LingWorlds: An Intelligent Object-Oriented Environment for Second-Language Tutoring

Sarah A. Douglas

CIS-TR-93-27
December 1993

Department of Computer and Information Science
University of Oregon

LingWorlds: An Intelligent Object-Oriented Environment for Second-Language Tutoring

Sarah A. Douglas
Department of Computer and Information Science
University of Oregon
(503) 346-3974
douglas@cs.uoregon.edu

To appear in M. Holland, J. Kaplan, & M. Sams (Eds.), *Intelligent language tutors: balancing theory and technology.* Lawrence Erlbaum publishers, 1994.

LingWorlds: An Intelligent Object-Oriented Environment for Second-Language Tutoring

Sarah A. Douglas
Department of Computer and Information Science
University of Oregon

ABSTRACT

LingWorlds is a computer-based research system for teaching beginning aural comprehension skills for second (natural) languages. The results of this research include: 1) LingWorlds, a microworld environment and construction-based authoring system for second-language tutoring which permits students to engage in listening oriented communicative interactions with simulations of real world problems; and 2) new insight into the nature of language learning and teaching gained by computational modeling of tutoring and learning based on a specific pedagogical theory. Both of these have impact on second-language instruction and on the design of other instructional software systems.

1. INTRODUCTION

Oral communication skills—the ability to comprehend and produce oral discourse—are crucial in nearly every educational, business, and scientific setting of language use. Yet the development of oral communication skills remains a difficult theoretical and practical problem, and traditional language teaching approaches regularly fail to help many learners. The central problem addressed by this research project was to design a computer assisted language instruction system which could help beginning language learners develop their aural comprehension abilities. The reasons for targeting beginners as well as listening comprehension were three. First, relatively little software is directed at true beginners. Second, relatively little software is directed at improving listening comprehension. Third, the computer environment is best suited to work in teaching listening.

The LingWorlds project initiated by Russell Tomlin a colleague in the Linguistics Dept. and myself draws on two important and innovative approaches to second language learning and teaching: the communicative and the comprehension approaches. Proponents of the communicative approach argue that successful language learning occurs when the student is provided the opportunity to solve non-language problems using the developing second language (Krashen, 1983; Widdowson, 1978). They criticize traditional language teaching for focusing too much effort on the conscious discussion and manipulation of rules of language usage and not enough effort on the acquisition of the second language grammar through efforts to use that grammar to solve actual communication problems. This philosophy integrates well with the general spirit of artificial intelligence approaches to education wherein teaching and learning are conceptualized as problem-solving processes.

Proponents of the comprehension approach argue that second language learning is enhanced when beginning stages of language learning are devoted to developing the ability to understand the second language. Obligatory oral production, as well as reading and writing, are delayed until the student is able to understand easily utterances in the second language. Delaying production improves student performance in other aspects of language acquisition (Asher, 1966; Asher, 1969; Asher, 1977; Postovsky, 1977; Postovsky, 1979; Winitz, 1981; Winitz, 1973).

Our project embraces both of these complementary approaches to language learning and teaching. The instructional system we have created, called LingWorlds, involves the student in solving communicative problems interactively with the system. The student participates in problem-solving simulations which allow manipulation of objects in a physical scenario or *microworld*. Information about the problem to be solved and information about the microworld are orally given in the second language. Meta-level commentary by the tutor is also in the second language. The teaching intervention in these simulations can vary from highly directed tutoring to coaching to purely student-controlled exploration. LingWorlds uses dynamically generated speech from a digitized phrasal lexicon to produce its side of the tutor-student dialogue. The student's part of the dialogue consists of acts in the microworld.

Our pedagogical model requires many small problem-solving environments to be built, as well as many simple tutoring control strategies. This motivated us to create a high-level authoring system. Our general philosophy in constructing this system is that we wanted the microworlds to be very knowledge-intensive and totally integrated with the interface. We also wanted them to be reusable. These two themes have pushed us to envision a sort of library of microworlds and tutoring components. LingWorlds which is a full object-oriented programming system offers the teacher a rather large amount of programming power, if the teacher wants to use it, while permitting teachers with less experience with the facility to build on simple simulations previously stored in a library.

Before discussing the particulars of LingWorlds as a language tutoring system, a more thorough understanding of the pedagogy on which it rests would help many readers who are unfamiliar with these approaches.

2. COMMUNICATIVE STUDIES OF HUMAN LANGUAGE TUTORING

Perhaps the most fundamental question for this project was posed early on: What is it that tutors using the communicative/comprehension approach teach and how do they decide what to do at any given point in a tutorial interaction? An answer to this compound question is needed if the computer tutor is to perform in a way similar to human tutors. The existing literature in communicative language teaching provides only the most general guidelines on this, guidelines which are not specific enough to incorporate in any way in the tutor system. Consequently, we have devoted considerable effort to understanding the nature of individual tutoring.

The communicative literature provides only the most general of assertions regarding what is taught and how that teaching is accomplished. The communicative approach (Widdowson, 1978; Widdowson, 1979) views language learning as a cognitive enterprise in which the learner entertains multiple hypotheses regarding the structure and function of target language constituents in natural discourse contexts until sufficient contextualized input is encountered to settle on and automate the learner's closest approximation of the native speaker norms. This process of creative construction of an *interlanguage grammar* (Selinker, 1972) is facilitated when linguistic input is comprehensible to the learner (Krashen, 1977; Krashen, 1982), when it is of sufficient quantity in a variety of discourse contexts, and when the affective environment does not constrain exploration and risk-taking (Krashen, 1977; Krashen, 1982; Schumann, 1978).

There are a number of "tenets" of the communicative approach that can elaborate briefly the general characterization provided above. Under the communicative approach language is viewed as situated social activity, as efforts of discourse production and comprehension, as *communication*.. Thus, in communicative language teaching:

- Systematic attention is paid to functional as well as structural aspects of language.
- Classroom work is aimed at the situational and contextualized use of language.
- Teaching and learning are made observable and transparent through content which is made real to the learner through pictures, sketches, diagrams, and other representations.
- Attention is focused on the ability to understand and convey information; i.e. on information transfer.
- The learner is seen a responsible partner in learning rather than as an object to be manipulated.

Language teaching represents the effort by the tutor to set up the conditions for learning described above. That is, with more or less finely grained teaching efforts, the tutor seeks to provide to the learner a sufficient quantity of comprehensible input drawn from a wide variety of genuine or authentic discourse contexts (Johnson, 1982; Krashen, 1983; Widdowson, 1978) in an affectively "supportive" environment.

This approach to early language teaching emphasizes acquisition of linguistic pragmatics, particularly

- Development of lexicon/vocabulary.
- Development of the grammar of spatial relations.
- Development of the grammar of reference including deictic and pronominal expression.
- Development of basic syntax and word order.
- Development of article systems.

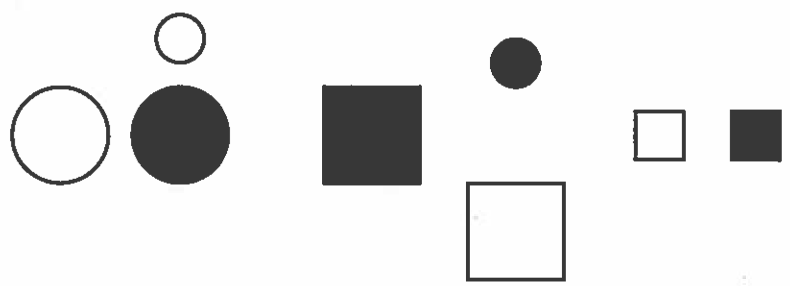
Empirical Study 1: Protocols of Face-to-Face Human Tutor-Student Pairs. While the observations above represent some of the general principles defining the communicative approach, these general principles do little to tell us exactly what teachers manipulate in tutoring and when and how they do it. In our efforts, then, we developed an approach which yielded understanding of what it is tutors do when they teach.

In the first study using a constrained semantic domain, we videotaped human tutors and students from the American English Institute at the University of Oregon. The tutors engaged in one-on-one, face-to-face tutorials with nil proficiency learners of English whose native language was not Indo-European. Five of the students were tutored by a master teacher who has many years of experience in the communicative method; two students were tutored by another tutor who was a relative novice to the method but an experienced teacher of English as a second language. The tutor was directed to teach two dimensional spatial relations using a set of eight cardboard circles and squares differing in two sizes, large or small, and two colors, red or blue. The tutor and student sat across from each other with a table between them. Video cameras captured activities on the table and interaction between the participants. A total of seven dialogues of these interactions were collected on videotape. The protocols varied in length from 20 to 41 minutes. The video tapes were treated as the transcript, and analysis of the discourse was performed by exhaustive viewing of the tapes. It is important to emphasize that this teaching task was previously unknown to both tutors. They were not allowed time to prepare a fixed curriculum. Our reasoning behind this was not only did we want to capture dynamic, interactive natural language, but we also wanted to observe realtime problem solving in a teaching situation.

The analysis of these tutoring sessions follows a discourse model developed by (Sinclair, 1975) but formally specified and elaborated to provide a sufficient description for an ITS system for these language tutoring sessions. The complete description of this method of analysis and formalization is published in (Douglas, 1991). However, a brief description follows to give the reader a general overview.

The tutoring interactions consisted of a number of distinct segments of dialogue that we have termed "exercises" which have a goal of teaching a particular concept of the curriculum, in this case a spatial relation. The length of these exercises varies from 2 to 15 minutes. The more experienced tutor included 6 or 7 of these exercises, which followed a consistent pattern in all her protocols. The novice tutor included 9 exercises in each of her sessions but made major changes in sequence between the first and second learner. Within each exercise occurs what we have termed episodes (or alternatively exchanges) consisting of rhetorical acts. A variable number of acts make up an exercise. No fixed ordering of these acts within exercises was observed, probably due to the dynamic nature of the tutoring situation, where the next act depends on the many factors that make up the control of discourse. A partial order was however evident. Acts are classified into one of eight classes of common motivating factors which are formally defined by precondition states of the tutoring goals and curriculum structure, assumptions about the student model, the context of the objects on the table, and the history of student (or tutor) failure and repair.

For example, suppose we wish to explain the following discourse which begins with a context of circles and squares constructed by previous interaction and shown in Figure 1:



Note: Black objects are "red"; white are "blue".

Figure 1. Protocol example from face-to-face tutoring study.

Tutor: "Now, take the small red circle"
<pause>
Student: <takes the card and moves it to the side of the table>
Tutor: "and put that below the large red circle."
Student: <hesitates>
Tutor: "Below the large red circle."
Student: <pushes the small red circle to below the large red circle>

This segment consists of three episodes denoted by the three separate acts of the tutor with corresponding non-verbal responses by the student. The tutor's first two acts are diagnostic and the third is a repetition of the previous referent phrase for clarification. What is immediately apparent is that the episodes must be coded at a level below the sentence and clause, usually at the phrase or even single lexical item level. A second observation is that many of the interaction cues are non-verbal—consisting of hesitations, intonations, physical actions, etc. Frequently, we observed that tutors broke the sentences into diagnostic units such as the above so that the complexity of identifying where the misconception occurs is reduced. Tutors observed all student actions intently *during their performance* to ascertain if trouble was imminent.

Empirical Study 2: Protocols of Machine-Mediated Human Tutor-Student Pairs. After studying face-to-face human tutors and students and observing the large amount of non-verbal interaction that was present in the interactions, we decided that we needed to study how human tutors adapted to a tutoring situation where, like the computer, the tutor did not have access to visual information on the movements of the learner's face, hands and eyes and had a much more constrained environment. In the second study we programmed a computer manipulable version of the previous cardboard circles and squares task called Flatland. In this version, the tutor was given a teaching environment consisting of an inventory of eight circles and squares (although black and white) which she or he could move using a mouse. The objects were initially located in a "stock" located in the right margin of the screen and separated from the main demonstration area by a vertical line. The student was given a second, separate computer system including mouse. All actions occurring on both computer screens were equally visible by the tutor and student. Tutor and student had oral access to each other but could not see each other. Six native English speaking students participated in this study. Three were taught Indonesian and three Japanese. All tutors were native speaking. A complete analysis of this study has been done by (Ungar, 1987) paying particular attention to more refined analysis of the rhetorical acts, event networks and the role of repetition in teaching with the communicative approach.

From these two studies, we were able to extract significant components and principles of second language tutorials regarding rhetorical acts of communicative/comprehension based language teaching, control strategy of the tutors, handling of tutoring and linguistic failure and repair, the role of non-verbal interaction, and the limitations of existing technology to support this method. These studies provided much of the detailed empirical experience that we needed to grasp the general demands of a computer-based system. Many of these insights are built into the LingWorlds system.

Empirical Study 3: Survey of communicative language teaching targeting listening comprehension examples. Although the previous empirical studies provided us with much needed knowledge about the process of teaching the communicative method, we were still relatively uncertain about what constituted a suitable set of exercises. We searched the pedagogical and theoretical literature in second language learning and teaching and we interviewed practicing second language teachers in order to create an inventory of listening activities. The inventory of activities is organized to reveal: (1) the necessary prerequisite skills required to perform the task, (2) the learning goals for the task, (3) the expected outcomes for the task, (4) the means/methods of presenting/executing the task, and (5) means of evaluating the effectiveness of the task. While no original work was done in this area, it remains useful in organizing suggestions for practicing teachers who might be interested in using the computer tutor system. These insights are reflected in a number of LingWorld's exercises and the LingWorlds teacher's manual.

3. COMBINING COMMUNICATIVE AND COMPREHENSION APPROACHES IN LINGWORLDS

It is clear from the descriptions above that implementing an intelligent tutor would demand a number of compromises. First and foremost, the computer system would not have the ability to assess student physical response other than simplistic mouse actions of pointing, dragging, and button pressing. Secondly, tutoring actions would be confined to focusing student's attention by moving and highlighting objects and simple spoken language. Finally, the sophisticated control structure of instruction that we observed in our empirical studies could not be replicated in any existing computer system since present technology lacks full participation, perceptually and socially, in a conversation. For more explanation of these limitations the reader is referred to (Suchman, 1987). Despite these difficulties, we still felt committed to exploring a more restricted version in which objects could be presented in a context (microworld) through linguistic descriptions of varying noun phrase and speech act complexity generated by a simple phrasal lexicon.

To illustrate the basic concept, let us consider an example of the most simplistic sort. It is well known that for Japanese learners of English learning appropriate use of prepositions is extremely difficult. The following three sentences represent three different states of the world:

(1) Put the baby in the car.
(2) Put the baby on the bus.
(3) Put the baby on the car.

While (2) could mean place the baby on the roof or hood of the bus, its more typical meaning, and the one problematic for many learners, is to place the baby inside the bus. A microworld in which the manipulation of objects [babies or suitcases or whatever] and their relations [in, on, under, and so on] to locations [cars, buses, etc.] is possible permits the student to learn about the semantics of English prepositions through simulated interaction with the microworld environment. The student is not told the meanings of *on* versus *in*; instead the student learns this difference by solving problems which require understanding that semantic difference of the desired outcome [getting the baby inside the bus] is to be obtained. Even though no oral response is required or expected of the student, the underlying grammar of the second language is directly developed as the student learns to understand the second language.

Though this example may seem a trivial one for native speakers of English, it is by no means trivial to beginning language learners. Similar problems clearly exist for all other languages one might desire to learn. And it is possible to build increasingly complex microworlds that challenge the learner with increasingly complex grammar in increasingly complex discourse.

Our work builds on the idea of simulation microworlds, first proposed by (Papert, 1980) for LOGO programming. A later paper by (Burton, 1982) on reactive learning environments urged the pedagogical value of exploratory learning in domains other than programming. Our work has been greatly influenced by research on embedded semantics in microworlds with direct manipulation interfaces: Programming by Rehearsal (Finzer, 1993), ARK (Smith, 1986), the work on STEAMER (Hollan, 1986), Thinglab (Borning, 1981), and Envisioning Machine (Roschelle, 1991).

However, in LingWorlds we build microworlds which are a simulation of the world as represented for tutoring purposes. This places our work closer to the work of Programming by Rehearsal than systems like ARK, STEAMER, Thinglab and Envisioning Machine which represent the knowledge of Newtonian mechanics and hydraulics. We are interested in packing as much "common sense" knowledge as possible into the microworld. Thus we want to allow the system to derive inferences beyond the facts explicitly declared. For example, we want the system to be able to compute spatial relations dynamically. We have found that AI work on scene analysis and diagram understanding, as well as the literature on spatial reasoning and data bases is related to our work. We have pursued our own research into spatial relations in order to provide a useful set of spatial pragmatics for the system (Douglas, 1987). Additionally, while we wish to continue the basic notion of exploratory learning, we also want to introduce more tutor-controlled strategies and ITS computational mechanisms into the microworlds. This in effect turns the microworld into what (Wenger, 1987) terms a knowledge presentation system (in exploratory mode) and an active knowledge communication system (in tutoring or coaching mode).

Although we have concentrated mostly on developing the microworld concept, we have other shared interests with research in ITS. ITS systems can be contrasted with traditional CAI (Computer-Assisted Instruction) in the following ways:

- The problem of learning and, consequently, teaching is seen as a cognitive, knowledge-intensive, and typically problem-solving process rather than a reinforcement process.
- The control structure is dynamically generated by the interaction of curriculum, student response, and heuristics for diagnosis and tutoring rather than simply stored by the program.
- The domain knowledge being taught is explicitly available for pedagogical decisions rather than embedded as numerical calculations (simulations or drill and practice) or blocks of text/images/sound (programmed instruction).
- Empirical, scientifically conducted studies of students and teachers form the foundation for the research, from initial data collection through to evaluation.

Influenced heavily by the cognitive science movement, ITS represent a significant attempt to model the cognitive aspects of teaching and learning, particularly in providing an individualized approach. Our project's empirical approaches to understanding the cognitive and social bases of comprehension and communicative language teaching places us squarely in this camp.

Likewise, the domain knowledge being taught in LingWorlds is explicitly available for pedagogical decisions rather than embedded as numerical calculations (simulations or drill and practice) or blocks of text/images/sound (programmed instruction). In LingWorlds, the characteristics of the domain objects are declaratively represented. This means that, as the object determines its behavior from its characteristics, so too can the tutoring control system make pedagogical judgments based on direct domain knowledge.

4. A LINGWORLDS EXAMPLE: *PROVISIONING THE LIFEBOAT*

In order to provide the reader with a concrete idea of how the system works, the following is an extended example called *Provisioning the Lifeboat*. In the "Provisioning the Lifeboat" simulation, the student is faced with the non-linguistic task of provisioning a lifeboat before an ocean liner sinks. The computer displays an introductory animation showing an ocean liner colliding with an iceberg. A simple oral narrative accompanies the animation: "Your ship has hit an iceberg. It is sinking." An on-deck scene of equipment and people near a lifeboat is then presented (Figure 2). At this point, student interaction begins. In tutoring mode, spoken language directs the student to provisions and equipment needed to use the lifeboat successfully. The student responds to the instructions by pointing at, or dragging objects with a mouse. Only through successful comprehension of the spoken language will the student be able to complete the tasks required to use the lifeboat.

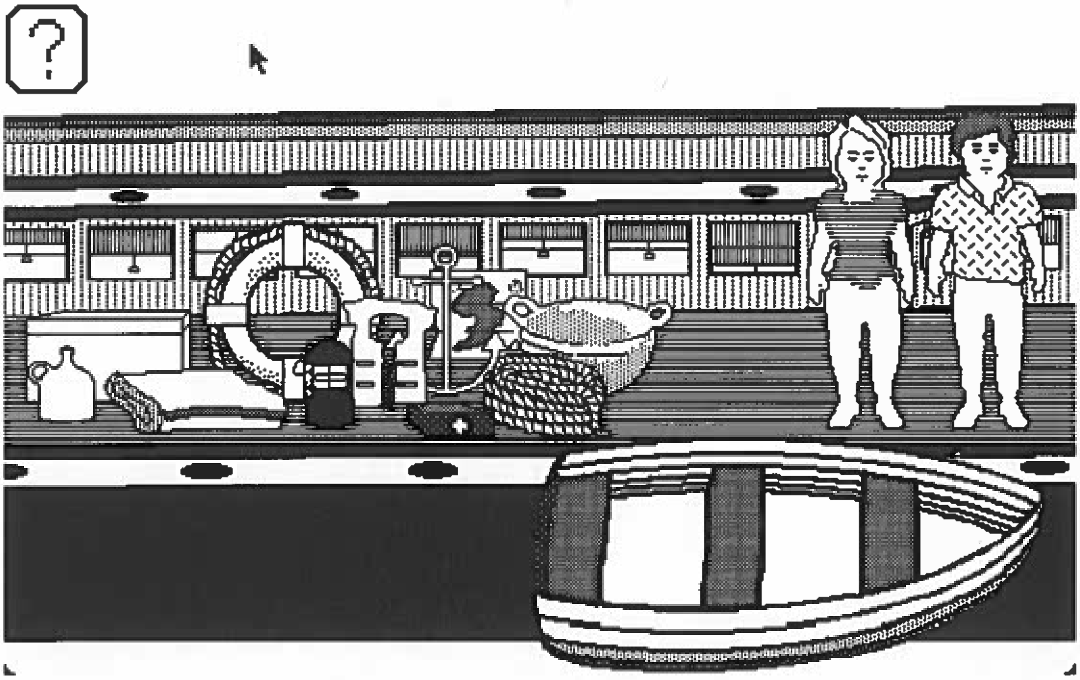


Figure 2. A Lifeboat configuration

As an example, an early, easy problem posed by the tutor is "Put the anchor in the lifeboat." The student uses the mouse to move the cursor to the anchor, attaches the anchor to the mouse, moves the anchor to the rope, and releases it. The student can see the anchor move and when it is released on the deck, the screen appropriately updates. With successful completion of the task, the computer provides the oral response: "Okay, let's keep going!". More advanced problems involve the tutor asking the student to "Put the binoculars in the basket in the lifeboat and put the waterjug beside them." Note that in this problem the student must solve several complicated problems: know the names of the objects, perform the task in the correct temporal order, select which of the two baskets is the correct location, and understand prepositions of location (beside, in, etc.).

Three versions of *Lifeboat* are described below which vary by the mix of student and tutor initiation and control complexity: an exploratory (student-initiated), a coaching, and a directed tutor. These control strategies can be combined in a single lesson. This demonstrates the kind of versatility that we want to provide, although it is a poor substitute for the kinds of complexity we observed in human tutor-student interactions.

Exploratory Lifeboat. In a totally exploratory mode, the student can use the mouse to single-click objects and hear their names ("*The basket.*") or double-click objects to hear a linguistic description of their locations in relation to another object ("*The woman is not in the lifeboat.*") The student is also free to drag objects from one location to another to hear the effect of a changed location. Help instructions are available by clicking a question mark (?) icon.

Coaching Lifeboat. During a game format, oral commands direct the student to locations of provisions and equipment needed to provision the lifeboat successfully. The sinking of the ship provides time pressure for completion of the tasks. The system keeps a score of successful tasks.

Tutored Lifeboat. The directed tutor mode supplements the oral commands of the game mode with remedial intervention. The control strategy that we used is derived from protocol studies of the human expert tutors described in section 2 above. The tutor maintains a curriculum of concepts to be taught and a differential student model to determine state transition information and diagnose types of errors. If the student fails to move any object after a few seconds, the command is repeated. A second such failure causes repetition of the instructions for the overall task; after the third failure the system demonstrates the action. If the student moves an object to the wrong place, it is returned (by the system) to its initial location and the command is repeated. After several failures, the system will move the object to the appropriate location, demonstrating the task. The student is then given another opportunity.

5. THE LINGWORLDS SIMULATION SYSTEM: IMPLEMENTATION

The LingWorlds system is illustrated by functional parts in Figure 3. The microworld component is essentially all object-oriented, as is the tutor component.

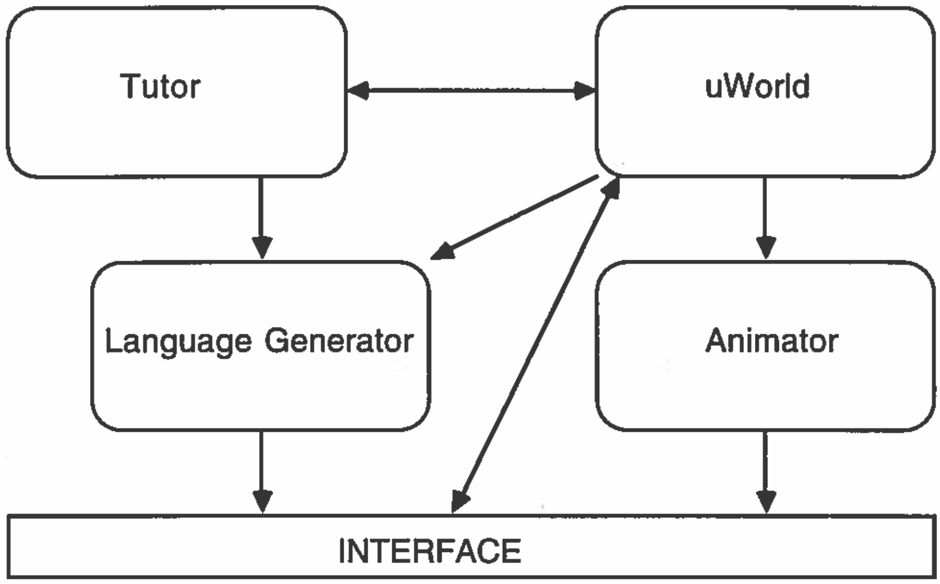


Figure 3. Functional Description of LINGWORLDS

The following briefly describes how LingWorlds works. Readers wishing a more complete and technical description are referred to (Douglas, 1992).

Contexts and Scenes. Each lesson or exercise consists of a sequence of contexts. Each context is either a "movie" sequence or else a "scene". Within a context, a series of movies or scenes may occur. Each scene can be composed of subscenes appearing simultaneously in the same window or in recursive windows. A scene is simply a set of problem-solving objects for the student to manipulate and to focus attention on. A scene may have a background which is simply a bit-mapped picture and not a full-fledged object.

Interactors. Each object represented in a scene is a highly specialized object called an "interactor" object. Interactors in the *Lifeboat* microworld (Figure 2) include the rope, the anchor, the lifeboat, and the question mark (?) icon. Interactor instances have various attributes, actions and relations with other interactors. Interactors are usually visible as a bit-mapped picture and arranged in x-y locations during the initialization of a scene. They can also be visible or invisible; draggable or not; highlighted or not.

Interactor Actions. Interactors can respond to both user actions, such as mouse clicks or dragging, and system actions such as message passing or tutor generated events. Interactors can highlight themselves and they can speak. System animation for the interactors can result from following any of three methods: an arbitrary path stored by the designer, a computed trajectory, or a location described by a natural language spatial relation, which allows a description of motion by displacement. It should be emphasized that these interactors, while representing concrete entities in the real world, do not manifest the physical laws of that world. Interactors do not "fall" under the influence of gravity, unless caused by the system designer. In other words, the microworld is more of an imaginary, linguistic world. Each interactor responds to 1-click, 2-clicks, press-and-hold, and press-and-drag mouse actions by the user. The designer can choose to enable or disable these events and then write a method using a special language which describes how the interactor should respond. Thus the system can enable drag actions on some interactors and ignore them on others.

Interactor Attibutes and Semantic Properties. Each interactor has a set of attributes which determine whether it responds to different user interface events and also semantic features that define it for linguistic purposes. These features include whether or not the interactor is animate, a person, an artifact, a vehicle, a vehicle of public transportation, or a container. Any other attribute-value pairs can be added by the designer. For example, an author could specify the interactor's color or the lexical item that designates its name. In a Japanese version of *Lifeboat*, animacy features were added to interactors for purposes of generating the correct morphemes during speech generation. Each interactor can also be decomposed into a subset of other interactors. This expresses the has-parts relation. For example a human interactor can be decomposed into further interactors of arms, legs, torso, and head. This relation can also be used for generalized possession as well. Interactors can also have ports which cause auto-message passing between them when they are contiguous.

Graphics and Spatial Relations. Each interactor has a graphic image and an initial position. Interactors are located on planes with each interactor occupying its own plane. Interactors can swap planes dynamically. This provides the scene with a simulation of three-dimensional reality as interactors are animated by the system or dragged by the user. The animator is a specialized commercially available program for running animation sequences as "movies" of previously digitized data rather than generating them directly in LingWorlds. However, the LingWorlds objects are themselves capable of animated sequences under direct program control of x-y locations and planes.

Probably the most difficult and interesting part of the system is the set of spatial relations that we have very thoroughly studied and built into the system. We focused particularly on the spatial relations used in prepositional phrases. These constitute a "common sense" description of space. We conducted extensive protocol analyses of human tutors using simple geometric shapes to teach the relations *left*, *right*, *above,below*, and *between*. (see Section 2). Based on these protocols, our own intuitions, and other psycho-physical experiments, we have developed algorithms to compute these relations in a manner similar to native English speakers (Douglas, 1987). These spatial relations algorithms have been incorporated into LingWorlds. Typically, in tutoring mode the system might move three objects into the main frame and then ask the student to touch one of them, based on salient descriptive characteristics. Although this was done initially for the pedagogical goal of teaching spatial relations involving prepositions, it has had interesting side effects for the overall design of the system. For example, it allows interactors to locate themselves in a scene relative to a spatial relationship with another interactors.

Our initial work with spatial prepositions included those in the essentially horizontal and vertical two-dimensional planes: *left of*, *right of*, *above*, *below*, and *between*. (Because of issues hinted at previously, our simulations are primarily two dimensional but because of the possibility of overlapping and movement, they approximate three dimensions. At this time we have yet to implement the depth plane prepositions: *in front of* and *in back of*.) Attempting to develop a complete computational model of human spatial relations language, as most readers know, is as yet an unsolved, if not intractable problem. Readers wishing more background on this problem are referred to (Herskovits, 1986; Talmy, 1983). However, because of the highly limited spatiality of the computer screen we were able to develop algorithms which primarily relied upon visual presentation elements of the objects, for example, their centroid.

In order to implement spatial relations in our tutor, we compute on demand for each interactor several spatial properties: center-of-area (centroid), distance, areas that project from edges, and angular displacements. These properties, as implemented in some rather straightforward message-passing algorithms, have sufficed to compute successfully the relations *left*, *right*, *above*, *below*, and a modified two-dimensional *in*. In some linguistic tasks proximity becomes a crucial factor for the computation of these relations and appears as the most difficult component to determine in the general case, since it is clearly dependent on social and psychological factors. We also implement certain spatial properties as features attributed to an interactor. This allows for the specification of a container feature for "in" relations.

This is not to suggest that there were no difficulties. A major problem for us was where to put the deictic origin of the speaker. Is the speaker, in this case the tutor, describing the scene from the position of the student looking at the scene on the CRT screen, or is the speaker sitting opposite the student? An argument can be made for either case. Certainly the voice is coming from an entity seated across from and facing the student. On the other hand, people using deictic expressions can always see the location of the speaker and make the relative orientation adjustments between what the speaker sees and what the hearer sees. In the case of the computer, one is simply not sure where the speaker is. The least ambiguous assumption is that the descriptions are from the point of view of the speaker in a position of the viewer since the location of the speaker isn't known. Thus spatial relations are computed for a scene from the deictic perspective of the student. In addition to the issue of point of view, as noted above, it is the case that judgements of these relations can vary depending on shape, size, discourse task, and contextual arrangement. There also may be prototype positions for some situations.

Since our primary goal is to allow generalized tutors for various languages, it seems reasonable to allow the teacher/author to change the semantics of spatial relations as they may vary by languages. While this might seem trivial to mono-lingual English readers of this paper, it can be a tricky and difficult problem. For example, imagine a bowl which contains an apple. Both English and Chinese speakers would say "The apple is in the bowl." Now imagine that the bowl is turned upside down (hopefully you imagined the first one in its canonical orientation). English speakers would now say "The apple is under the bowl," (even though it is still within the containment of the bowl), but Chinese speakers would still say "The apple is in the bowl." Clearly the same problems occur in instances like "Get on the bus." in English, which French speakers would imagine as someone on top of the bus. These semantic differences suggest that the knowledge representations for spatial relations will have to be open to change by the authoring system. This could be an exceedingly difficult goal to achieve.

Speech Output. The speech output of LingWorlds is activated by user generated, tutor generated or other internal events directed to interactors. Methods executed in response to these events may contain the "say" programming statement which must refer to an entry by label in a stored phrasal lexicon. A phrasal lexicon is speech consisting of units as small as a single word or as large as a phrase (Fischer, 1988). In developing a LingWorlds lesson the teacher must predetermine what the units of speech will be. Items in the lexicon can vary from entire phrases such as "in the lifeboat" to individual words with varying intonations such as sentence final. For example, in the Lifeboat problem, "the rope", "the woman" and "in the lifeboat" are all entries in the lexicon. Using entire phrases reduces the number of entries in the lexicon and reduces processing time while dynamically concatenating lexicon units during runtime. Since interactors can contain semantic features future extensions of the system include adding a case-frame semantic grammar and processor. A separate processor would be necessary for each language.

Because of the demands for speech quality and non-English languages in a language tutoring program, we decided to use digitized human voice recorded under studio conditions rather than any of the available speech synthesis systems(Streeter, 1990). After preparing the application vocabulary, a trained speaker records each of the lexicon entries, whether they be phrases or single words in a studio. Care is taken to make intensity and inflection as appropriate as possible often by embedding the desired lexicon in complete continuously spoken sentences. If there are differing inflections of the same word or phrase, these are all separately recorded. For example, "the rope" as in "The rope is to the right of the jug." versus a question inflection such as "Is the jug to the right of the rope?" The recordings are edited using a commercially available digital waveform editing program to excise the predetermined phrases and words and to remove silent portions between phrases. Items may need to be adjusted to ensure proper amplitude levels among lexicon entries. The edited lexicon is stored on disk as digitized file entries. On output the phrases and words are concatenated and converted from digital to analog form.

Integrating the Microworld with the Tutor. The preceding description of the microworld components describes how program code is modularized by what is essentially display and control of the interface. Since much of this is generalizable, it is highly productive to have it available as part of the class definitions. However, as just described, the basic control appears to be primarily student initiated. If the student clicks with the mouse on a particular interactor on the screen, a method is appropriately activated which may move the interactor, cause it to say something, etc. Since the programming language for these methods has conditionals, it is possible to make an object quite context sensitive in its response to an event. This is the basic flavor of all exploratory learning environments. It is the case, though, that we often want to introduce more teaching intervention into student actions. Thus, ITS systems vary along a continuum from totally event-driven exploratory environments, to coaches which are embedded within a game structure, to goal-directed tutors which have a highly specified control structure. Teaching expertise includes how to tutor, what instructional approach to use, and why and how often to tutor the student.

Control from the tutor is introduced into LingWorlds by the tutor object which is global to the interactors in the microworld. In LingWorlds the control is knit together by message-passing between interactor objects and the tutor object. A totally exploratory microworld has its control locally defined with the behavior of each object tied to student actions. A game microworld increases tutor control. Finally, a goal-directed tutor microworld controls most of the interaction through a task-based agenda. Even in the goal-directed tutor, interactors still retain individual control over the semantics of their own actions. For example, the tutor can be notified that a particular interactor has been dragged, and through additional message-passing with the interactor determine the location. Thus, the object-oriented paradigm allows an event-driven format that accommodates user-initiated actions as well as internally triggered actions.

6. THE LINGWORLDS SIMULATION SYSTEM: AUTHORING

Our system, like Programming by Rehearsal, ARK, ThingLab, and STEAMER, is almost entirely object-oriented. We explore the utility of several important concepts in this context:

- Object Oriented Paradigm
- Direct Manipulation Programming Environment
- High-Level User Interface Programming Language

We believe that the combination of these features has resulted in a uniquely powerful blend of flexibility and simplicity, allowing non-programmers to construct a wide range of complex interfaces.

Object Oriented Paradigm. Previous work comparing rule-based with object-oriented ITS tutors(Douglas, 1986) suggested that there are significant advantages possible with an object-oriented implementation, not the least of which is the accommodation of event-driven, student-initiated control with event-driven, tutor-initiated control.

In LingWorlds every functioning interface object *is* an object in the programming sense. Unlike most object-oriented languages, QUICK's objects are prototypical rather than classed (Borning, 1986; Borning, 1981; Ungar, 1987). That is, every object is created with all the characteristics of the basic prototypical object. This prototypical object, called an interactor, has a potential for a graphical image, animated movement and responding to user interface actions. There is no notion of inheritance or class-based specialization. This design choice reflects a decision to support the initial bottom-up design of interface objects from a set of fundamental parts as a construction kit (Fischer, 1988), rather than as a top-down taxonomic specialization using inheritance. We feel that this approach lends itself to cognitive simplification. At the same time, we provide strong support for the two primary advantages of the class-based approach, abstraction and reuse. Abstraction is supported in that the user may define new "classes" by aggregating groups of objects, thereby defining a new type of object. Reuse is supported in that objects (simple or aggregate) may be duplicated.

Direct Manipulation Programming Environment. The LingWorlds system is developed as a rapid prototyping environment with a easy switch between run and authoring environments. Typically, the author creates images using a painting program, and then imports these in the LingWorlds system to create a background and the images for interactors. The programmer works directly with these objects through a direct manipulation programming environment. Since interactors are visible they can be positioned in the scene and animated by directly moving the mouse (as opposed to specifying the x-y coordinates symbolically). Figure 4 depicts the menu-based programming choices for a selected interactor.

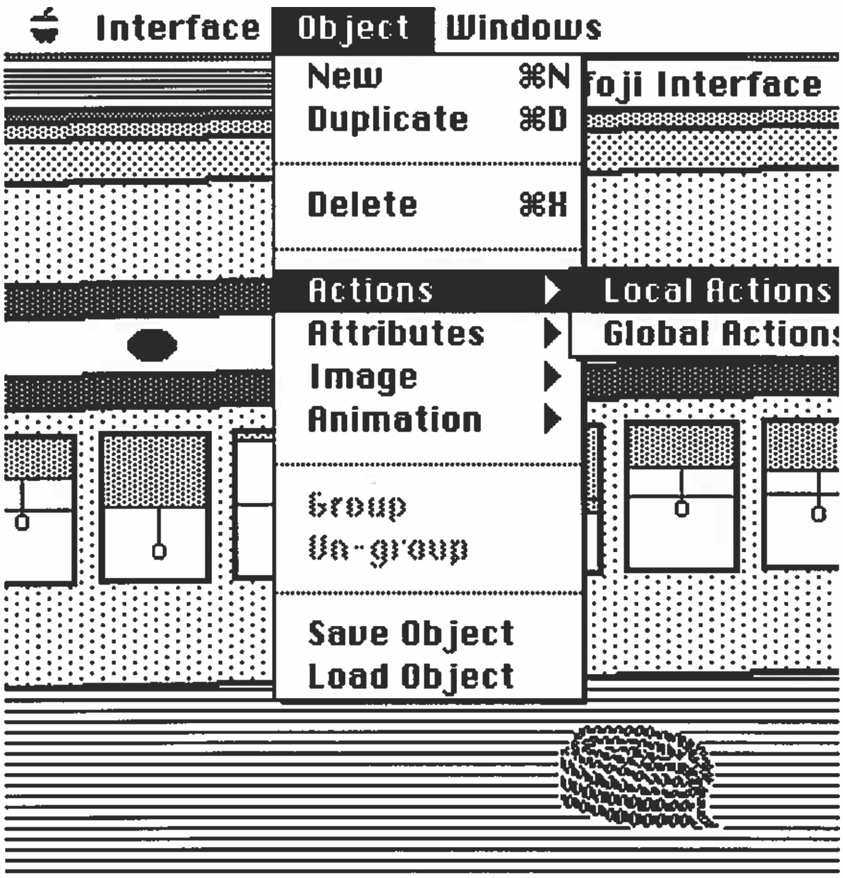


Figure 4: Programming options for an interactor

Please note that in Figure 4 in addition to options for programming interactor actions and attributes, there are basic options for creating, deleting, duplicating, saving (to a file) and loading (from a file) any interactor. This supports the reusability of interactors between microworlds.

High-Level User Interface Programming Language. Actions for interactors are created using a relatively simple English-like language. Methods for the tutor can be created using the same language. The following are the basic commands, functions, and control structures of the language.

Animate <list of objects>
Flash <object>
Highlight <object>
Unhighlight <object>
If <test> <then> <else>
For-each <list of objects> <code>
Set-attribute <object> <attribute> <value>
Say <list of sounds>
Move <object> <location>

Trigger-action <object> <action>

Despite the simple nature of the QUICK language, programming would still be error-prone for novice users in a free-form environment. Thus, we provide a hierarchically structured editor that at each stage presents the user with only *legal* options to choose from. In fact, the only time the user is actually asked to type anything is when an object or a file needs to be named. Figures 5 and 6 below give the flavor of the structure editor.

The language is fully extensible so that higher level actions can be composed from these elements and added to the system. For example, Figure 5 depicts a new action called "arrange."

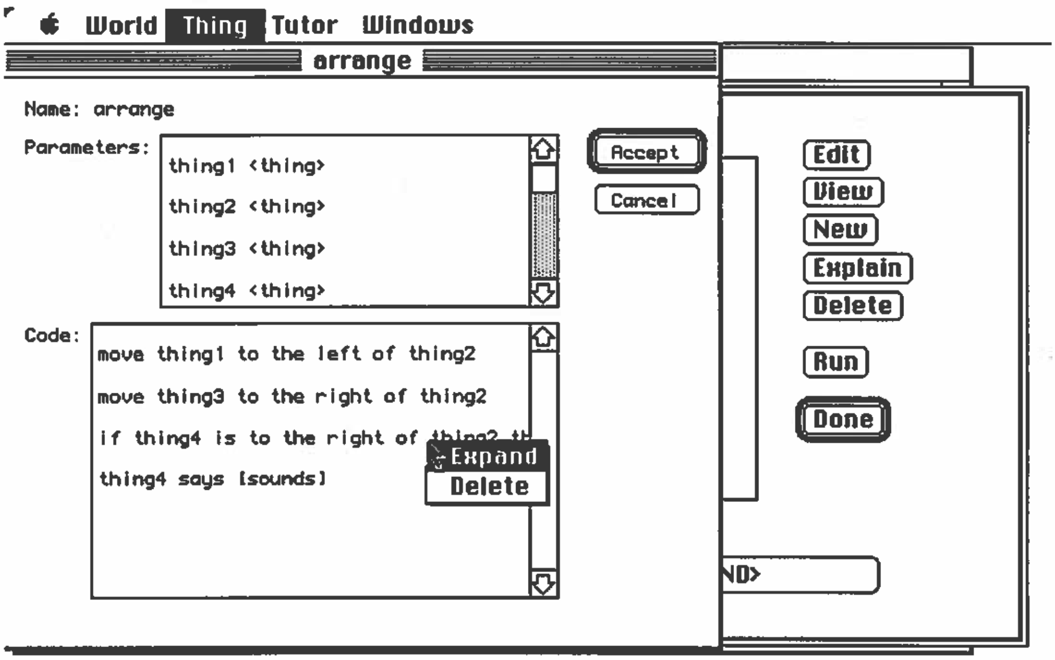


Figure 5: Programming an action

Actions determine the response of an interactor to events generated by user actions (mouse clicks, dragging), other interactors and the tutor object. In Figure 6, we are in the process of programming the *1-click-action* of the **Rope** object of the lifeboat application to announce its name. The stacking of the dialogs illustrates the hierarchical nature of the editor. The user selects the *1-click-action* to edit, is presented with a code dialog, selects **say** from a popup menu (no longer visible in the figure), is presented with a specialized dialog for **say**, and selects the (initially empty) list of sounds to work on. At the moment of this snapshot, the user has just depressed the mouse to display a popup menu of available sounds and has selected the digitized recording of "The rope."

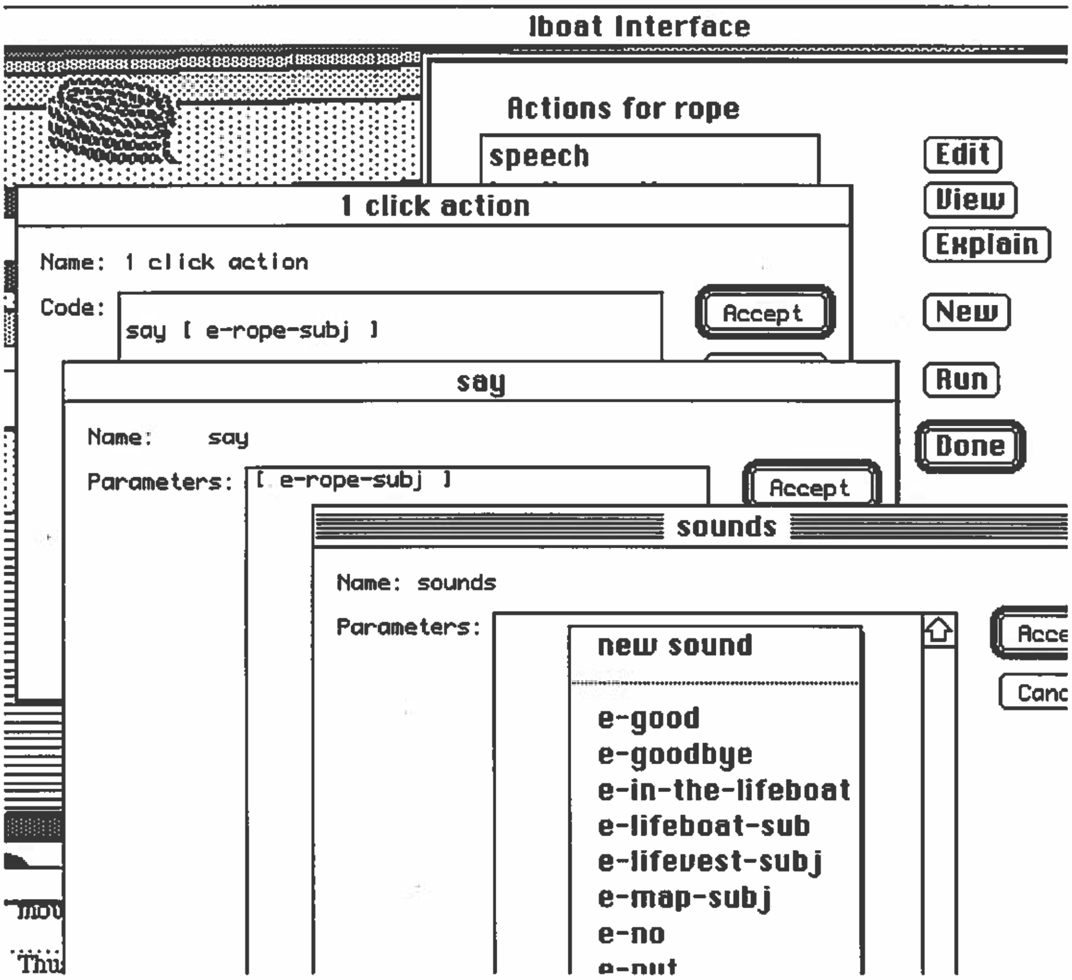


Figure 6: Programming a spoken phrase

Thus, the structure editor combines the *fill-in-the-blanks* and *multiple-choice* paradigms to create a highly constrained programming environment which is easy to use for teachers constructing microworlds.

7. OTHER LINGWORLDS SIMULATIONS

We have built both an English and Japanese version of "Provisioning the Lifeboat", with all the expected savings in programming, since interactors and tutors can be reused for any particular microworld. We built the exploratory version first, and then added the directed tutor version. A game version has not been programmed, but would be trivial. The Japanese version of the "Lifeboat" is virtually identical to the English except for the addition of animacy features required for speech synthesis.

In addition to "Lifeboat" described above, we have created an inventory of other pedagogical microworlds.

FlatLand. In this simulation, the student builds simple two-dimensional configurations from a limited set of objects: black or white, large or small, circles or squares. The tutor initially trains students to deal with the lexical items needed to identify individual objects. It then introduces the specific spatial relations manipulated (*above*, *below*, *left*, *right*, *between*). Finally, the student is directed to build configurations of these objects like the one shown in Figure 1. This is done by placing one object at a time in its proper location in response to a single, complex utterance in the target language. The implementation of Flatland took less than two weeks, since almost all of the code was reusable from the Lifeboat problem.

Mystery world. This simulation places the student in the position of a witness to some set of events. The events are presented as an animated film of some kind or other. Associated with the film is a descriptive sound track which relates the actions witnessed by the student. This narration, whose contents are provided by the teacher, can precede, follow, or occur simultaneously with the movie.

The student engages in two kinds of tasks. First, the student practices ordinary listening, matching the narration with the ongoing events portrayed in the movie. Second, and more interesting, the student can then interact as witness to the computer tutor's detective to solve a mystery presented in the movie. The detective can interview other witnesses in the scene and the student must listen and verify the truth and accuracy of those interview responses. In addition, the detective can interview the student directly, through the judicious use of yes-no questions. This interaction can take the student through the traditional hierarchy of question types in listening: (1) questions of observed fact, (2) questions of inferred fact, (3) questions of inferred motivations, and (4) questions of evaluation.

Shopping Mall involves the task of shopping for a list of items by exploring a shopping mall containing different stores. The student is given a limited amount of money to spend. Each store can say its name and be opened into a new window showing the contents of items. Each item can say its price. The student can choose to buy an item by dragging it to a shopping cart.

MapTiles allows the system to construct a map from tiles containing streets, intersections, houses, etc. The student is given instructions to follow a particular set of directions in order to find some object. The system can determine the student's understanding by requesting that the student point to a current location.

Build! gives the student instructions in assembling objects from simple parts. The system can ``point" to the object by moving the cursor and by highlighting it. The student builds the object by dragging the part into position. A simple problem would be to build a house. A more elaborate problem would be to build a plumbing system. Again, each part is an actor and the system can always determine the correct (symbolic) position of the part by querying for its spatial relation(s).

8. CONCLUSIONS

For those of us interested in computer-based language teaching LingWorlds represents a major, comprehensive attempt to integrate available computer technology with a real language teaching pedagogy, that of the communicative and comprehension approaches. From the point of view of the student, we believe that the system we have developed is challenging and interesting. The student cannot accomplish the needed tasks simply through knowledge of the world, but must comprehend the second language utterances. That is, the student is engaged in true communicative behavior, using the developing second language to solve meaningful, non-language problems–the essence of the communicative approach. The student demonstrates mastery of the linguistic task by physical manipulation of the world rather than by linguistic production. Thus, the student works on aural comprehension, the essence of the comprehension approach to second language learning.

Future work in LingWorlds will be on extending the simpler semantics of object behavior to issues of common sense causality as well as time. We are also interested in adding a simple case frame speech generated. Finally, we need to begin an evaluation period with real language learners.

9. ACKNOWLEDGMENTS

I would like to thank Russell Tomlin, David Novick, Gary Smithrud, Suzanne Shroyer, Kira Carroll, Sue Olivier, Sharon Collins, Joe Clifford, and Brad Tollefson for help in programming, protocol analysis, pedagogical analysis, and imaginative creation of some microworlds. Research for this project was sponsored under FIPSE grant #84.116C and NSF grant DCR-8612578.

10. REFERENCES

Asher, J. (1966). The learning strategy of the total physical response: a review. *Modern Language Journal*, *50*, 79-84.

Asher, J. (1969). The total physical response approach to second language learning. *Modern Language Journal*, *53*, 3-17.

Asher, J. (1977). *Learning another language through actions: The complete teacher's guidebook.* Los Gatos, CA: Sky Oaks Publications.

Borning, A. (1986). Classes versus prototypes in object-oriented languages. In *Proceedings of the ACM / IEEE Fall Joint Computer Conference*, . Dallas, TX:

Borning, A. H. (1981). The programming language aspect of ThingLab, a constraint-oriented simulation laboratory. *ACM Transactions on Programming Languages and Systems*, *3*(4), 353-387.

Burton, R. R. &. B., J.S. (1982). An investigation of computer coaching for informal learning activities. In D. S. &. J. S. Brown (Eds.), *Intelligent tutoring systems.* London: Academic Press.

Douglas, S. A. (1986). Prospects for an ICAI authoring system: A review of computer-based design. In *Proceedings of the IEEE Systems, Man, and Cybernetics Conference*, . Atlanta, GA:

Douglas, S. A., Novick, D.G., and Tomlin, R.S. (1987). Consistency and variation in spatial reference. In *Proceedings of the Ninth International Conference on Cognitive Science*, . Seattle WA:

Douglas, S. A. (1991). Tutoring as interaction. In P. Goodyear (Eds.), *Teaching knowledge and intelligent tutoring* Norwood, NJ: Ablex.

Douglas, S. A., Doerry, E. and Novick, D. (1992). QUICK: A tool for graphical user interface construction. *The Visual Computer*, *8*(12), 117-134.

Finzer, W. a. G., L. (1993). *Rehearsal world: Programming by Rehearsal.* In A. Cypher (Eds.), *Watch what I do: Programming by demonstration* (pp. 79-100). Cambridge, MA: MIT Press.

Fischer, G. a. L., A.C. (1988). Construction kits and design environments: Steps toward human problem-domain communication. *Human-Computer Interaction*, *3*(3), 179-222.

Herskovits, A. (1986). *Language and spatial cognition: An interdisciplinary study of the prepositions in English.* Cambridge: Cambridge Univ. Press.

Hollan, J., E.L. Hutchins, T.P. McCandless, M. Rosenstein, and L. Weitzman (1986). *Graphical interfaces for simulation* No. ICS Report 8603). Institute for Cognitive Science, UC San Diego, La Jolla, Ca.

Johnson, K. (1982). *Communicative syllabus design and methodology.* Oxford: Pergamon Press.

Krashen, S. (1977). *Second language acquisition and second language learning.* Oxford: Pergamon Press.

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

Krashen, S. a. T., T. (1983). *The natural approach.* San Francisco: Alemany Press.

Papert, S. (1980). *Mindstorms: Children, computers and powerful ideas.* New York: Basic Books.

Postovsky, V. (1977). Why not start speaking later? In M. Burt (Eds.), *Viewpoints on English as a Second Language* New York: Regents.

Postovsky, V. (1979). Effects of delay in oral practice at the beginning of second language learning. *Modern Language Journal*, *58*, 229-239.

Roschelle, J. (1991) *Students' construction of qualitative physics knowledge: Learning about velocity and acceleration in a computer microworld.* Ph. D. Dissertation, U.C. Berkeley.

Schumann, J. (1978). Social and psychological factors in second language acquisition. In J. Richards (Eds.), *Understanding Second and Foreign Language Learning: Issues and Approaches* Rowley, MA: Newbury House.

Selinker, L. (1972). Interlanguage. *International Review of Applied Linguistics*, *10*, 209-231.

Sinclair, J. M., Coulthard,R.M. (1975). *Toward an analysis of discourse.* London: Oxford University Press.

Smith, R. (1986). The Alternate Reality Kit. In *1986 IEEE Computer Society Workshop on Visual Languages*, . Dallas TX:

Streeter, L. (1990). Applying speech synthesis to user interfaces. In M. Helander (Eds.), *Handbook of human-computer interaction* (pp. 321-343). Amsterdam: North-Holland.

Suchman, L. (1987). *Plans and situated actions: The problem of human machine communication.* Cambridge: Cambridge University Press.

Talmy, L. (1983). How language structures space. In H. &. A. Pick L. (Eds.), *Spatial Orientation: Theory, Research and Application* New York: Plenum Press.

Ungar, D. a. S., R.B. (1987). SELF: The power of simplicity. In *Proceedings of the Object-Oriented Programming Systems, Languages and Applications Conference*, . Orlando, Florida:

Wenger, E. (1987). *Artificial intelligence and tutoring systems: Computational and cognitive approaches to the communication of knowledge.* Los Altos, CA: Morgan Kaufmann.

Widdowson, H. G. (1978). *Teaching language as communication.* Oxford: Oxford University Press.

Widdowson, H. G. (1979). *Explorations in Applied Linguistics.* Oxford: Oxford University Press.

Winitz, H. (Ed.). (1981). *The Comprehension Approach to Foreign Language Instruction.* Rowley, MA: Newbury House.

Winitz, H. a. R., J. (1973). Rapid acquisition of a foreign language by the avoidance of speaking. *International Review of Applied Linguistics, 11*, 295-317.