The DARWARS Tactical Language Training System

W. L. Johnson, S. Marsella, H. Vilhjálmsson
CARTE, USC / Info. Science Institute
Marina del Rey, CA
Johnson@isi.edu, marsella@isi.edu, hannes@isi.edu

ABSTRACT

The DARWARS Tactical Language Training System (TLTS) helps learners acquire basic communicative skills in foreign languages and cultures. Learners practice their communication skills in a simulated village, where they must develop rapport with the local people, who in term will help them accomplish missions such as post-war reconstruction. Each learner is accompanied by a virtual aide who can provide assistance and guidance if needed, tailored to each learner's individual skills. The aide can also act as a virtual tutor as part of an intelligent tutoring system, giving the learner feedback on their performance. Learners communicate via a multimodal interface, which permits them to speak and choose gestures on behalf of their character in the simulation. The system employs video game technologies and design techniques, in order to motivate and engage learners. A version for Levantine Arabic has been developed, and versions for other languages are in the process of being developed. A first version is scheduled to be transitioned into use by US Special Forces in late 2004.

The TLTS project has developed and integrated several advanced technologies, including speech recognition tailored for learner speech, motivational tutorial dialog, learner modeling, and multi-agent social simulations. The virtual aide in the game is implemented as a pedagogical agent, able to interact with learners at a motivational and social level as well as a cognitive level. Character behavior in the game is controlled by the Psychsim cognitive modeling system, that models the motivations of social agents. Multi-user authoring tools enable linguists, instructional designers, and simulation developers to collaborate in the specification and construction of lessons and simulations in multiple languages. The TLTS is part of the DARWARS Training Superiority program developing new technologies for military training.

ABOUT THE AUTHORS

W. Lewis Johnson is director of the Center for Advanced Research in Technology for Education at the Information Sciences Institute of the University of Southern California. Dr. Johnson holds an A.B. degree in linguistics from Princeton University and a Ph.D. in computer science from Yale University. He is principal investigator of the Tactical Language project. He was program co-chair of the International Joint Conference on Artificial Intelligence in Education in 2002, and program co-chair of the International Conference on Intelligent User Interfaces is 2003. He is past president of the International Artificial Intelligence in Education Society, and past chair of the ACM Special Interest Group for Artificial Intelligence.

Stacy Marsella is a project leader at the University of Southern California's Information Sciences Institute and a research assistant professor of Computer Science at the University of Southern California. Dr. Marsella received his Ph.D. from Rutgers University in 1993. His research interests include multi-agent systems, computational models of emotion, modeling social interaction and group behavior as well as the use of simulation and interactive drama in education.

Hannes Vilhjálmsson is a research scientist in CARTE at the Information Sciences Institute of the University of Southern California. Dr. Vilhjálmsson holds a B.S. degree in computer science from the University of Iceland and a M.S. as well as a Ph.D. in media arts and sciences from the Massachusetts Institute of Technology. His research focuses on the role of nonverbal cues in face-to-face interaction and how these cues can be autonomously generated in interactive animated characters based on linguistic and social context.

The DARWARS Tactical Language Training System

W. L. Johnson, S. Marsella, H. Vilhjálmsson
CARTE, USC / Info. Science Institute
Marina del Rey, CA
Johnson@isi.edu, marsella@isi.edu, hannes@isi.edu

INTRODUCTION

The Tactical Language Training System (TLTS) gives learners basic training in foreign language and culture, focusing on communicative skills. An intelligent agent coaches the learners through lessons, using innovative speech recognition technology to assess their mastery and provide tailored assistance. Learners then practice particular missions in an interactive simulation, where they speak and choose appropriate gestures in encounters with autonomous, animated characters. Game technologies and design methods are employed to maximize learner engagement. We aim to provide effective language training both to high-aptitude language learners and to learners with low confidence in their language abilities. The simulation-based approach gives learners extensive practice in spoken communication, so that they can acquire oral proficiency rapidly. This work is being conducted iteratively; successive versions are being developed and evaluated, and the results are used to motivate further development. The TLTS is part of the DARPA Training Superiority program, which is developing just-in-time training technologies incorporating games, simulations, and intelligent tutoring capabilities.

OBJECTIVES

The Challenges of Less Commonly Taught Languages and Cultures

There is a severe shortage of personnel with training in foreign languages, particularly less commonly taught languages. In the United States, approximately ninety-one percent of Americans who study foreign languages in schools, colleges, and universities choose Spanish, French, German, or Italian, while very few choose such commonly spoken languages such as Chinese, Arabic, or Russian (NCOLCTL, 2003). Arabic accounts for less than 1% of US college foreign language enrollment (Muskus, 2003). Consequently there is a great shortage of expertise in key languages such as Arabic (Superville, 2003).

To fill this gap, the US military has developed its own training courses for strategically important languages, at the Defense Language Institute and elsewhere. Unfortunately these courses are insufficient to meet current needs. DLI courses require a substantial commitment of time in residence, and tend to be reserved for active duty personnel with a high aptitude for language. There are many other military personnel, such as special operations teams, civil affairs specialists, and military police, who are not trained as linguists but could benefit from a basic knowledge of language and culture. Moreover, the demand for skills in particular languages can change rapidly, in response to new crises overseas. Thus there is a continuing need for just-in-time training in basic communication skills, to complement the in-depth courses offered by the DLI.

The Tactical Language Training System (TLTS) provides integrated computer-based training in foreign language and culture. It employs a task-based approach, where the learner acquires the skills needed to accomplish particular communicative tasks (Doughty & Long, 2003), typically involving encounters with (simulated) native speakers. Emphasis is placed on oral proficiency. Vocabulary is limited to what is required for specific situations, and is gradually expanded through a series of increasingly challenging situations that comprise a story arc or narrative. Grammar is introduced only as needed to enable learners to generate and understand a sufficient variety of utterances to cope with novel situations. Nonverbal gestures (both "dos" and "don'ts") are introduced, as are cultural norms of etiquette and politeness, to help learners accomplish the social interaction tasks successfully. We are developing a toolkit to support the rapid creation of new task-oriented language learning environments, thus making it easier to support less commonly taught languages. The project has developed an initial version of a training system for Levantine Arabic, and an Iraqi version is under development. Training systems for Farsi and other languages will follow.

The Role of Games and Simulations

Simulations, incorporating synthetic agents playing the role of native speakers, have clear potential relevance to language learning, but are insufficient by themselves. Speech recognition technology is required so that learners can converse with simulated people – a challenge in its own right, since beginners make many errors that recognizers trained on fluent speech handle poorly. Even if a simulation were to do a perfect job of simulating native speakers, it would be insufficient for most learners. Language learners benefit from feedback on their mistakes (Lightbown & Spada, 1999), and native speakers sometimes overlook learner errors out of politeness. Language instructors are better at giving learners feedback on their errors, but many learners find classroom language instruction to be boring and unmotivating.

The TLTS addresses this problem by providing learners with two closely coupled learning environments with distinct interactional characteristics. The Mission Skill Builder incorporates a pedagogical agent that provides learners with feedback concerning their errors. The Mission Practice Environment provides authentic conversational practice in a game-like atmosphere, accompanied by an aide character who can assist if needed. The curricula of the two environments are closely related, so that the lessons that one learns in the Skill Builder apply directly to the Practice Environment, allowing learners to improve their performance in the game.

The Mission Practice Environment is built using computer game technology, and exploits game design techniques, in order to promote learner engagement and motivation. The game-based, interaction-based approach may motivate a wide range of learners, even those with low self-confidence in their language abilities, to persevere until they can successfully engage in conversation within the game. Although there is significant interest in the potential of game technology to promote learning (Gee, 2003; Swartout & van Lent, 2003), there are some important outstanding questions about how to exploit this potential. One is *transfer* – how does game play result in the acquisition of skills that transfer outside of the game? Another is how best to exploit *narrative structure* to promote learning. Although narrative structure can have the positive effect of making learning experiences more engaging and meaningful, it could also have the undesirable effect of discouraging learners from engaging in learning activities such as exploration, study, and practice that do not fit into the story line. By combining learning experiences with varying amounts of narrative structure, and by evaluating transfer to real-world communication, we hope to develop a deeper understanding of these issues.

Managing simulation behavior in this context poses particular challenges. The Mission Practice Environment incorporates multiple autonomous agents, whose behavior needs to be coordinated in order to help achieve pedagogical, social, and dramatic goals. The behavior should adjust to the level of expertise of the learner. For example when interacting with beginners the characters should be patient and allow learners as much time as they need to think of what to say. Character dialog is modeled after sample dialogs that instructional designers create to address particular learning objectives, but should not be limited to those dialogs; if a learner says something that departs from the reference dialogs but which is still appropriate in context, the non-player characters should respond appropriately. Unlike earlier speech-enabled foreign language simulations such as Conversim (Harless et al., 1999), learners do not select what to say from set of pre-enumerated utterances. At the same time character behavior should be consistent with the profiles of the characters and narrative structure. Behavior control should be fully autonomous; unlike simulation tools such as ModSAF the TLTS does not rely on a human exercise controller who supervises and controls the behavior of the non-player characters.

The TLTS builds on ideas developed in previous systems involving microworlds (e.g., FLUENT, MILT) (Hamberger, 1995; Holland et al., 1999), interactive pedagogical drama (Marsella et al., 2000; 2003) conversation games (e.g., Herr Kommissar) (DeSmedt, 1995), speech pronunciation analysis (Witt & Young, 1998), learner modeling, simulated encounters with virtual characters (e.g., Subarashii, Virtual Conversations, MRE) (Bernstein et al., 1999; Harless et al., 1999; Swartout et al., 2001). It extends this work by combining rich game interaction and language form feedback, in an implementation that is robust and efficient enough for ongoing testing and use on commodity computers

EXAMPLE

The following scenario illustrates how the TLTS is used. The scenario is a civil affairs mission in Lebanon. The learner's unit is has been assigned the task of entering a village, establishing rapport with the local people, making contact with the local headman in charge, and arrange to carry out post-war reconstruction.

To prepare for the mission, the learner goes into the Mission Skill Builder to practice the necessary communication skills, as shown in Figure 1. Here, for example, the virtual tutor is introducing a common greeting in Lebanese Arabic, /marHaba/. As the learner practices saying /marHaba/ his speech is automatically analyzed for pronunciation errors and the virtual tutor, Nadiim, provides immediate, supportive feedback. If the learner mispronounces the guttural /H/ sound in /marHaba/, Nadiim points out the error and encourages the learner to keep practicing until he is able to pronounce it correctly. Meanwhile a learner model keeps track of the communication skills that the learner has mastered.

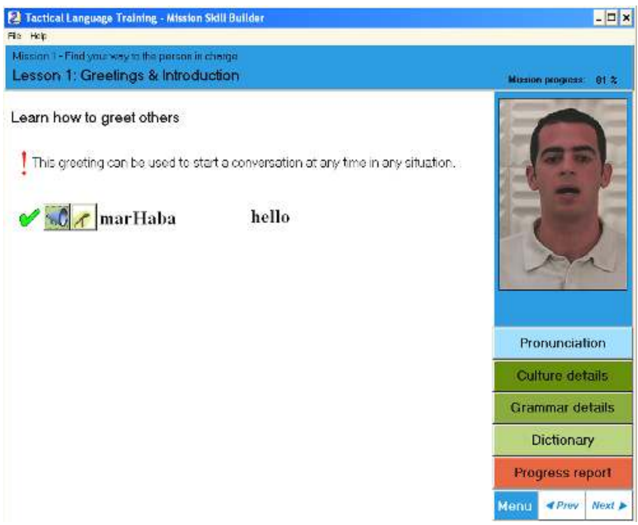

Figure 1. A coaching section in the MSB

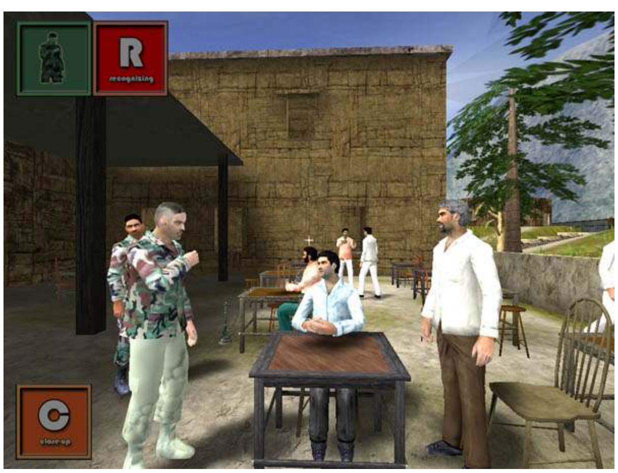

Figure 2. Greeting a Lebanese man in a café

When the learner is ready, he enters the Mission Practice Environment. His character in the game, together with Nadiim now acting as his aide, enters the village. They enter a café, and start a conversation with a man in the café, as shown in Figure 2. The learner speaks in into a microphone, while choosing appropriate nonverbal gestures from an on-screen menu. In this case the learner chooses a respectful gesture, and his interlocutor, Ahmed, responds in kind. If the learner is uncertain about what to say next the aide can offer a suggestion, as shown in Figure 3. The suggestions depend upon the learner's mastery, as indicated by the learner model. If the learner is familiar with the necessary vocabulary then the aide will give a hint in English, as shown in the figure. If the learner does not know the necessary vocabulary, Nadiim suggests a specific Arabic phrase to say. The learner thus learns new vocabulary in the context of use.

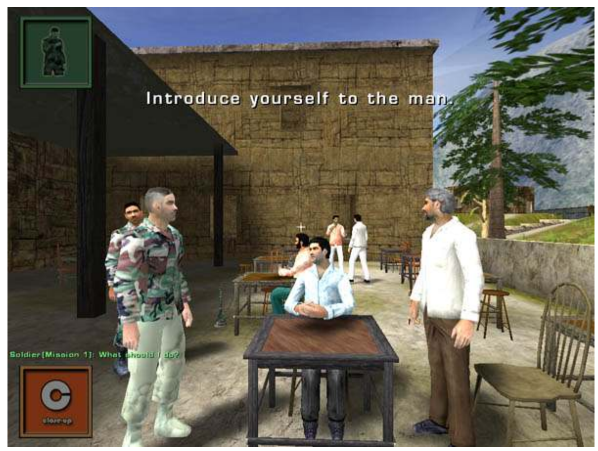

Figure 3. Receiving guidance from the aide

OVERALL SYSTEM ARCHITECTURE

The TLTS architecture is designed to support several important internal requirements. A Learner Model has to be accessible by both the Skill Builder and the Practice Environment and allow for run-time queries and updates. Learners need to be able to switch back and forth easily between the Skill Builder and the Practice Environment, as they prefer. The system must support relatively rapid authoring of new content by teams of content experts and game developers. The system must also be flexible enough to support modular testing and integration with the DARWARS architecture, which is intended to provide any-time, individualized cognitive training to military personnel. Given these requirements, a distributed architecture makes sense (see Figure 4). Modules interact using content-based messaging, currently implemented using the Elvin messaging service.

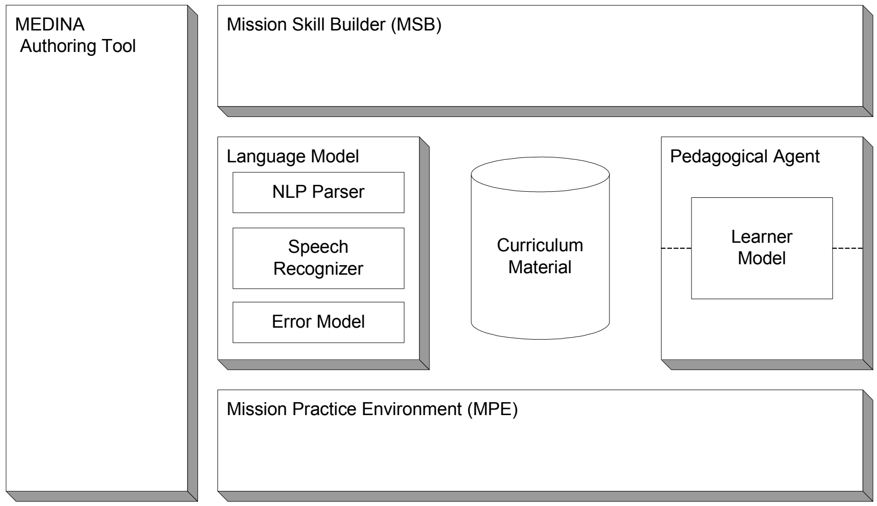

Figure 4: The overall TLTS architecture

Pedagogical Agent

The Pedagogical Agent monitors learner performance, and uses performance data both to track the learner's progress in mastering skills and to decide what type of feedback to give to the learner. The learner's skill profile is recorded in a Learner Model, which is available as a common resource, and implemented as a set of inference rules and dynamically updated tables in an SQL database. The learner model keeps a record of the number of successful and unsuccessful attempts for each action over the series of sessions, as well as the type of error that occurred when the learner is unsuccessful. This information is used to estimate the learner's mastery of each vocabulary item and communicative skill, and to determine what kind of feedback is most appropriate to give to the learner in a given instance. When a learner logs into either the Skill Builder or the Practice Environment, his/her session is immediately associated with a particular profile in the learner model. Learners can review summary reports of their progress, and in the completed system instructors at remote locations will be able to do so as well.

Language Model

To maintain consistency in the language material, such as models of pronunciation, vocabulary and phrase construction, a single Language Model serves as an interface to the language curriculum. The Language Model includes a speech recognizer that both applications can use, a natural language parser that can annotate phrases with structural information and refer to relevant grammatical explanations and an Error Model which detects and analyzes syntactic and phonological mistakes (Mote et al., 2004). Currently the parser is applied to curriculum materials as they are authored, and the speech recognizer and error model are applied to learner speech as the learner interacts with the TLTS.

Curriculum Materials

While the Language Model can be thought of as a view of and a tool to work with the language data, the data themselves are stored in a separate Curriculum Materials database. This central database contains all missions, lessons and exercises that have been constructed, in a flexible Extensible Markup Language (XML) format, with links to media such as sound clips and video clips. It includes exercises that are organized in a recommended sequence, and tutorial tactics that are employed opportunistically by the pedagogical agent in response to learner actions. The database is the focus of the authoring activity. By having all data reside in a single place, the system can more easily keep track of modifications and their effects. Entries can be validated using the tools of the Language Model. Authoring tools operate on this data, allowing people with different authoring roles can bring up different views of the curriculum material and edit it individually while overall consistency is ensured.

Speech Processing

Since speech is the primary input modality of the TLTS, robustness and reliability of speech processing are of paramount concern. The variability of learner language makes robustness difficult to achieve. Most commercial automated speech recognition (ASR) systems are not designed for learner language (LaRocca et al., 1999), and commercial computer aided language learning (CALL) systems that employ speech tend to overestimate the reliability of the speech recognition technology (Wachowicz & Scott,1999). To support learner speech recognition in the TLTS, our initial efforts focused on acoustic modeling for robust speech recognition especially in light of limited domain data availability (Srinivasamurthy & Narayanan, 2003). In this case, we bootstrapped data from English and modern standard Arabic and adapted it to Levantine Arabic speech and lexicon. Dynamic switching of recognition grammars was also implemented. The recognition algorithm was then extended to generate confidence scores, which the Pedagogical Agent uses to choose what type of feedback to give to the learner. The structure of the recognition networks is distinct for the MSB and the MPE environments. In the MSB mode, the recognition was based on limited vocabulary networks with pronunciation variants and hypothesis rejection. On the other hand, in the MPE mode, the recognizer supports less constrained user inputs. In our ongoing work, we plan to generate confidence scores at suprasegmental levels, to provide feedback on pronunciation quality at different time scales.

Mission Skill Builder Architecture

The Mission Skill Builder (MSB) is a one-on-one tutoring environment which helps the learner to acquire mission-oriented vocabulary, communication skills, pronunciation skills, and knowledge of relevant gesture and culture. It consists of a set of lessons, exercises, and quizzes, as well as supplementary materials dealing with pronunciation, grammar, and culture. The MSB interacts with the pedagogical agent and speech recognizer in order to evaluate the learner's speech input, evaluate performance, and generate feedback. The MSB also includes a progress report generator that summarizes the learner's current level of mastery.

The Mission Skill Builder user interface is currently implemented in SumTotal's ToolBook, augmented by the speech recognizer. The learner initiates speech input by clicking on a microphone icon, which sends a "start" message to the automated speech recognition (ASR) process. Clicking the microphone icon again sends a "stop" message to the speech recognition process, which then analyzes the speech and sends the recognized utterance back to the MSB. The recognized utterance, together with the expected utterance, is passed to the Pedagogical Agent, which in turn passes this information to the Error Model (part of the Language Model), to analyze and detect types of mistakes. The results of the error detection are then passed back to the Pedagogical Agent, which decides what kind of feedback to choose, depending on the error type and the learner's progress. The feedback is then passed to the MSB and is provided to the learner via the virtual tutor persona, realized as a set of video clips and sound clips.

The Skill Builder also provides learners with a variety of supplementary materials. A Pronunciation page presents close-up video clips showing how the various sounds of the Levantine Arabic language are produced. The combine close-up videos of a speaker's mouth with animations of the vocal tract generated using the Baldi toolkit (Massaro, 2004). A Language Details page presents translations, grammatical parses, and relevant grammar rules, synthesized automatically using the TLTS's authoring tools. A similar collection of relevant cultural information is currently under development, based upon material developed by the Defense Language Institute (DLI, 2004).

Mission Practice Environment Architecture

The Mission Practice Environment (MPE) is a 3D game environment where the learner can put the knowledge acquired in the MSB to the test in a simulated world populated with interactive characters. The work builds on previous work in interactive pedagogical drama (Marsella et al., 2000) and generation of social behavior in virtual actors (Vilhjalmsson, 2004). The learner can engage in dialogue with these characters by speaking into a microphone and choosing gestures with the mouse. To make the experience engaging and provide a strong context for the interaction, the learner progresses through scenes that form a compelling story. Each scene has a set of objectives that practice certain communicative skills.

In the scene depicted in Figure 2, the learner is having a conversation with two characters (multi-party conversation), a young man (sitting) and an older man (standing). The soldier standing right behind the learner is the aide that follows the learner around and provides assistance when needed. There are also other characters in the scene, sitting or standing around the café, that provide a dynamic backdrop much like one would experience in a film.

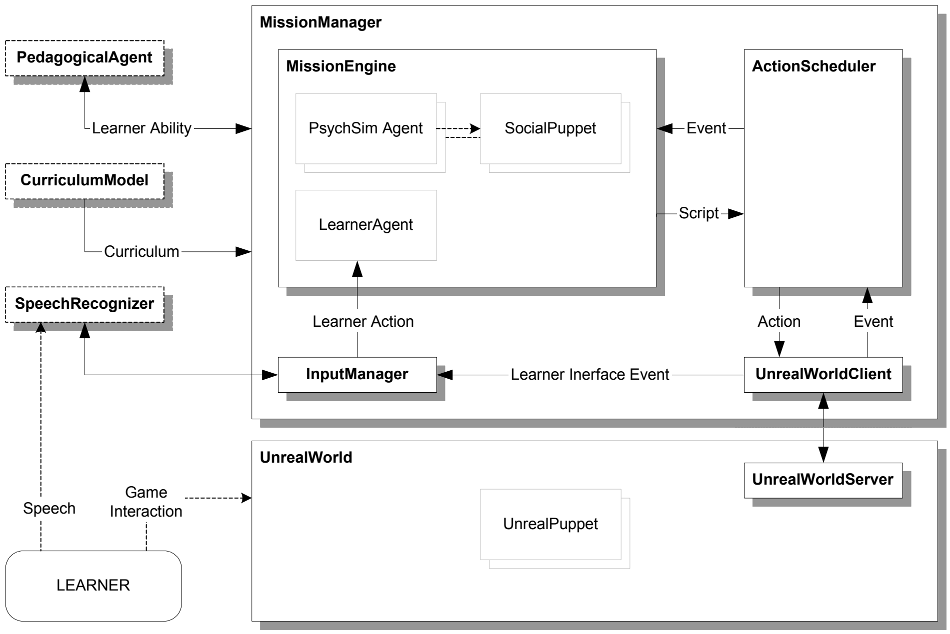

Figure 5: The Mission Practice Environment architecture

However, unlike in a film, each one of those characters can be engaged in conversation to give the illusion of a living town.

The Mission Practice Environment has two parts: The Mission Manager and the Unreal World (see Figure 5). The former controls what happens while the latter renders it on the screen and provides a user interface.

The Unreal World uses the Unreal Tournament 2003 game engine where each character, including the learner's own avatar, is represented by an animated figure called an Unreal Puppet. The motion of the learner's puppet is for the most part driven by input from the mouse and keyboard, while the other puppets receive action requests from the Mission Manager through the Unreal World Server, which is an extended version of the Game Bots server (Kaminka et al., 2002). In addition to relaying action requests to puppets, the Unreal World Server sends information about the state of the world back to the Mission Manager.

The Mission Manager is a Python application that hosts several processes that together coordinate the simulated scenes. Learner events that come in from the user interface, such as mouse button presses, are first processed in an Input Manager, and then handed to a Mission Engine where a proper reaction by other characters is generated. The Input Manager is also responsible for communicating with the speech recognizer.

The Mission Engine controls all characters by assigning an agent (an autonomous control program) to each one. The engine receives events from the environment, such as when the learner's character walks into the café, speaks or gestures. The learner speaks by toggling into a recording mode with a mouse click (see the red icon in Figure 2), which results in the Input Manager creating a speech event with the output from the speech recognizer. The learner can also select a gesture with the mouse wheel (see the green icon in Figure 2) which the Input Manager integrates into the same speech event. The system also has access to the Pedagogical Agent that was described in the previous section and the Curriculum. Based on these inputs, the Mission Engine decides how to respond, based on what each agent wants to do.

Each agent must automatically generate behavior in response to the learner's actions that appears realistic, regardless of what the learner chooses to do. To accomplish this, The Mission Engine simulation runs through the following three stages. During the first stage, each agent updates its beliefs about the state of the world and the other agents based on available inputs. For example, if the learner has just announced that his mission is peaceful, both the old man and the young man in the café scene will believe that they can trust him more than before. The second stage involves deciding how the agents should respond to the event. Currently, we have a simple mechanism for deciding who takes the dialog turn. After the learner speaks, each character (in a predefined order) can decide whether it wants to respond (based on the recent dialog). This may mean for example, that the learner says something, the old man may respond and then the young man has the option to say something as well (in response to either the student or the old man).

However, this strict order can be altered by emotional arousal level. Each character has an arousal level, which indicates how angry or worried this character is. The arousal level is updated each time a new action is perceived. Occasionally, a character may have a very high arousal level, guaranteeing that the character gets the next turn. For example in the café scene, the young man starts out trusting the learner less than the old man does. If the learner fails to build sufficient trust (for example, failing to describe his mission or using the inappropriate gesture), the young man will interrupt the dialog between the learner and the old man and accuse the learner of being a spy. In the third and last stage the character that has the turn decides what action to take. This character's action will then be added to the queue of actions that can be perceived by all characters.

The underlying agent technology used for the characters is the PsychSim multi-agent system [12]. PsychSim was chosen for several reasons. It models social relationships and reasoning, a key requirement for realizing the social and cultural interactions required in TLT. For example, PsychSim models factors such as trust and support between agents. Agents also have mental models of other agents and can employ those models to inform their decision-making about whether to believe another agent, what action to take, etc. In addition, the agents are realized as Partially Observable Markov Decision Processes (POMDP). Partial observability provides a mechanism to populate an entire "world" where agents may not have access to complete set of observations.

Once an agent has decided on the action to take, a layer termed the Social Puppet coordinates the realization of the embodied action by the puppet that represents the agent in the virtual environment. This layer is responsible for planning the actual verbal and nonverbal behavior that appropriately and expressively realizes the agent's communicative intent given the social context, based on a model of particular culture and language. This layer is also responsible for generating appropriate reactive behavior in all the puppets involved in the scene according to social rules, such as glances and posture shifts, to reflect the tight coordination of behavior by all members of any social gathering. The Social Puppet layer finally hands scripts of puppet behavior to an Action Scheduler that coordinates the execution of these behaviors in the Unreal Puppets within the Unreal World.

The difficulty of a scene can be adjusted according to learner's language ability as reported by the Pedagogical Agent, so that learner will have an experience that is hard enough to provide good practice, but not so hard it leads to frustration. The difficulty of scenes can be adjusted by altering the personalities and goals of the virtual characters. For example, for a learner who is starting out with low language ability, it will take less to convince the virtual characters that the purpose of learner's mission is to help their village, and thus the above confrontation would be avoided.

Also in order to prepare learners for entering the social simulation, the MPE is being extended with smaller interactive games that can be played between scenes. These arcade style games give learners practice with particular communicative skills. For example, in one such game the learner has to follow verbal directions to exit a maze.

AUTHORING PROCESS

Authoring is an iterative process, supported by tools that operate on partial XML descriptions of the lesson content. Authoring begins with descriptions of *scenes*, which identify characters, stage settings, and possible dialog exchanges between characters. Dialog exchanges are developed in parallel in English and the target foreign language, through collaboration between curriculum developers and native speakers of the target language. These scene descriptions are progressively extended with stage directions, gesture timings, and indications of communicative intent, all used to specify the behavior of the agents in the MPE. A structured text editor, that presents the XML scene descriptions in easy-to-read form, is currently used to create and edit the scene descriptions. The scene descriptions may be further refined as the agent models are constructed, and it is determined that the agent models are capable of generating a wider range of communicative behaviors than is captured in the scene descriptions.

As scene dialogs are constructed, an interconnected set of language resources for words and phrases appearing in the dialogs is assembled: a foreign language lexicon, a foreign language-English dictionary and an ontology to support the foreign language. The Contex (Hermjakob, 2000) natural language processing engine facilitates this process. Contex assembles lexical resources from the dialog descriptions and identifies words that have not previously occurred and also can not be derived using a morphological analyzer, often because of inconsistent spellings in the scene descriptions. Contex then parses the dialog lines and uses the parse tree to annotate the lines with context-sensitive English glosses, the syntactic and semantic structure of a sentence, as well as relevant grammar explanations that are automatically selected from a library of grammar notes and tailored to the foreign language sentence. These annotations are then summarized in interactive HTML pages that are accessible from the Mission Skill Builder.
.
The scene descriptions are also used as a starting point for defining appropriate Skill Builder lessons. Example dialogs serve as a starting point for specifying relevant vocabulary pages and exercises. Once lessons are specified in XML, the XML descriptions are used to automatically generate ToolBook lesson pages.

The scene description authoring tools have been augmented to support multi-lingual authoring. The same dialog lines can be displayed side by side in different languages, and the authoring tool lets the author choose which language she wishes to edit at a given time. We have used this technique to author variants of the civil affairs scenario described above in Iraqi Arabic dialect, a first step toward creating a new Tactical Language training for Iraqi Arabic.

An authoring tool named Medina is being developed that provides a unified view of the authoring process. Medina provides a drag-and-drop interface that allows authors to construct scene scripts, construct lessons from templates, assemble lessons into sets focused on particular missions, and define nonlinear sequencing between lessons.

EVALUATION

System and content evaluation is being conducted through a staged, systematic process that involves both critiques from second language learning experts and feedback from learners. Learners at the US Military Academy and at USC have worked through the first set of lessons and scenes and provided feedback. Meanwhile learner speech and system response is automatically recorded and logged, to provide data to support further improvement of the speech recognition and feedback models.

In May 2004 a formative evaluation was performed with seven college-age subjects at USC in May 2004. The subjects found the MPE game to be fun and interesting, and were generally confident that with practice would be able to master the game. This supports our hypothesis that the TLTS will enable a wide range of learners, including those with low levels of confidence, to acquire communication skills in difficult languages such as Arabic.

However, the learners were generally reluctant to start playing the game, because they were afraid that they would not be able to communicate successfully with the non-player characters. Learners usually started playing the game only when experimenters insisted that they do so. To address this problem, we are modifying the content in the MSB to give learners more conversational practice and encourage learners to enter the MPE right away.

The evaluation also revealed problems in the MSB Tutoring Agent's interaction. The agent applied a high standard for pronunciation accuracy, which beginners found difficult to meet. At the same time, inaccuracies in the speech analysis algorithms caused the agent in some cases to reject utterances that were pronounced correctly. The algorithm for scoring learner pronunciation has since been modified, to give higher scores to utterances that are pronounced correctly but slowly; this eliminated most of the problems of correct speech being rejected. We have also adjusted the feedback selection algorithm to avoid criticizing the learner when speech recognition confidence is low.

The evaluations also revealed some problems with the current distributed architecture. Messages between modules were sometimes lost or fell out of sequence, making it necessary to restart the program. Although the distributed approach facilitates rapid software prototyping, it will be necessary to move to a more integrated approach as the TLTS is prepared for deployment.

The next series of evaluations will take place at Ft. Bragg in the summer of 2004. Learners will work with the TLTS under four different experimental conditions: 1) using the complete TLTS system, 2) using the MSB alone, 3) using the MPE alone, and 4) using the MSB without pronunciation feedback. The evaluation will assess learning outcomes and learner attitudes toward language learning.

CONCLUSIONS AND CURRENT WORK

The Tactical Language Project has been making rapid progress toward the development of computer-based language training incorporating speech recognition, autonomous multi-agent simulations, and intelligent tutoring. In recognition of this progress, the US Special Operations Command has entered into an agreement with DARPA to support technology transition into Special Operations foreign language training. A complete version of the Lebanese Arabic training system is scheduled to be delivered to Ft. Bragg in late 2004.

New scenes and new lessons are currently under development. Further evaluations are planned to test the effectiveness of the system in promoting learning. In addition, the project plans to undertake the following tasks:

- Develop additional authoring tool support, and integrate the Medina authoring interface into the authoring suite,
- Detect and automatically give feedback on wider range of syntactic and semantic errors,
- Incorporate situation-specific cultural training in the Skill Builder,
- Incorporate automated tracking of learner focus of attention, to detect learner difficulties and provide proactive help, and
- Design and conduct summative evaluations to assess the effectiveness of the TLTS in promoting learning, and analyze the contributions of the various features of the TLTS to learning outcomes.

ACKNOWLEDGEMENTS

The project team includes, in addition to the authors, CARTE members Catherine M. LaBore, Carole Beal, David V. Pynadath, Nicolaus Mote, Shumin Wu, Ulf Hermjakob, Mei Si, Nadim Daher, Gladys Saroyan, Youssef Nouhi, Hartmut Neven, Chirag Merchant and Brett Rutland; from the US Military Academy COL Stephen Larocca, John Morgan and Sherrie Bellinger; from the USC School of Engineering Shrikanth Narayanan, Naveen Srinivasamurthy, Abhinav Sethy, Jorge Silva, Joe Tepperman and Larry Kite; from Micro Analysis and Design Anna Fowles-Winkler, Andrew Hamilton and Beth Plott; from the USC School of Education Harold O'Neil, and Sunhee Choi, and Eva Baker from UCLA CRESST. This project is sponsored by the US Defense Advanced Research Projects Agency (DARPA).

REFERENCES

Bernstein, J., Najmi, A. & Ehsani, F. (1999). Subarashii: Encounters in Japanese Spoken Language Education. CALICO Journal, 16 (3), 361-384.

DeSmedt, W.H. (1995). Herr Kommissar: An ICALL conversation simulator for intermediate German. In V.M. Holland, J.D. Kaplan, & M.R. Sams (Eds.), Intelligent language tutors: Theory shaping technology, 153-174. Mahwah, NJ: Lawrence Erlbaum.

DLI (2004). Iraqi Familiarization. Multimedia course, Defense Language Institute.

Doughty, C.J. & Long, M.H. (2003). Optimal psycholinguistic environments for distance foreign language learning. Language Learning & Technology 7(3), 50-80.

Gee, P. (2003). What video games have to teach us about learning and literacy. New York: Palgrave Macmillan.

Hamberger, H. (1995). Tutorial tools for language learning by two-medium dialogue. In V.M. Holland, J.D. Kaplan, & M.R. Sams (Eds.), Intelligent language tutors: Theory shaping technology, 183-199. Mahwah, NJ: Lawrence Erlbaum.

Harless, W.G., Zier, M.A., and Duncan, R.C. (1999). Virtual Dialogues with Native Speakers: The Evaluation of an Interactive Multimedia Method. CALICO Journal 16 (3), 313-337.

Hermjakob, U. (2000). Rapid Parser Development: A Machine Learning Based Approach for Korean. Proceedings of the North-American Chapter of the Association for Computational Linguistics (NAACL-2000), pp. 118-123

Holland, V.M., Kaplan, J.D., & Sabol, M.A. (1999). Preliminary Tests of Language Learning in a Speech-Interactive Graphics Microworld. CALICO Journal 16 (3), 339-359.

Kaminka, G.A., Veloso, M.M., Schaffer, S., Sollitto, C., Adobbati, R., Marshall, A.N., Scholer, A. and Tejada, S. (2002). GameBots: A Flexible Test Bed for Multiagent Team Research. Communications of the ACM, 45 (1). 43-45.

LaRocca, S.A., Morgan, J.J., & Bellinger, S. (1999). On the path to 2X learning: Exploring the possibilities of advanced speech recognition. CALICO Journal 16 (3), 295-310.

Lightbown, P.J. & Spada, N. (1999). How languages are learned. Oxford: Oxford University Press.

Marsella, S., Johnson, W.L. and LaBore, C.M (2003). An interactive pedagogical drama for health interventions. In Hoppe, U. and Verdejo, F. eds. Artificial Intelligence in Education: Shaping the Future of Learning through Intelligent Technologies, Amsterdam: IOS Press.

Marsella, S., Johnson, W.L. and LaBore, C.M. (2000), Interactice pedagogical drama. In Proceedings of the Fourth International Conference on Autonomous Agents, pages 201-308, New York, 2000. ACM Press.

Massaro, D. (2004). Symbiotic value of an embodied agent in language learning. Proceedings of the HICCS Conference.

Mote, N., Johnson, W.L., Sethy, A., Silva, J., & Narayanan, S. (2004). Tactical Language detection and modeling of learner speech errors: The case of Arabic tactical language training for American English speakers. InSTIL Symposium on NLP and Speech Processing in Language Learning, in press.

Muskus, J. (2003). Language study inceases. Yale Daily News, Nov. 21, 2003.

NCOLCTL (2003). National Council of Less Commonly Taught Languages. http://www.councilnet.org

Srinivasamurthy, N. and Narayanan, S. (2003), ``Language-adaptive Persian speech recognition", Proc. Eurospeech (Geneva,Switzerland).

Swartout, W.,Gratch, J., Johnson, W.L., et al. (2001). Towards the Holodeck: Integrating graphics, sound, character and story. Proceedings of the Intl. Conf. on Autonomous Agents, 409-416. New York: ACM Press.

Superville, D. (2003). Lack of Arabic speakers hurts U.S. Associated Press, Nov. 18, 2003.

Swartout, W. & van Lent, M. (2003). Making a game of system design. CACM 46(7), 32-39.

Vilhjalmsson, H. (2004). Animating Conversation in Online Games. Proceedings of the International Conf. on Entertainment Computing. Eindhoven. In press.

Wachowicz, A. and Scott, B. (1999). Software That Listens: It's Not a Question of Whether, It's a Question of How. CALICO Journal 16 (3), 253-276.

Witt, S. & Young, S. (1998). Computer-aided pronunciation teaching based on automatic speech recognition. In S. Jager, J.A. Nerbonne, & A.J. van Essen (Eds.), Language teaching and language technology, 25-35. Lisse: Swets & Zeitlinger.