The NLP Role in Animated Conversation for CALL

Michael Schoelles and Henry Hamburger
Dept. of Computer Science
George Mason University
Fairfax, VA 22030 USA
mikejs@cs.gmu.edu & henryh@cs.gmu.edu

Abstract

Language learning is a relatively new application for natural language processing (NLP) and for intelligent tutoring and learning environments (ITLEs). NLP has a crucial role to play in foreign language ITLEs, whether they are designed for explicit or implicit learning of the vocabulary and grammar. FLUENT is an implicit approach, in which NLP and shared-control animation support a two-medium conversation, designed to foster implicit learning of language. This report highlights specific ways that FLUENT already uses NLP, suggests potential benefits from additional use of NLP and grounds the method in widely used language learning pedagogy. It concludes by describing and evaluating the system's use in the classroom, with a particularly challenging kind of learner.

1 Introduction

Language learning is a relatively new application for natural language processing (NLP), compared to translation and database interfaces. The 1988 Trieste conference on foreign language intelligent tutoring systems did not manage a proceedings, so few were the functioning systems using NLP and other techniques of artificial intelligence (AI). Indeed the field of Computer-Assisted Language Learning (CALL) has been dominated by the work of language teachers who, tired of waiting for us, have proceeded to learn some programming or hire computer science students to make their own systems, most of them just for fairly inflexible drill-and-practice. The 1990s have witnessed increasing interest in bringing AI to language learning systems, reflected in Chanier's (1994) special issue, Holland's anthology (1995) and Zock's (1996) panel. The AI is principally of two kinds: NLP and what we will call ITLE, the overlapping categories of intelligent tutoring systems and interactive learning environments. In this paper, we report on a foreign language ITLE that has moved from the lab into the classroom. After noting alternative strategies for using NLP in CALL (section 2), we describe our system and its NLP requirements (section 3), as well as its pedagogical foundations (section 4). Finally, we describe its graduation into the classroom, for use with a particularly challenging kind of learner: the highly motivated but computer-unfamiliar immigrant.

2 A Crucial Role for NLP in CALL

Developers of foreign language ITLEs have only recently begun to expand the use of NLP in language learning systems beyond relatively simple uses of syntax. More sophisticated and widespread use has awaited greater robustness and ease of use of NLP software as well as better speed and memory of the hardware.

Recent improvements along these lines are playing key roles in meeting the special NLP requirements of foreign language ITLEs. Since the language is the subject matter to be learned, ITLEs must expect the input to be in error, and therefore require very robust NLP systems with error detection capabilities. In addition, conversational ITLEs must respond to student input in a time frame that parallels natural dialog. Advances in parsing technology and faster machines with more memory are making NLP attractive to the developers of ITLEs. The flexibility, increased coverage and other advantages of NLP are finding their way into the design of language learning systems. The issue is no longer whether language learning systems should use NLP, but how NLP can best be applied to language learning.

One possible strategy for creating a language ITLE can be called the direct approach. A tenet for ITLEs in general is that they represent the knowledge to be learned in a domain expertise module. In the case of language learning, the knowledge to be learned already exists in computational form as the lexicon and grammar of a NLP system, so the idea would be to incorporate these NLP knowledge bases as the ITLE domain expertise. However, successful language use, in contrast to other ITLE domains, is not just a matter of overt stepwise reasoning. The goal of language learning is not (only) to conjugate verbs or passivize sentences, but to understand and produce meaningful language in context. Even for the limited role of teaching grammar explicitly, a direct approach encounters two obstacles. First, a performance-oriented representation of domain knowledge is typically not structured in a way that is cognitively suitable for learners (Clancey, 1987). Moreover, NLP knowledge bases cannot be used directly in explanation since their representation language is incomprehensible to students (Chanier et al., 1992).

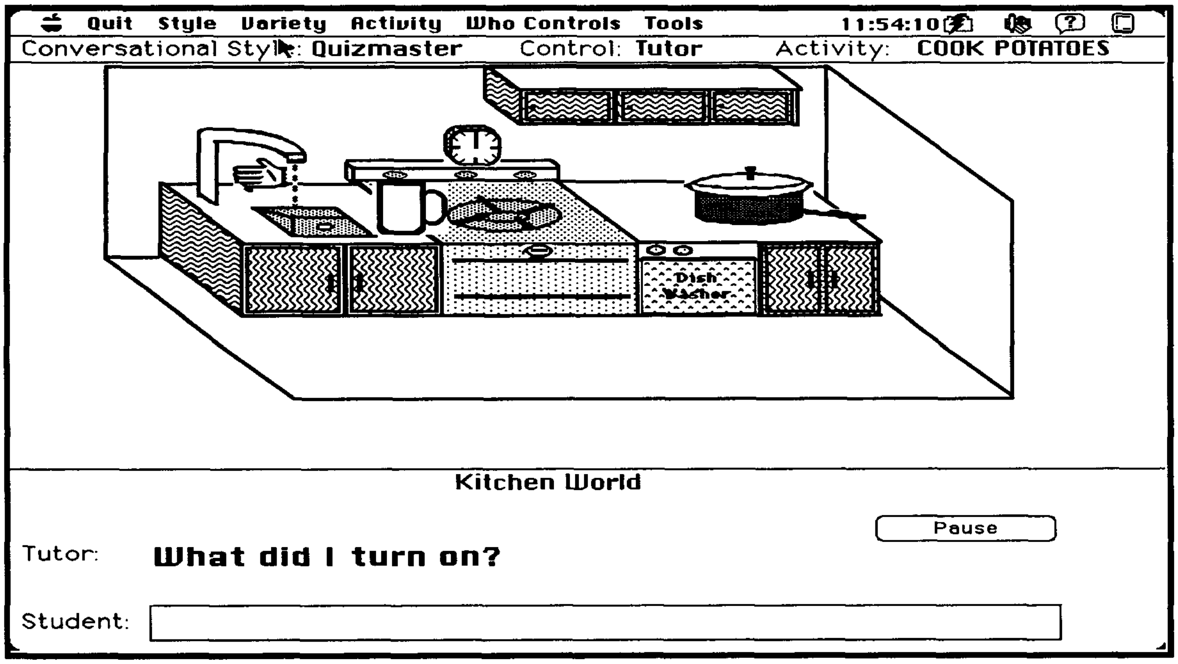
To avoid these pedagogical problems, we turn to an alternative strategy that deploys NLP principally in the ITLE's learning environment, rather than as explicit expertise. This approach is more concerned with the performance of the NLP rather than its competence. ITLEs applying NLP in this manner immerse the student in the language by situating the student in a life-like conversational setting.

Pedagogical motivation for this kind of foreign language learning environment comes from classroom methods like Total Physical Response or TPR (Asher, 1977) and the Natural Approach (Krashen and Terrell, 1983). These techniques require only comprehension at first, as students act out appropriate responses to the teacher's commands. As student actions indicate sufficient progress, constituents of the utterances are recombined to increase the skill at understanding. There are also questions, which TPR students answer by pointing. In the natural approach, production is included, beginning with questions that take very short answers. There is a gradual progression from yes-no questions to other single-word-answer questions and either-or questions, and then on to phrases and ultimately sentences.

The flexibility, efficiency and error handling of NLP is crucial to implementing the above pedagogy in an ITLE. A system that puts NLP to crucial use in support of these pedagogical considerations is FLUENT, to which we now turn.

3 FLUENT and NLP

FLUENT uses NLP to converse with the student in the context of a realistic microworld situation. Actions performed by both the student and the system tutor appear as context-dependent animation and cause updates in the underlying representation. The generativity of both the NLP and the animation is crucial to the flexibility of this conversation, allowing it to be steered in pedagogically useful directions, yet also permitting student exploration. The NLP module generates descriptions, commands and questions that may be related to actions that the student carries out graphically with the mouse or may be related to the states of objects in the current situation. The generated language appears as both text and speech. Figure 1 shows FLUENT's Kitchen Microworld.



Conversational Style: Quizmaster Control: Tutor Activity: COOK POTATOES

Kitchen World

Tutor: What did I turn on?

Student:

Figure 1. A FLUENT Microworld

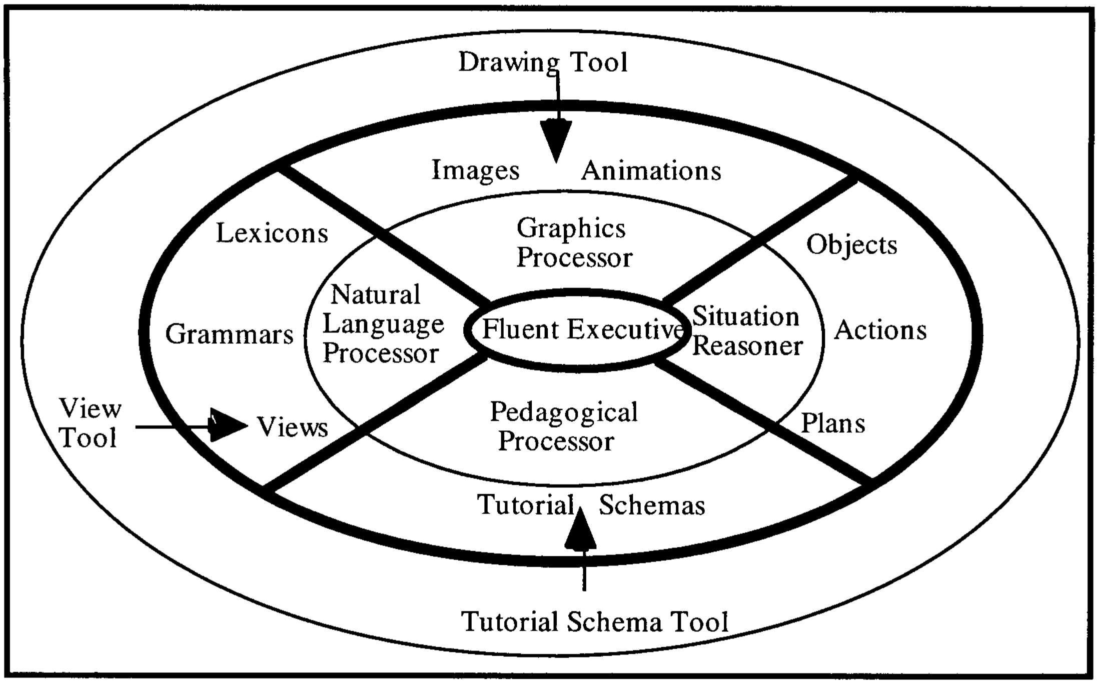


Figure 2. FLUENT Architecture

This section describes first the overall architecture of FLUENT and then turns to the specific NLP components, noting their roles and requirements in the context of FLUENT's pedagogical goals. Flexibility, extendibility and teacher involvement are achieved in FLUENT through its layered architecture, shown in Figure 2. In this framework, independent processes work on declarative structures that have been acquired by the system through tools that do not demand a knowledge of programming or linguistic theory (Schoelles and Hamburger, 1996). This work shares the interface orientation of Caldwell and Korelsky (1994), but is more abstract and flexible in that user-specifications are independent of, and combinable with, domain plans. On the other hand we do not follow work like Traum and Allen (1994), in pushing toward computation about more and more discourse phenomena

The outer ring of Figure 2 depicts the knowledge acquisition level. It identifies existing tools for building tutorial schemas, language usage structures and the graphics of objects. The tutorial schema tool lets a teacher express pedagogical expertise in the form of lessons. The view tool allows the teacher some degree of control over the language generated by the system. The teacher inputs the language specifications for a lesson by manipulating a graphical user interface. The tools process these specifications to produce instantiated objects for the declarative level.

The knowledge base, in the second ring, consists of:

- Tutorial Schemas - structures, created by a teacher using tools, that coordinate the activities, conversational style and conversational variety of FLUENT when the system is in tutoring mode.
- Views - structured objects that provide language specifications and constraints to achieve conversational variety, control over the difficulty of input presented to the student, and situational flexibility.
- Lexicon - currently 550 words entered in base form, with information for finding inflected forms and with irregular forms are subentries. Contains both subcategorization and theta grids.
- Grammar - context-free productions plus reduction functions that restrict the structure, perform agreement and other checks, build the parse tree and reference the case frame interpreter. See Felshin (1995) for more on the lexicon and grammar.
- Plans and actions - rules, expressed in terms of classes of objects, that specify how the tutor organizes activity in the microworld, how the student initiates actions and how the microworld state is altered.
- Microworld - a hierarchy of objects with their situational and graphical properties, as well as the associated plans and actions

The procedural level in the third ring shows the processes that operate on the knowledge base to provide language tutoring or a learning environment. The main processes that achieve these functions are:

- Pedagogical Processing - to decide what the tutor will do and say next, by selecting a tutorial schema from the knowledge base and executing it.
- NLP Processing - a generative system which interprets language specifications in declarative form. These specifications are derived from the tutorial schema (incorporating views and microworld information) to construct an abstract syntactic structure, which in turn undergoes grammatical and morphological processing to produce a sentence.
- Situation Reasoning - a plan executor that instantiates the plan and action rules based on the current situation.
- Microworld Processing - maintains object information and performs the graphical actions in the microworld.

The innermost ring represents the system executive which controls the interaction between the student and the system and coordinates the main processes based on this interaction. The student's interaction with the system takes two forms, learning and control. In a learning interaction, the student causes linguistic and graphical events in the microworld. In a control interaction the student can take control of choices about the plans and actions, the conversational style and the degree of linguistic variety. Further details on the operation of FLUENT can be found in Hamburger (1995) and in Schoelles and Hamburger (1996).

The remainder of this section describes the functions of NLP within the FLUENT architecture. We point out their importance in this framework to FLUENT's pedagogical goals. As seen in Figure 2, NLP is involved in three levels of the architecture: the knowledge acquisition level, the declarative level and the procedural level.

FLUENT, like many other natural language generation systems distinguishes between an early or deep phase that determines content and organization (what to say) and a later or surface phase for lexical content, morphology, and syntactic structure (how to say it). The deep phase is driven by the current tutorial schema, which, as noted earlier, coordinates the activities, conversational style and conversational variety.

A tutorial schema is a list of triples, each consisting of a plan or action, an interaction type and a list of views, as in the example in Figure 3. The resulting conversation depends in part on student responses and some randomized choices, but would be something like:

I am picking up a cup. [Tutor puts cup on table.] The cup is on the table. There is another cup on the sink. What is on the table? [Student: The cup.] Good! Pick up a cup. [Student turns on the water(!)] You did not pick up a cup, but you turned on the faucet. Turn off the faucet. [Student does so.] Great! You turned off the faucet. Pick up a cup...

Plan/Action	Interaction	View
Pick-up	Tourguide	Present-Progresv
Put-down	Tourguide	(Location Location-other)
None	Quizmaster	Wh-on-top
Pick-up	Commander	(Command-indef Present-action Command-fail)

Figure 3. Part of a Tutorial Schema (Lesson)

A plan groups together subplans and individual actions to achieve a goal. The plans and actions are flexible in that their arguments are variables which can be bound when the schema is created, or at the time of execution. The sequence of plans and actions determines what to talk about by specifying predicates and arguments.

Another component of a tutorial schema is the interaction type. There are three interaction types that can be specified in a tutorial schema. In Tourguide mode, the system performs the actions as well as generating language about the action or state. Commander mode is similar to Tourguide mode, except that for each action in the plan the system tells the student what to do (i.e. gives the student a command), the student then uses the mouse to perform the action command. The system will respond based on whether the student successfully or unsuccessfully carried out the command. In Quizmaster mode, the system performs an action and then asks a question about the action performed. The student responds by typing in an answer (i.e. a noun phrase) which is checked by the system.

How to talk about the predicates and arguments or what type of question to ask is specified in the view. This data structure, whether selected or created using the view tool, becomes part of the tutorial schema and serves as the input to the surface generator. As an example, a portion of the Command-fail view used in Figure 3 is spelled out in Figure 4. It gives rise to the final portion of the conversation example accompanying Figure 3.

View-Type:	Complex	
Relation:	Contrast	
Subview-1:	View-Type:	Simple
	Polarity:	Negative
	Subtype:	Action
	Tense:	Past
	...	
Subview-2:	View-Type:	Simple
	Polarity:	Positive
	...	

Figure 4 - A Specific View

The view is the interface to the syntactic/semantic component. In a tutorial schema, an interaction type and view can by specified for entire plans, actions within a plan, or individual actions. Thus, at each step in the schema, how to talk about the predicate/argument structure can be specified. Views are also specified to handle error conditions. Some of the parameters specified in a view are the tense, aspect, modality, type of question (Wh or Yes/No), minimum or maximum number of arguments, and definiteness of a noun phrase. Views also specify whether to talk about the action itself or the state of the world before or after the action.

Views mediate between the language teaching functions of the system and the NLP component that performs the actual text generation. This component is the NLG system developed for the Athena Project at M.I.T. by Sue Felshin. It is responsible for morphology, syntax, and some semantic functions. It was chosen because it does provide some semantics and a well-defined language independent interface called an Interlingua Structure (ILS). The ILS is a specification of data structures containing syntactic and semantic information. These structures are built by the knowledge module and passed to the NLG. The specific semantic features that this framework provides is an extensive hierarchical system of theta roles.

Generation of linguistic responses by the system translates from the interlingua structure to Case Frame(CF)-structure to D-structure and to S-structure in separate steps. First, focusing chooses how to refer to objects. CF-structure is then built. Next, D-structure is built by applying the parameters of the English to principles of universal grammar. S-structure is produced by applying code transformation to D-structure. Finally, the S-structure is flattened to produce printable text (Felshin 1995).

A limitation of the Athena framework is the lack of an interface tool to the lexicon. Lexical entries can only be added or changed by editing the source files and recompiling the lexicon. A task that can not be easily performed by a language teacher or exercise designer. Another limitation is that in some instances when variation in surface structure is possible the choice is made randomly thereby diminishing the application's control over the language generated.

4 FLUENT Pedagogy and NLP

Even a flexible, extendible and teacher-accessible system may be of little value unless it is also engaging and beneficial to students. Our initial results, presented in the next section, suggest that it is engaging to the students who have used it. Here we argue that one should expect it to be beneficial as well, in view of its relationship to several pedagogical issues.

Language learning pedagogy is made complex by the great complexity and variety of both languages and learners. It is therefore out of the question to make a universally useful pedagogy-based requirements statement and proceed to the ultimate, correct system design. Our aim in designing FLUENT has thus never been to create a panacea, but we do claim to be developing an important form of language learning interaction. Besides supporting this claim, we also point out the crucial role of NLP in enabling our ambitious approach.

We consider five pedagogical issues. The first is specific to the domain of language: the explicit teaching of language facts versus implicit learning from ordinary conversation about non-language topics. The next two issues - situatedness and error-handling - are of general concern but play out in unique ways in the domain of language. Finally, active learning and the choice of difficulty level are issues that relate to general cognitive considerations and so are relevant for the learning of other domains as well as for language.

The explicitness issue arises from a curious observation: all children succeed at implicit learning of their first language, while many adults fail in attempts to learn a second one. Is it that children spend all day at it, or, lacking a language, are more motivated? Do adults know too much, including another language, or have they lost some language capacity in maturation? Or are they just not properly taught? If it's the teaching, can we adapt the child's immersion experience to adults, taking care to make the situation simple and clear at all times as we do with toddlers?

Success with immersion is not confined to first-language learning. Many an adult who has experienced second-language immersion will testify to its benefits, and Krashen reports classroom benefits with his method. FLUENT provides a conversational setting where a linguistic fact is not presented explicitly but rather can be indirectly acquired by the student through exposure to it in a meaningful context. To learn to communicate, one must sooner or later practice doing so, and FLUENT permits practice in an ongoing and completely understandable situation.

NLP plays the crucial role of providing the variability required by the conversation. Even fairly simple microworlds lead to a combinatorial explosion of possible events and situations, along with a combinatorial explosion of sentences that must occur in appropriate circumstances.

Next, consider the pedagogical proposal that situating a realistic activity in an authentic environment can promote learning both cognitively and motivationally. Without pursuing this issue in general, we note that it has a special significance for language, because the choice of a linguistic construct often hinges on aspects of the current situation, including the conversation itself. These pragmatic and discourse phenomena can be made to arise in FLUENT, since it conducts conversation in an ongoing situation. The resulting practice on these aspects of language can lead to the kind of implicit learning described above.

An example of a situation-based aspect of language that FLUENT presents to students is that the choice of a definite article may depend on there being only one object of some type. Another is that different verb forms are used in descriptions that occur during versus after the action described. An example that relates to the conversation itself, as opposed to the situation, is that the discourse center is a prime contender for pronominal referent. Another is the correlation of imperative, interrogative and declarative forms with the type of conversational interaction.

Pragmatics and discourse are very difficult for NLP. Although FLUENT does not engage in sophisticated NLP-style discourse reasoning, its tutorial schema tool and view tool make it easy for us or a teacher to prescribe the right discourse results without losing computational flexibility at the semantic and syntactic levels.

A third pedagogical issue is the handling of syntactic and semantic errors made by the student. This issue is also related to the explicitness issue. For people and systems that teach linguistic form explicitly, error explanations are paramount. In the communicative approach, errors are tolerated in order to achieve the primary goal of communication. It is argued that explanations are often not well understood and in any case disrupt communication.

A fully communicative system needs robust NLP on the language input side to make some sense of what the student is trying to communicate even in the presence of errors. We have had very positive response from teachers to a different use of NLP, on the generation side. In FLUENT, a misconception on the student's part is often indicated by an incorrect action performed by the student in relation to a command given by the system. In such a case, the system can (if the teacher has so specified) comment on the contrast between the instruction and the actual student action. To do so requires a two-clause sentence, of the form "I asked you to <do[present] action-x>, but you <do[past] action-y>." Generating such responses is clearly a job for NLP. Before getting back to a repeat of the original command, the NLP module generates an instruction to perform an action to undo the original erroneous action.

A fourth pedagogical issue is active versus passive learning. Active involvement by the student has proven successful in many domains. For language learning in particular, TPR and the natural approach (see section 2) let students respond to a human teacher both via language, in the case of questions, and with actions, for commands. In the computational realm, ITLEs use problem-solving environments to afford students the opportunity to act. In its tutorial mode, FLUENT is an ITLE for TPR. Beyond that, in its student mode (and student-driven aspects of tutorial mode too) the student can actively participate by initiating and controlling actions though the mouse.

The NLP component can express these student actions as verbs. The benefit of a NLP approach over a direct mapping between mouse actions and stored text, or even templates with fillers, is that a wider coverage of language can be achieved in an efficient manner. For example, a click on a graphical object could represent the action of picking up that object. To express this in past tense versus present tense only requires a change of a single parameter to the NLP module, whereas with a direct mapping two separate sentences must be stored. Clicking on a different object for a NLP based system again requires only the changing of a single parameter. Moreover, the same object can be described in terms of different properties at various times, according to the situation and discourse. The combinatorial nature of this design clearly permits the system to expose the student to a greater variety of language than would be practical for a non-NLP system.

Finally, a NLP component provides the flexibility to give the student material at the right level of difficulty. In FLUENT, simply by varying a parameter the system can generate a command, a declarative statement or a question. Commands require understanding, but questions are more demanding, in that they require language output from the student as well. Another way to vary difficulty involves verbs with optional arguments. Use of NLP lets us choose whether a sentence is to contain all the arguments that a verb can subcategorize for or only the required ones. Again this is done by changing a single parameter input to the NLP module.

5 FLUENT in the ESL Classroom

FLUENT is currently being evaluated in an English as a Second Language (ESL) environment. The Arlington Education and Employment Program is using FLUENT in its language laboratory. The students in this program are non-English speaking adults. The evaluation results reported in this paper are from two groups of students at level 2 of the program, with limited English and little or no exposure to computer-based systems. What exposure they have had has been with text-based systems mainly for vocabulary. FLUENT is their first exposure to an interactive conversational language learning system. The first group of 14 students generally worked in groups of two or three in a cooperative effort. In the second group, students worked individually. This time, students were so engrossed that they were allowed to work longer, so there was time for only six of them. The most difficult aspect of the interface for the students to master was the use of the mouse. However, the students who used the system a second or third time showed increased proficiency with the mouse.

The evaluation procedure required each student to complete three lessons that had been prepared using the tutorial schema tool, with the advice of their teacher. This was followed by approximately 5 minutes in Movecaster mode to explore other facets of the microworld. All three lessons used the same plan, but progressively harder interaction types: first Tourguide, then Commander, and finally Quizmaster. In Commander mode, a student who did y when told to do x was told, in effect, "You did not do x. Undo y." In Quizmaster, each WH-question required a noun phrase. If the student's answer was not correct, the system displayed a dialog box with a list of possible answers. The student was then to click on the correct choice. If the student made another error at this point - something that happened only once - the system displayed the correct answer.

The students were asked to complete the questionnaire shown in Figure 5. The results are shown in Figure 6. Ten of the 14 students in the first group and all of the second group responded.

1. How well did you like it? It was
   __VERY GOOD __GOOD __POOR
2. How good was it for learning? It was
   __VERY GOOD __GOOD __POOR
3. Was it too hard? Was it too easy?
   __TOO HARD __GOOD __TOO EASY
4. Was the voice OK? __YES __NO
5. Do you want to say more? ___________

Figure 5. The questionnaire

1.	5	VERY GOOD	5	GOOD	0	POOR
2.	5	VERY GOOD	5	GOOD	0	POOR
3.	1	TOO HARD	9	GOOD	0	TOO EASY
4.	8	YES	1	NO	1	<no response>

Figure 6a. Questionnaire Results - Group 1

1.	6	VERY GOOD	0	GOOD	0	POOR
2.	5	VERY GOOD	1	GOOD	0	POOR
3.	1	TOO HARD	4	GOOD	1	TOO EASY
4.	6	YES	0	NO	0	<no response>

Figure 6b. Questionnaire Results - Group 2

Observations made during the sessions point to some strong points. The verbal protocol shows that the students liked the positive responses in Commander mode. When the student successfully completes the action, the system responds with a positive comment such as "Great, you picked up the cup." or "Good, you turned on the faucet." The students also liked being able to pick the correct answer in Quizmaster mode. They often knew the answer but didn't know how to spell it. When they were able to choose the answer they often carefully noted the spelling and were able to spell the word correctly the next time.

Observations made during the first session indicated some weak points. These led us to make changes which apparently led to an even better response in the second session. Several students were frustrated in the first session when the system was doing the action and generating language, because the sequence of actions was occurring too rapidly for them to understand the utterances. They would be concentrating on what was said and their train of thought would be broken when the sentence was overwritten with the text of the next action. This was corrected for the second group by providing a button that allowed the student to advance the system to the next action.

Another annoyance for the student was when a response was given that was correct but not exactly what the lesson required. For example, one action is to "turn on the faucet". When asked "What did I turn on?", several students responded "the water". Although not what the system was expecting (i.e. "the faucet") this response should be accepted. For the second session the system was able to accept various correct answers.

The speech generated is a female voice using the text-to-speech capabilities of the Macintosh. A point of concern to us was how the students would react to this somewhat automated sounding voice. This decision to use text-to-speech was made after consulting with teachers of ESL They feel the quality is good enough for the level of students we are targeting. The student responses show that in general they were not distracted by the quality of the voice. They seem to be concentrating on the syntax and meaning of the sentence more than the pronunciation.

6 Conclusions

NLP has an important role to play in foreign language intelligent tutoring and learning environments. The generativity and variability that NLP brings to the conversational tutor FLUENT are crucial to its meeting its pedagogical goals. In particular, the layered design of the NLP, from teacher specification to actual text is key to allowing teacher control over the language input to the student, while at the same time producing language that is meaningful to the student in the evolving situation.

7 Acknowledgements

Our thanks to the Arlington Education and Employment Program, and especially to Daniel Norton for his support and suggestions.

References

Asher, A. 1977. *Learning Another Language Through Actions: The Complete Teacher's Guidebook.* Sky Oaks Productions, Los Gatos, California.

Caldwell, D. and Korelsky, T. 1994 Bilingual Generation of Job Descriptions from Quasi-Conceptual Forms. 4th Conference on Applied Natural Language Processing, Stuttgart, Germany.

Chanier, T. 1994. Special Issue on Language Learning: Editor's Introduction *Journal of Artificial Intelligence in Education.* Association for the Advancement of Computing in Education, Charlottesville, Virginia.

Chanier, T., Pengelly, M., Twidale, M. and Self, J. 1992 Conceptual Modelling in Error Analysis in Computer-Assisted Language Learning Systems. In M. L. Swartz and M. Yazdani, editors, *Intelligent Tutoring Systems for Foreign Language Learning.* Springer-Verlag, Berlin, Germany.

Clancey, W. J. 1987. *Knowledge-Based Tutoring The GUIDON Program.* The MIT Press, Cambridge, Massachusetts.

Felshin, S. 1995. The Athena Language Learning Project NLP System: A Multilingual System for Conversation-Based Language Learning. In V. M. Holland, J. D. Kaplan, M. R. Sams, editors, *Intelligent Language Tutors: Theory Shaping Technology.* Lawrence Erlbaum Associates, Mahwah, New Jersey.

Hamburger, H. 1995. Tutorial Tools for Language Learning by Two-Medium Dialogue. In V. M. Holland, J. D. Kaplan, M. R. Sams, editors, *Intelligent Language Tutors: Theory Shaping Technology.* Lawrence Erlbaum Associates, Mahwah, New Jersey.

Holland, V. M. 1995. Introduction: The Case for Intelligent CALL. In V. M. Holland, J. D. Kaplan, M. R. Sams, editors, *Intelligent Language Tutors: Theory Shaping Technology.* Lawrence Erlbaum Associates, Mahwah, New Jersey.

Krashen, S. and Terrell, T. 1983. *The Natural Approach: Language Acquisition in the Classroom.* Pergamon, Oxford, England.

Schoelles, M. and Hamburger, H. 1996 Teacher-Usable Exercise Design Tools. In C. Frasson, G. Gauthier and A. Lesgold, editors, *Intelligent Tutoring Systems: Proceedings of Third International Conference*, Springer-Verlag, Berlin, Germany.

Traum, D. R. and Allen, J. 1994 Discourse Obligations in Dialogue Processing. *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, Las Cruces, New Mexico.

Zock, M. 1996. Computational Linguistics and its Use in Real World: the Case of Computer Assisted-Language Learning. In *COLING-96 Proceedings.* Center for Sprogteknologi, Copenhagen, Denmark.