, thereby providing better recognition for a smaller set of utterances at the expense of generality. This approach is especially applicable to SLLS due to the lesson plan capability of the system. In the future, when a user selects a lesson plan, a recognizer sculpted to the lesson could be created on the fly for the conversation, which would then be more tolerant to non-native speakers.

Synthesis

Synthesis is vital in spoken language education because synthesized speech acts as a model for users to emulate. The use of Envoice for synthesis in SLLS allows SLS to incrementally improve the generated speech by improving the capabilities of the Envoice system. As mentioned in Chapter 4, Envoice has been augmented to include phrases for Conversant Phrasebook and Learning Jupiter. Similar augmentations will be required for additional phrases. Although one way to ensure perfect synthesis is to continue to have native speakers record all the possible combinations of utterances used by the system, this approach is unscalable, hence the need for a concatenative approach such as Envoice. However, currently the synthesis from Envoice is still far from perfect, and so continued research is necessary for SLLS to have dynamic quality synthesis.

7.1.3 Incorporate Grammar Learning

Beyond displaying grammar tutorials, it is also possible to augment SLS technologies further and loosen the grammar constraints to accommodate common mistakes made by beginners. This would allow users with good pronunciation but incorrect grammar to still undertake conversations with the system, and the system would provide paraphrases with correct grammatical syntax. Furthermore, it would be possible to score the pronunciation and the grammatical structure separately, providing separate

metrics for users to improve on.

7.1.4 Improve Grading Schema

We have adopted confidence scores from the recognition as the initial grading metric for a user's speech. However, as hinted above, there are many other ways in which users can be assessed, and research is necessary to ascertain the optimal way to gauge a user's language proficiency. The PhonePass system described in Chapter 2 provides possible approaches to follow, and, given the commercial success of the system, gives us confidence that there are achievable ways to provide quality computer generated assessments. For Mandarin, tone production is a particularly difficult task for native English speakers, so a separate score for the tone production would be very beneficial.

7.2 Reduce Administrative Burden

Although we have taken the first step to developing tools to administrate and maintain SLLS, there is still much work to be done to empower users to continue to grow SLLS. We envision a system that will eventually allow non-experts in Galaxy and Internet technologies to perform all SLLS related tasks online through the web site, streamlining the development process and reducing the administrative burden. We envision a team of experts whose main responsibility would be to verify that the system has correctly processed the lesson and to manually repair any mistakes introduced in the automatic process.

7.2.1 Online Galaxy Management

Perhaps the most problematic area of the system is the instability of some of the Galaxy components. Even though a restart of the malfunctioning server is typically enough to remedy the problem, currently there is no ability for users to restart the servers remotely. To reduce the down time of the system, in the next version of SLLS, administrators should be able to check the status of the systems, restart the servers,

and be notified when servers are down all through the web site.

7.2.2 Data Acquisition and Incorporation

Recognition systems require training data to improve, and with SLLS, there is the potential to harness a wide spectrum of user speech. Currently, adding phrases and training data to the recognizer remains an involved process. However, this is being continuously improved, and hopefully this whole process can be automated, reducing the burden of the administrators and improving the recognition performance.

7.2.3 From Requests to Empowerment

The *Requests* interface is required to facilitate the exchanges of SLLS users because there are certain tasks, such as adding vocabulary to the system, that can only be completed by SLLS administrators. This burden on the administrators can eventually be reduced by tapping the resources of the SLLS user base. Although the majority of the users will be beginners hoping to learn a foreign language, there will also be teachers who are looking to use SLLS in their classes. These teachers have the motivation, the spoken language ability and the language knowledge that more than qualifies them to help in SLLS development. Functionality needs to be developed for SLLS to allow us to tap these resources, and when we are able to provide teachers with the tools that will empower them to grow SLLS, many of the concerns regarding the polishing of the language learning experience in the previous section will be addressed.

7.3 Emerging Technologies

Given the ever changing technology environment, it should come as no surprise that there are many emerging technologies that could potentially be used in SLLS. Below we discuss mobile applications, multi-modal interactions and VoiceXML as the technologies with the greatest potential impact in the future.

7.3.1 Mobile Applications

One of the key motivations to spoken language learning is to be able to speak with a foreigner in their language whenever and wherever you are. The advent of mobile computing and wireless Internet access provides the infrastructure for enabling anytime anywhere access to SLLS, and with the full dictionary of the Phrasebook system, this could be a very useful and powerful tool. The key developments necessary to make this a reality are a light weight version of SLLS for mobile clients and a recording client for the user input.

7.3.2 Multi-modal Experience

SLLS has limited multi-modal experience by displaying the conversation in real-time on the web site while it is in progress over the telephone. Studies, such as [9], have shown that using auditory and visual information together is more successful in language learning than auditory alone. Users are able to glean subtle yet important information from watching the movement of the lips, and this greatly improves their listening ability. In the future, this might mean having computer animated talking avatars on screen to engage in conversation with the user. Figure 7-1 depicts how SLLS could incorporate such a development, having one avatar for translation, and another for the conversation. Another direction that would be enriching is to enhance audio interaction with some kind of multi-modal experience such as pen-based navigation on a map.

7.3.3 VoiceXML

VoiceXML is a mark up language aimed at bringing the full power of web development and content delivery to voice response applications, and to free the authors of such applications from low-level programming and resource management. By standardizing and simplifying the development of voice applications, VoiceXML is trying to empower non-experts to develop voice systems over the telephone and over the Internet. Since VoiceXML is still in development, we do not foresee its use in SLLS

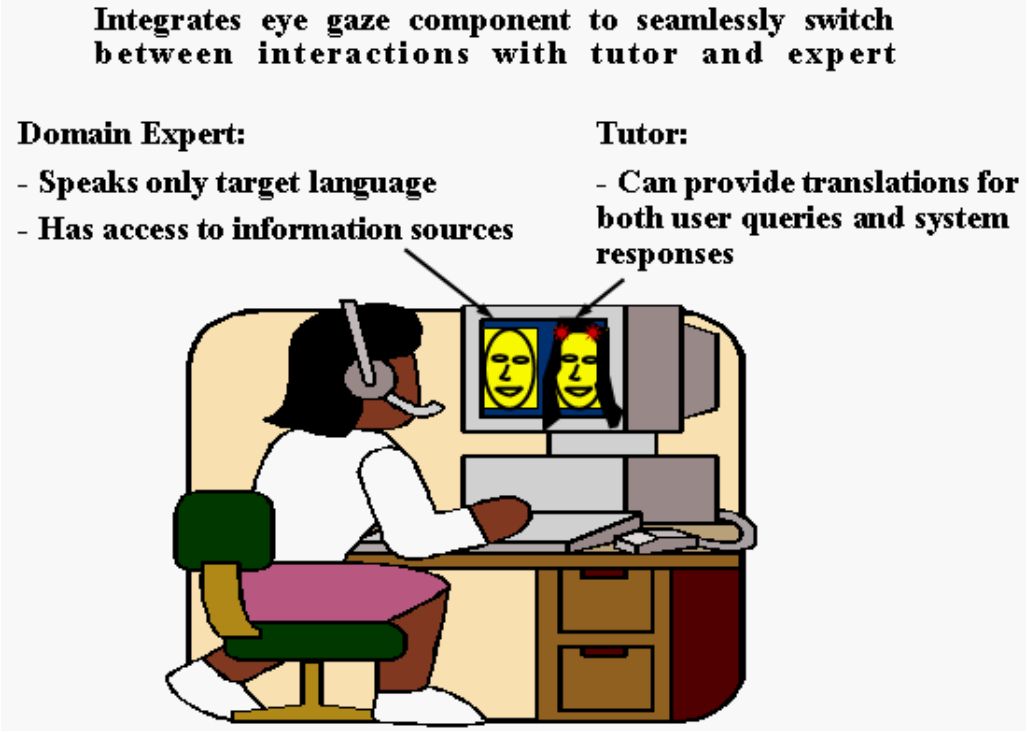


Figure 7-1: Two computer animated talking avatars with differing roles in SLLS

quite yet. However, VoiceXML may be the key to providing the empowerment tools described previously to allow non-experts to help grow SLLS.

7.4 Summary

We have introduced the first version of the Spoken Language Learning System, an online interactive platform showcasing the multilingual capabilities of SLS. Motivated by a real world demand for spoken language learning and access to research technology, we started on this ambitious project to develop a unique service. Although the system continues to be a work in progress, we were able to satisfy the primary set of goals, delivering a working prototype on an extensible platform. The limited evaluation we performed provided us with humbling yet encouraging feedback, and the vast array of future work marks a long but hopeful path to a successful spoken language learning experience.

Bibliography

[1] Speech at Carnegie Mellon University. Janus. http://www.is.cs.cmu.edu/mie/janus.html.

[2] C. Chuu. Lieshou: A Mandarin conversational task agent for the Galaxy-II architecture. Master's thesis, Massachusetts Institute of Technology, 2002.

[3] S. Crockett. Rapid configuration of discourse dialog management in conversational systems. Master's thesis, Massachusetts Institute of Technology, 2002.

[4] ELTNews. Talking on the Telephone: An interview with Professor Jared Bernstein. http://www.eltnews.com/features/interviews/index.shtml.

[5] M. Eskenazi, Y. Ke, J. Albornoz, and K. Probst. The Fluency Pronunciation Trainer: Update and user issues. In *Proceedings INSTiL2000*, Dundee, Scotland, 2000.

[6] D. Goddeau, E. Brill, J. Glass, C. Pao, M. Phillips, J. Polifroni, S. Seneff, and V. Zue. Galaxy: A human language interface to on-line travel information. In *Proceedings ICSLP*, pages 707–710, Yokohama, Japan, 1994. International Conference on Spoken Language Processing.

[7] Google. Google search for spoken computer aided language learning systems. http://www.google.com/. Search Terms: Computer-aided, language, learning.

[8] J. Kuo. An XML Messaging Protocal for Multimodal Galaxy Applications. Master's thesis, Massachusetts Institute of Technology, 2002.

[9] D. Massaro and R. Cole. From speech is special to computer aided language learning. http://mambo.ucsc.edu/psl/dwm/dwm_files/pdf/instil.pdf. Department of Psychology, University of California, Santa Cruz and CSLR, University of Colorado, Boulder, CO.

[10] Voice of America. Study Mandarin Chinese using VOA. http://www.ocrat.com/voa/.

[11] S. Seneff, C. Chuu, and D. S. Cyphers. Orion: From on-line interaction to off-line delegation. In *Proceedings of the 6th ICSLP Volume 2*, pages 142–145, Beijing, China, 2000.

[12] S. Seneff, E. Hurley, R. Lau, C. Pao, P. Schmid, and V. Zue. Galaxy-ii: A reference architecture for conversational system development. In *Proceedings ICSLP 98*, Sydney Australia, 1998.

[13] L. Mayfield Tomokiyo, L. Wang, and M. Eskenazi. An empirical study of the effectiveness of speech-recognition-based pronunciation training. In *Proceedings of the 6th ICSLP Volume 1*, pages 677–680, Beijing, China, 2000.

[14] C. Wang, S. Cyphers, X. Mou, J. Polifroni, S. Seneff, J. Yi, and V. Zue. Muxing: A telephone-access Mandarin conversational system. In *Proceedings of the 6th ICSLP Vol 2*, pages 715–718, Beijing, China, 2000.

[15] T. Xie. Conversational Mandarin Chinese Online. http://www.csulb.edu/ txie/ccol/content.htm.

[16] J. Yi, J. Glass, and L. Hetherington. A flexible, scalable finite-state transducer architecture for corpus-based concatenative speech synthesis. In *Proceedings of the 6th ICSLP Volume 3*, pages 322–325, Beijing, China, 2000.

[17] B. Zhou, Y. Gao, J. Sorensen, Z. Diao, and M. Picheny. Statistical natural language generation for trainable speech-to-speech machine translation systems. In *Proceedings of the ICSLP 2002 Volume 3*, pages 1897–1900, Denver, CO, 2002.

[18] V. Zue, S. Seneff, J. Glass, J. Polifroni, C. Pao, T. J. Hazen, and L. Hetherington. Jupiter: A telephone-based conversational interface for weather information. In *IEEE Transactions on Speech and Audio Processing Volume 8*. IEEE, January 2000.

[19] V. Zue, S. Seneff, J. Polifroni, H. Meng, and J. Glass. Multilingual human-computer interaction: From information access to language learning. In *Proceedings ICSLP*, pages 2207–2210, 1994.