Northwestern University

The Institute for the Learning Sciences

SIMULATION-BASED LANGUAGE LEARNING:

AN ARCHITECTURE AND A MULTIMEDIA AUTHORING TOOL

Technical Report # 30 • June 1992

Enio Ohmaye

Established in 1989 with the support of Andersen Consulting

NORTHWESTERN UNIVERSITY

Simulation-Based Language Learning:

An Architecture and a Multimedia Authoring Tool

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY
Field of Computer Science

By

ENIO OHMAYE

EVANSTON, ILLINOIS
June 1992

This work was supported in part by the Defense Advanced Research Projects Agency, monitored by the Air Force Office of Scientific Research under contract F49620-88-C-0058 and the Office of Naval Research under contract N00014-90-J-4117, by the Office of Naval Research under contract N00014-89-J-1987, and by the Air Force Office of Scientific Research under contract AFOSR-89-0493. The Institute for the Learning Sciences was established in 1989 with the support of Andersen Consulting, part of The Arthur Andersen Worldwide Organization. The Institute receives additional support from Ameritech, an Institute Partner, and from IBM.

Abstract

Simulation-Based Language Learning:

An Architecture and a Multimedia Authoring Tool

Enio Ohmaye

Foreign language instruction typically gives students (1) limited interactions with native speakers, (2) limited exposure to the target culture, and (3) limited individual instruction and feedback. Consequently, students are often incapable of interacting with native speakers in a foreign language even after extensive instruction. Computer-based multimedia environments can give learners these crucial experiences. I introduce Dustin, a language learning environment, that incorporates simulated interactions with native speakers, exposure to the target culture, and individualized attention in a cohesive model of instruction -- the apprenticeship model. I also introduce an authoring tool, MOPed, to implement Dustin-like systems. Understanding, organizing, and maintaining the complex knowledge networks necessary to implement Dustin-like systems requires tools to help us visualize, reuse, and contextualize information. MOPed provides a mechanism to group and reuse information and visual aids to improve readability.

Acknowledgment

Roger C. Schank, my advisor, has created an amazing research lab. ILS combines a vibrant community, constantly generating ideas, with the manpower and equipment necessary to turn ideas into software, all with a healthy touch of realism. Roger gave me access to these resources and zealously coached me through my Ph.D. program. By example, he has taught me about thinking, writing, presenting, designing, arguing, and enjoying it all. Most of all, Roger inspired me with his passion, and by living his bliss, he inspires me to seek my own. Thanks for coaching me through this journey Roger, and thanks for the privilege of being a part of your life.

Allan Collins has taught me about education and academic life. He has always pushed me to reflect upon my experience and articulate my insights. Though often showing me faults in my reasoning, Allan has the gift of always making me feel ok about not knowing. Larry Birnbaum showed me how to get excited about ideas. His emotional involvement with his intellectual life deeply influenced the way I feel about ideas -- either I love them or I hate them, but never do I feel luke-warm about them. Chris Riesbeck helped me think about software design. He always rescued me when I was stuck with an implementation problem.

The work described here was done by a team. Vitas Daulys helped design and nurture the seeds of Dustin from day one, and worked on some of everything during its development. Beth Beyer took over whenever we had to deal with people, keeping us in touch with the real world. Mark Chung did an excellent job coding, skillfully engineering most of Dustin and MOPed into what they became. Faina Mostovoy contributed ten years of experience teaching English as a second language. As a team, this extended family provided each member with encouragement, room for commiseration, and many reasons to celebrate. Thanks guys for the barbecues, concerts, work-outside days, work-late days, and frozen yogurts. And most of all, thanks for a superb job -- if I may say so myself.

Other people joined the team and helped us along the way. Angel Ohmaye helped us produce the videos, and helped me write my dissertation in English. Jacob Mey contributed invaluable expertise in linguistics. Mark Schaefer produced the videos, and

Jarrett Knyal created the graphics. William Fitzgerald and Max Silberztein helped us with data base and graphics tools respectively. Maureen Zabloudil, Matt Greising, and Laura Reichert acted in our videos; Mike Engber, Lucian Hughes, and Richard Lynch contributed code.

This project gained direction and purpose from addressing an authentic need, and Andersen Consulting was instrumental in helping us identify and work on a real-life problem. In addition, in their training center in St. Charles and here at the Institute, Andersen Consulting employees contributed essential expertise and helped us collect information, obtain necessary resources, and produce the simulation videos. Employees from Japan and Spain assisted us in testing and redesigning early versions of Dustin. The cooperation of these professionals and Andersen Consulting's institutional support were fundamental to this work.

A number of people at ILS enriched my academic life. I was fortunate to share an office with Lucian Hughes, who taught me much about science and AI. I had helpful meaning-of-life conversations with Lucian, Liz Gearen, Laura Reichert, and Alex Kass. Diane Schwartz helped me chill out. My classmates, Kemi Jona, Danny Edelson, Louise Pryor, Matt Brand, Dick Osgood, Michael Freed, and their spouses, Sue Jona and Vivian Edelson, provided company and support.

Angel creates and nurtures an environment that enriches our lives and supports our growth. For being so lucky as to have such a beautiful spring of love and fun, I am very thankful to life, and to you Angel, to whom I dedicate this work.

Table of Contents

Chapter 1 - Introduction 1

A problem: Talking to Americans 1

Different worlds: Classroom & Reality 2

Technology can help 5

A solution: Dustin 6

Implementing Dustin 11

What's next? 16

Chapter 2 - Language Learning 17

Introduction 17

Language Teaching Methods 18

Grammar-Translation 19

Direct Method 20

Audiolingual Method 22

A need for change 25

Natural Approach 27

The State of Language Teaching 30

Naturalistic Learning 31

The Problems 32

Conclusion 34

Chapter 3 - Technology in Language Learning 37

Introduction 37

Audiovisual 38

French in Action 39

Computers 40

Supplementing classroom activities 41

A microcomputer game in French culture and civilization. 42

The Dark Castle - An adventure in French 45

Multimedia Exploratory Environments 46

The Zarabanda Notebook 47

Multimedia Simulations 48

Athena Language Learning Project (ALLP) 49
Direccion Paris: A la rencontre de Philippe 50
No Recuerdo 52
Conclusion 55

Chapter 4 - Design Issues 57
Introduction 57
What's so good about living in Japan? 58
The Situations 61
Context (Social/Physical) 62
Perception (Audiovisual) 63
Behavior 64
The Student 65
Role 65
Goals 66
Interaction: Student & Situations 67
Sequence of events 67
Exploration: Let the student choose 69
Threat 69
Resources 71
Tutor 72
Problems: Limitations of technology 74
Conceptual Feedback 74
Natural Language 76
The silent period 77
The Classroom Proficient 77
Conclusion 78

Chapter 5 - Dustin - A Language Learning Environment 79
From Principles to Practice 79
The Components 79
The Real-Life Situation 80
Dustin 80
Principle 2 83
Principle 1 84

Principle 4 86
Principle 15 88
Principle 3 91
Principle 11 92
Principle 14 93
Principle 16 94
Principle 9 97
Principle 10 97
Principle 5 98
Principle 6 101
Principle 7 105
Principle 13 107
Principle 12 108
Principle 8 108
Conclusion 113

Chapter 6 - The RPSS Architecture 115
What exactly is Dustin? 115
A method of teaching 117
Dustin's Components 118
Tutor 118
Examples 119
Simulation 119
Tools 120
Dustin's Model of Instruction 120
What's good about Dustin? 123
Who's using it? 125
What do they report? 125
Situated learning & Transfer 126
Exploring learning strategies 127
Exploring the social context 127
I want to belong 128
RPSS and other CALL systems 128
What's RPSS good for? 130

Teaching other languages 130
Other Apprenticeship 130
Communication Skills 131
What next? Ookie? No, Cookie. 132
Conclusion 133

Chapter 7 - MOPed - An Authoring Tool 135
A problem... a BIG problem 135
A Solution (MOPed) 137
A Uniform Representation 137
A Visual Artifact 138
MOP - Memory Organization Packet 140
MOPed - A Demo 142
HUH? 147
Input Patterns 148
Keywords 149
Tutor Interventions 149
Handling Default 149
Help Messages 150
Button Handler 150
Where does MOPed fit in? 151
What's MOPed? 153
What's MOPed good for? 155
Conclusion 157

Chapter 8 - Conclusion 159
What's important? 159
Language Learning 159
Realistic Simulations 160
Templates & Tools 161
MOPed & AI 162
Powerful Systems 163
Extending Intelligence 165

References 167

List of Figures

1. Dustin introduces the simulated environment to the student 7
2. The tutor instructs the student 8
3. The agent talks to the student 8
4. The tutor intervenes 9
5. Selecting a location in the simulation 10
6. A knowledge representation structure in an old version of Dustin 11
7. A MOP: Going through immigrations at O'Hare International Airport 12
8. Post-its connect 13
9. Responding to a series of HUH?s 13
10. A MOP: Ways of saying "I need..." 14
11. Using a MOP inside another 14
12. Simulations can be divided into three major components 79
13. Tutor buttons in Dustin 93
14. Dustin Tools 97
15. Dialog box 97
16. Dustin lets the student select a task 110
17. Components of the RPSS Architecture and their functions 118
18. Elements involved in language acquisition 133
19. Reception scenario described in incomprehensible language 136
20. Rule-base for dialogs (Absurd!) 137
21. St. Charles - A big sheet of paper with post-its 139
22. Going through immigrations at O'Hare International Airport 142
23. The tutor box appears on the screen 143
24. Failing at first try, the tutor takes over 143
25. Watching Maria go through immigrations 144
26. Modified MOP - Skips the dialog and starts with a video 144
27. The upper portion of the "(DI) O'Hare/Customs" MOP 145
28. The "How long will you be in the US?" subdialog 146
29. Patterns in the HUH? MOP 147
30. Responding to a series of HUH?s 147
31. Time period MOP 148

32. The speech patterns that are acceptable 148
33. Keywords 149
34. Encoding tutor interventions 149
35. The default handling MOP 150
36. Help messages in a MOP 150
37. Button Handler 150
38. The higher level MOP - Sunday in St. Charles 151
39. Authoring tools for RPSS systems 152
40. Palette 154
41. Creating a MOP 154
42. A MOP that is reused in four other MOPs 155

List of Tables

1. Methods of Language Instruction 35
2. Types of technology-based language learning systems 55
3. Features of Dustin and other CALL systems 56
4. Variables that influence lang. acq. in natural settings 60
5. Variables that influence language learning 61
6. Desirable characteristics of a CALL environments 78
7. Desirable characteristics of CALL environments 116
8. Comparing the real-life Japan-experience and the St-Charles-experience 124
9. Features of the RPSS architecture 124
10. Features of Dustin and other CALL systems 129

Chapter 1

Introduction

A problem: Talking to Americans

Twelve hours after I packed up and left Sao Paulo, Brazil, to try to make it in New York, I was in Manhattan looking for the YMCA. "Where is the YMCA?" I asked a man on 42nd street, where I had gotten off the bus. "Mumble, mumble, mumble looking for?" he said, speaking loud and firing words at me. "Y-M-C-A," I said tentatively. "Mumble, mumble, mumble," he replied, and pointed south. I knew I could find the Y, so I wasn't worried about getting there. But after having studied English in school for seven years, what that man said to me made absolutely no sense, and that scared me.

Later that day, I had to concede that what I had learned in seven years of English classes in Brazil just didn't have much to do with what they used on the streets of New York. I asked the clerk at the YMCA on 34th street to repeat and simplify every sentence three times before I could understand him. Still, despite the time I had spent studying the intricacies of the English language, I could not get him to store my valuables in a safety deposit box. Maybe they didn't have them, but if that's what the clerk was trying to say, I didn't understand that either. At 19, the reality of life in the United States shocked me. Alone in my room that afternoon, I fought the suspicion that I just wasn't prepared to stay and work in the US as I had intended.

But things did get better. I found my way to a country club in Monticello, NY, where I got a job as a busboy. There, I also found the ultimate English course: serving coffee and asking people "Is there anything else you need?" while hoping that their answers would be comprehensible. I needed the job, and keeping it depended on my understanding the guests and their understanding me. It took a number of trips to the kitchen before I learned to distinguish "rye bread" from "white bread," since they both sounded exactly the same. However, although "Worcestershire sauce" still sounded like a gray-haired guest was trying to make fun of me, two months later I had become very good at distinguishing names of food.

I remember taking classes, listening to tapes, and practicing drills. Unfortunately, though, most of what I remember about studying English is just that, a painful process of listening and repeating boring material. Some of the skills I had acquired through classes did help me survive in the US, but they required much adaptation before they were useful. In fact, because I had learned to speak from non-native speakers, I had to unlearn many speech patterns. As if it were not enough that I had to fight the interference in speech from my own language, I also had to correct bad speech habits acquired through seven years listening to non-native speakers of English.

Today, sixteen years after my New York experience, when I think of learning a foreign language, I dread the idea of picking up another grammar book or taking another class. I am interested in learning Japanese, for example, but the idea of studying Japanese in any way other than by living in Japan horrifies me. If I am going to have a hard time once I get there anyway, why should I suffer in advance? Obviously, I should just go to Japan.

Different worlds: Classroom & Reality

What is the problem with language instruction? I witnessed other foreigners having the same problems, so it wasn't just me. What we learned in the classroom was not what we needed to survive on the streets of New York. In New York, in the midst of heavy traffic, we need to understand a person with an eccentric accent speaking fast and impatiently using points of reference that are only meaningful in Manhattan. Instead, classroom instruction prepares us for a world that doesn't exist. We acquire skills that are often useless -- more like learning auto mechanics when we need to know how to drive. When the man on the street responded to my asking "Where is the YMCA?" by saying something like "The 'Y'? On 34th or 47th?... which one'ya lookin' for?" I could have criticized the structure of his response, but I couldn't even make out the words coming out of his mouth. It became clear to me that, while I knew the structure of the English language, I just didn't know how to use it.

If we want to understand directions from a New Yorker or follow a business discussion in Texas, why do we learners spend so much time studying the structure of the language instead of how it is used in real life? How did language instruction get to be so out of touch with reality? What is the problem with language instruction?

One problem is that the goal of language instruction has not always been to prepare students to interact with foreigners. At the turn of the century, for example, learning languages was primarily an intellectual exercise for the well-educated (Richards & Rodgers, 1986), having nothing to do with communicating with others. Learners used to study the structure of languages, memorizing long lists of words and their translations, and focusing on rules of grammar. Given this view of language, it is not surprising that the skills students developed were useless in real-life interactions.

Over the last century, however, this much has changed. Methods have evolved to accommodate changing needs, and now language instruction is primarily concerned with developing communicative competence, the ability to interact with others in a foreign language. Current methods emphasize interactions, both as a goal and as a means to learning, and try to give students experiences that prepare them to perform in real life. Naturalistic methods, as methods that replicate real-life experiences are referred to, try to expose students to the target culture and engage them in realistic interactions.

Nevertheless, even though the goal of language instruction and the methods of instruction have changed -- now we want people to learn how to interact, and people learn by interacting with others -- classroom instruction still falls short of preparing students to perform in real life. The reason is that, despite the concern with real-life performance, better theories, and the more pragmatic approach, classroom instruction still faces a problem that has always afflicted language instruction: lack of resources.

Language acquisition depends heavily on interactions with native speakers, and native speakers are hard to come by. Regardless of the method of instruction or how well-intentioned teachers are, without native speakers to interact with, instruction often falls back on practices that are useless in real life. In my case, for example, during seven years of English lessons I never spoke with an American -- I had never heard anybody say "which one are you looking for," in less than one second. Worse yet, because I didn't have access to native speakers of English, I ended up practicing with learners whose bad speech habits only reinforced my own bad habits.

In addition, interactions always occur in a larger cultural context, and students need access to this context. During interactions, the physical and social context provide important cues that are integral parts of the process of communicating. When the man in Manhattan asked me, "On 34th or 47th?" he was referring to street names in New York, a feature of the target environment that I knew nothing about. When students are getting ready to come to New York, they must familiarize themselves with the way New Yorkers give directions. The problem with language instruction is that, even when native speakers are available, it is hard to bring the target environment into the classroom.

Individual instruction and feedback is another scarce commodity. To be motivated to learn, students have to work on problems that are relevant to them. Instead, classroom instruction forces every student to participate in uninteresting group dialogs, and since they must wait their turn to interact with the teacher, students have limited individualized feedback. When instruction doesn't address the specific needs of a student, he usually looks out the window and mechanically repeats whatever he has to say. The interesting thing about interacting one-on-one is that it forces us to exchange information, express feelings, and perform social transactions. Unless students address personal needs by engaging in interactions that are meaningful to them, the information conveyed is irrelevant. Unfortunately, in the typical classroom, teachers cannot give each student the required degree of individual attention.

Due to the lack of adequate resources, classroom instruction still provides (1) limited interactions with native speakers, (2) limited access to the target culture, and (3) limited individual attention and feedback. Despite the improved methods of language instruction, students still don't get the chance to develop the skills necessary to interact in a foreign language. Proficiency tests that emphasize language structure further aggravate this problem by discouraging teachers from searching for adequate solutions and encouraging students to focus on rules and memorization that have very little to do with interacting in real life.

Technology can help

Ideally, students should learn by living in the target environment, interacting with native speakers, being immersed in the target culture, and receiving intense individual instruction and feedback. Since this is not always possible, however, the alternative solution is to bring those experiences to the student. This is what early language laboratories tried to do. They tried to bring native speakers and the target culture into the classroom by using instructional material such as tape recorders, audiovisual, video, and, more recently, computers.

Traditionally, however, laboratory material has been non-interactive. This has been a problem because in order to learn to interact, interactivity is essential. Watching others passively, like going to see French movies three times a week, doesn't really teach much. The unfortunate consequence of this limitation is that instead of providing the much-needed experiences, technology has been perpetuating ineffective practices. Tape recorders have been drilling students in patterns out of context, videos have been keeping students in passive roles, and multiple-choice computer programs have been testing students on memorized grammar rules. Technology can be used more effectively. The main reason language instruction fails is that students have limited exposure the target culture.

This is where multimedia and artificial intelligence come in. Short of importing a lot of native speakers, computer-based simulations are the closest we can get to living in the target culture. Given the plummeting cost of hardware, computer-based multimedia simulations of real life can be the ideal playground for language learning. Simulations can give students much needed interactions with native speakers, access to the target culture, and individualized attention. It is even conceivable that in certain situations simulations could be better than real life experiences.

To be used properly, however, technology must simulate those elements of the environment that promote language acquisition. Simply simulating conversations, for example, is not enough: that would be like throwing an apprentice mechanic into an auto shop with cars to fix and no tools, instruction, or guidance. Instead, we must consider the

student's needs, the problems he will confront in real life, the type of guidance and information he requires, and the tools that he needs to accomplish his task.
In addition, and very importantly, to design language learning environments we need to consider how people acquire languages, the conditions that are necessary for learning, what motivates people to learn, and the resources and experiences that they need to develop communications skills. This requires the combination of content expertise (second language acquisition research in this case), theories of learning (educational research and cognitive science), and issues of motivation and tutoring systems design (psychology and intelligent tutoring systems).

When we consider these issues of language acquisition, education, learning, and motivation, it becomes clear that certain design principles should guide the development of computer-based language learning environments. For example, knowledge is always a product of the activity, context, and culture in which it is used (Brown, Collins, Duguid, 1989), and consequently, language should always be presented in the context in which it is used in real life. The environment provides the conditions for applying and organizing knowledge. Also, since students learn best when they are using language as a tool to achieve their goals, the goal of their interactions must always be meaningful and clearly defined. Moreover, students need to have control over the environment so that they can address their own needs and explore their own learning strategies. In terms of feedback, second language research shows that correcting grammar mistakes doesn't help much. Concerning interface design, experience with intelligent tutoring systems raises some problems. For instance, since computers don't understand facial expressions, a common way of expressing communication breakdowns, simulations must provide users with alternative means of expressing the same information. I discuss these design principles in Chapter 4.

A solution: Dustin

I introduce a multimedia language learning environment, Dustin, that addresses the problems afflicting classroom instruction and incorporates the principles mentioned above. Dustin allows students to interact with and observe native speakers in the target culture while providing them with tools and individual instruction and feedback.

Dustin helps foreign employees of Andersen Consulting who are coming to the United States for the first time. These employees undergo three weeks of intensive training in St. Charles, Illinois, with little time to practice or study English, and they face the same kinds of problems that I faced during my first visit to the US. Dustin helps these first-timers minimize their problems understanding and interacting with Americans by exposing them to the St. Charles environment before-hand, so that they brush up their language skills before coming to the United States.

To bring the student into the simulated environment, Dustin starts by introducing him to the experiences he will face in the simulation, which includes typical events during a trainee's first twenty-four hours in the US. The introductory video shows a sample of the tasks the student will face in O'Hare and St. Charles, establishing the context of the experiences that lie before him (see Figure 1).

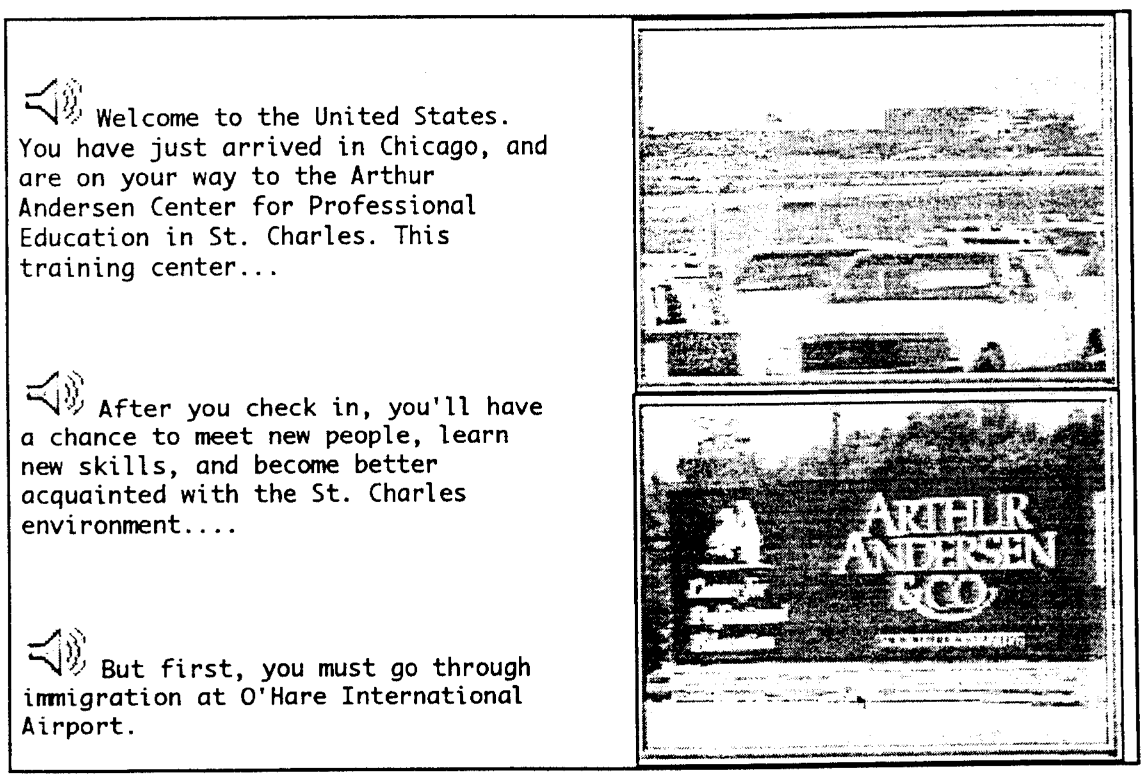


Figure 1. Dustin introduces the simulated environment to the student.

At this point (see Figure 2) the tutor assigns him his first task: "Go through Immigration at O'Hare International Airport." The video on the small window beside the message shows that he is approaching the booth where the Immigration officer is waiting.

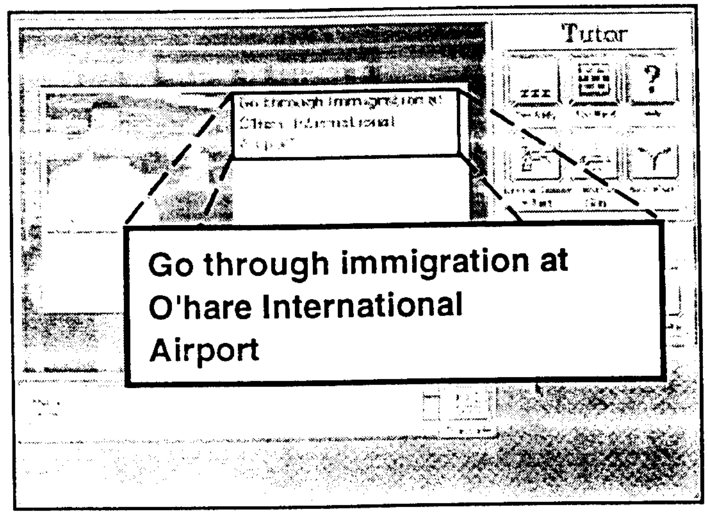


Figure 2. The tutor instructs the student.

If the student agrees with the task, he finds himself interacting with the Immigration agent who says: "May I have your passport please?"

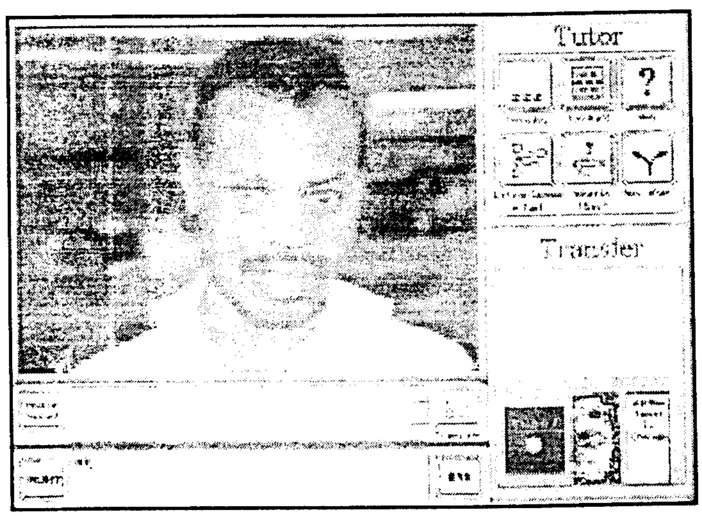


Figure 3. The agent talks to the student.

The student either types an answer, manipulates objects such as passport or money, or pushes buttons in response to the simulated agent's utterances. Figure 3 shows what Dustin looks like in this situation. Chapter 5 describes an extensive session with Dustin.

The tutor assigns tasks, gives individualized instruction to the student, and helps him interact with native speakers. Whenever the student is unable to perform a task, or gets stuck somewhere, the tutor either gives some instruction or shows an example of someone else performing the same task, letting the student model the correct behavior. Figure 4 shows the tutor intervening after the student refused to turn in the passport a number of times.

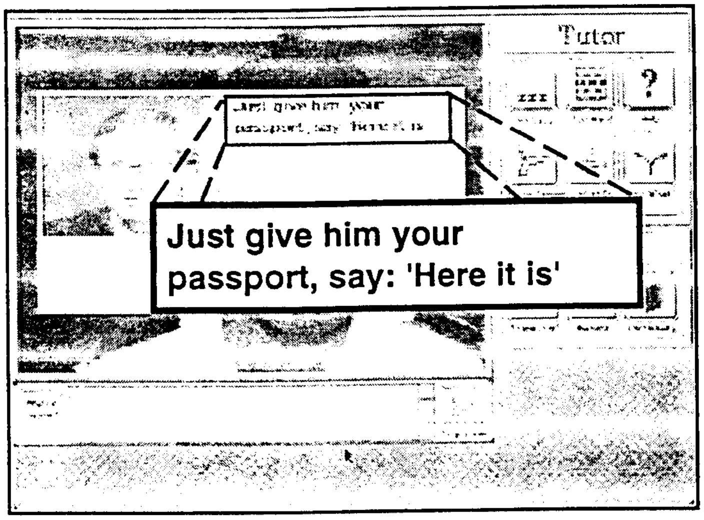


Figure 4. The tutor intervenes.

After going through Immigration, the student has to find transportation to St. Charles, and once there, he must check in at the hotel, find his room, and meet his roommate. At the end of the day, he gets a snack at the local coffee shop. The next morning, the student goes to the cafeteria, where he will have most of his meals, and gets some breakfast. Immediately after breakfast, his classes start. In the classroom, he interacts with the instructor and helps another student who comes late for class. Overall, Dustin includes sixteen tasks in nine different scenarios -- four additional tasks in the roommate scenes accommodate either male or female students. All tasks involve achieving goals that require interacting with native speakers who work in St. Charles.

Dustin gives the student a guided tour of St. Charles, throwing him in interactions and showing him examples. Under normal conditions, the tutor sequences the tasks for the student, following a storyline that reflects the experiences of a typical trainee. However, the student can also take control over the environment, choosing what to do and what to watch. If a task is uninteresting to the student, or on the other hand, if it is so complex that the student wants to revisit it, he can do so by simply choosing to select a task himself. Figure 5 shows a map with the different locations in the simulation. The student chooses a place to go to, and then the tutor asks him what task he would like to perform (e.g., check in or ask for directions at the reception). In addition, Dustin provides a number of tools to assist the student during his tasks (e.g., dictionary, translations, transcript, recorder).

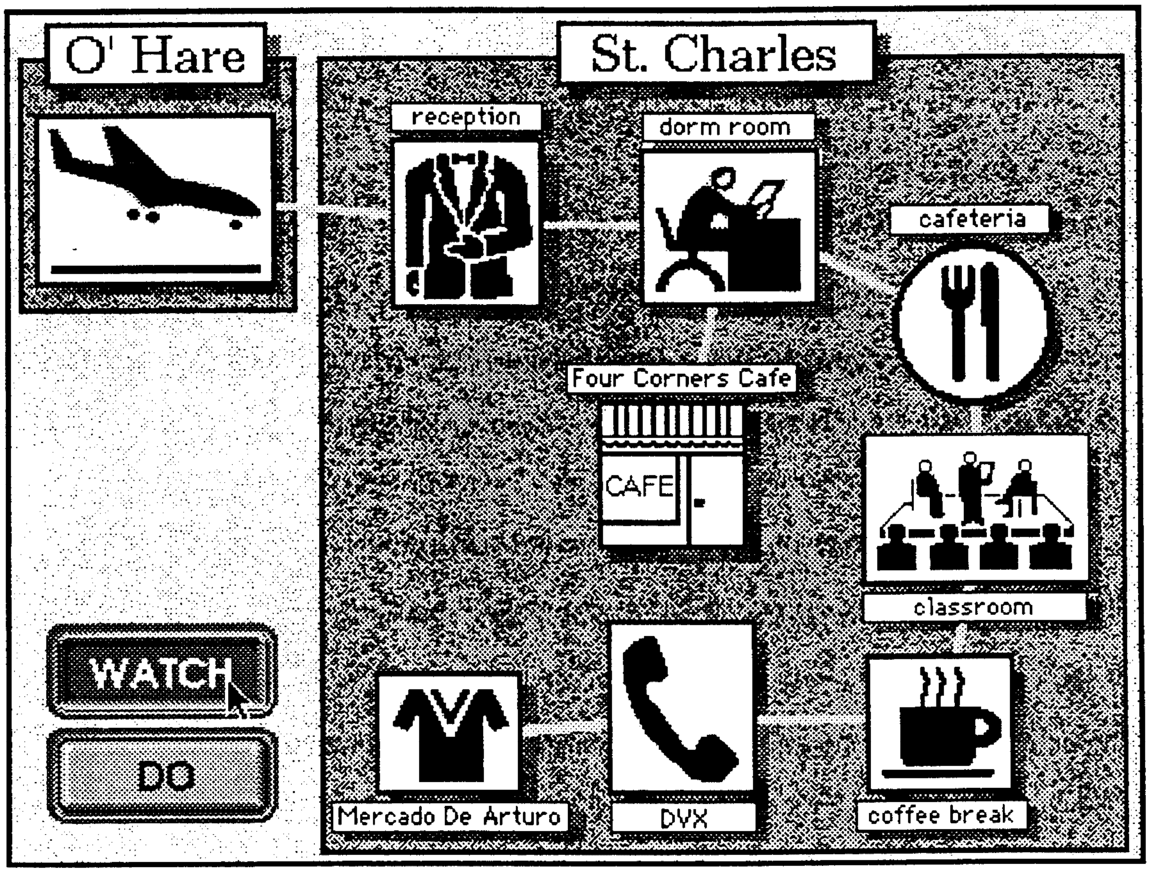


Figure 5. Selecting a location in the simulation.

More than just implementing simulations of dialogs, Dustin implements an environment in which students learn from experience. Instead of transmitting information to the student, Dustin engages the student in realistic tasks and gives him the support (i.e., tutor, tools, help, examples) necessary for him to learn experientially. As I suggest later, Dustin's architecture can be used to teach other well-scripted skills (e.g., teach other languages, train bank tellers, receptionists, and even auto mechanics).

Implementing Dustin

Implementing systems that respond to a large number of events, organize a large number of scenarios in different ways, provide contextualized help, simulate dialogs and provide remediation when needed, requires complex knowledge representation structures. In addition, the complexity of these knowledge networks increases very rapidly with size, and maintaining such knowledge networks can be a nightmare.

In working with Dustin, we found it impossible to maintain its complex networks of information without adequate tools to help organize and visualize them. Networks consisting of knowledge structures such as the one in Figure 6, from an early implementation of Dustin, become very hard to maintain. Their intricate interdependencies and connections become very hard to visualize even when a small number of structures are involved. Solving this problem required the development of an authoring tool to implement Dustin-like, interactive social simulations.

```
(in-package "USER")

(SOF SCE-REC-CHECK-IN
 :NAME SCE-REC-CHECK-IN
 :DESC "Check in"
 :REQUIRED-P T
 :WATCH-TEXT "Watch John Harrison check in"
 :WATCH-SCRIPT SCR-REC-CHECK-IN
 :NEXT-TASK (TASK OPT SCE-REC-CHECK-IN)
 :DO-TEXT "Go to the reception desk and check in"
 :DO-AGENT A-KELLEY
 :SUCC-TEXT "Good! Now, go to your room and meet Scott."
 :SUCC-TASK (TASK DO SCE-ROO-MEET)
 :FAIL-TEXT "Let's watch something simpler"
 :FAIL-TASK (TASK ?COMPUTE ?NEXT)
 :OPT-TEXT "Can you check in at the reception desk?"
 :OPT-YES-TASK (TASK DO SCE-REC-CHECK-IN)
 :OPT-NO-TASK (TASK ?COMPUTE ?NEXT)
 :GOAL-TEXT "Try to check in"
 :GOAL-OK-TEXT "You've already checked in. Leave"
 :GOAL-TEST (DONE-P D-REC-CHECKIN)
 :GOAL-SAY ""
 :EXAMPLES NIL
 :PLAN (SCE-REC-GREET SCE-REC-PURPOSE SCE-REC-NAME SCE-REC-PACKAGE SCE-REC-THANK)
 :DIALOGS "d-rec-checkin")
```

Figure 6. A knowledge representation structure in an old version of Dustin.

The solution I introduce here, MOPed, based on an artificial intelligence model of memory organization proposed by Schank (1982), enabled us to organize, reuse, contextualize, visualize, and understand the information in Dustin. Being more than a tool to organize and maintain knowledge, MOPed helps developers understand complex information by providing visual aids to improve readability and mechanisms to contextualize data.

The idea underlying MOPed's memory organization scheme is very simple. MOPed is based on the idea of having a memory structure whose only function is to organize other memory structures (Schank, 1982), like having a sheet of paper on which to organize post-its. The basic unit in this scheme, a Memory Organization Packet, or MOP, serves as a sheet of paper, and the simpler memory structures serve as post-its. Figure 7 shows a MOP that organizes the events in the Immigration scene at O'Hare International Airport.

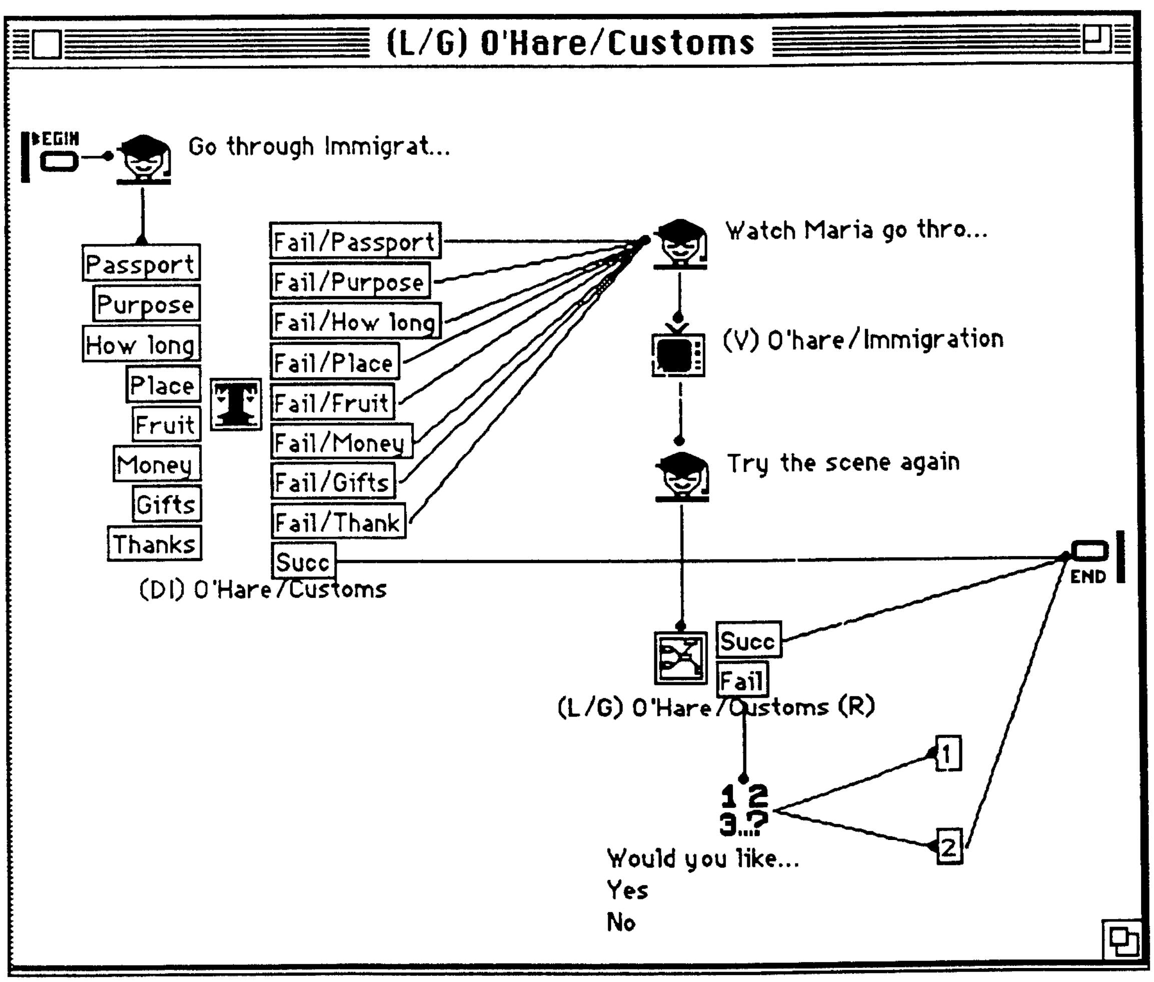


Figure 7. A MOP: Going through immigration at O'Hare International Airport

The simpler memory structures, or post-its, can be things like a tutor message, 

, or a video-clip, 

. When connected to each other, as in Figure 8, they indicate that Dustin should display a tutor message saying, "Watch Maria go through Immigration," and then show a video-clip in which Maria interacts with the Immigration officer.

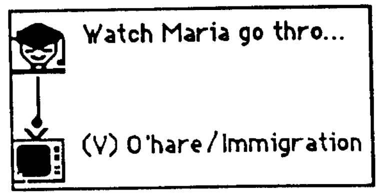


Figure 8. Post-its connect.

The interesting thing about MOPed is that the same knowledge representation and visualization scheme organizes everything in Dustin. It organizes Dustin's storyline with its sixteen tasks; in each task it organizes tutor messages, video-clips, and dialogs; in each dialog it organizes the simulated interactions, speech patterns, help messages, and button handlers. Figure 9 shows a small portion of a simulated dialog.

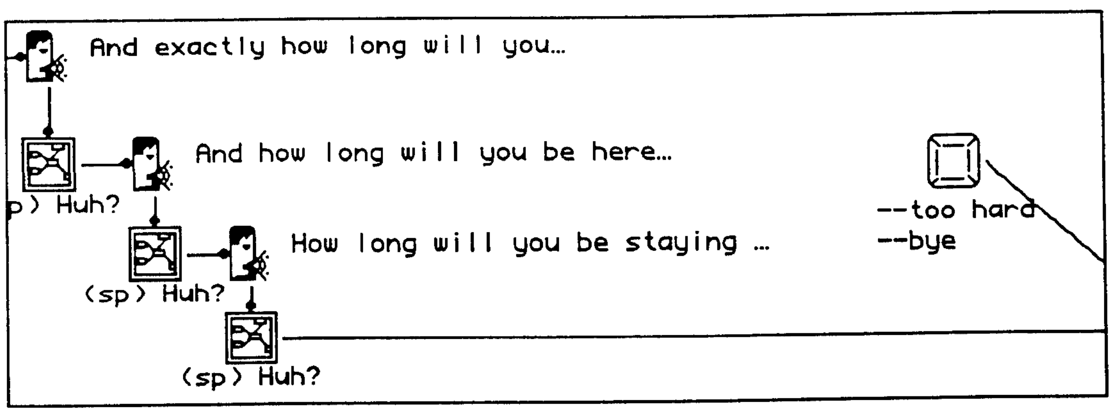


Figure 9. Responding to a series of HUH?s.

In addition to helping visualize information, MOPed allows us to reuse existing MOPs. For example, a MOP like the one in Figure 10, which occurs in many dialogs, can be reused instead of recreated. When the MOP "(sp) I need to," from Figure 10, is used inside another MOP (see Figure 11) then it appears as a post-it. The MOP itself is not copied into the MOP in which it appears, instead the post-it only points to it, avoiding duplication of existing structures. The MOP in Figure 11 parses sentences beginning with, for example, "I am trying to" + "find..."

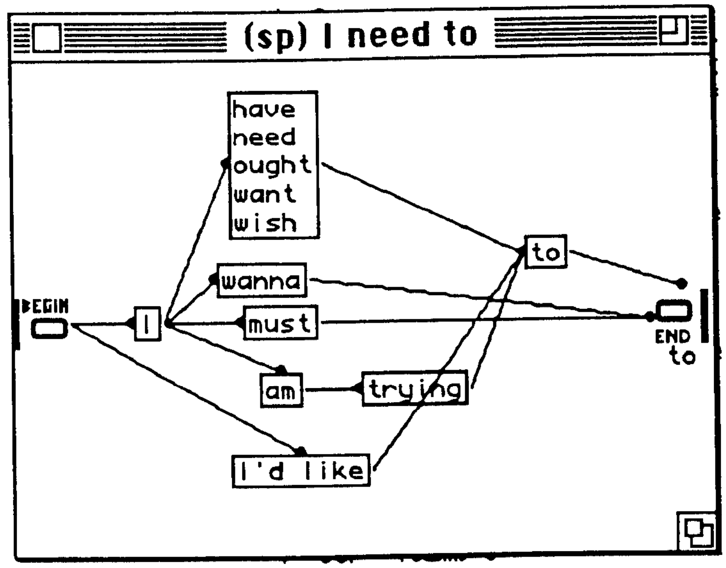


Figure 10. A MOP: Ways of saying "I need...".

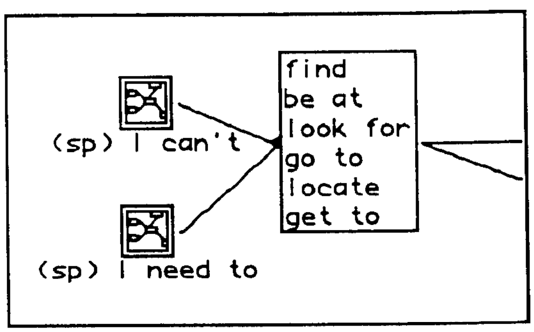


Figure 11. Using a MOP inside another.

MOPed also simplifies the development of templates for lessons. Determining the sequence of events in each lesson (i.e., tutor messages, simulations, examples, remediation, interventions) involves a process of approximation that usually involves numerous changes. In earlier versions of Dustin, without MOPed, creating these lesson templates involved code modification and recompilation. With MOPed, simple object manipulation (e.g., dragging, deleting, connecting post-its) does the job. In addition, once a template is created, it can be copied through simple Copy/Paste operations. In fact, most of the lessons in Dustin were copied from a few original templates.

Another benefit of using this memory organization scheme is that knowledge is contextualized by the MOPs in which it appears. For example, when the student is checking in at the hotel, only the Checking-In MOP and the MOPs inside it are active. Consequently, during the dialog with the receptionist, the parsing mechanism checks the input only against those patterns predicted in that context. In other words, in Dustin, the search space is significantly reduced by the context in which a event occurs.

MOPed is a tool to organize, visualize, contextualize, and reuse knowledge structures. One way of thinking about MOPed, from a developer's perspective, is as a tool with which to organize messages exchanged by objects in an object-oriented environment. In most object-oriented environments, few tools help developers organize and understand the interactions between the messages that objects exchange. This is what MOPed does. Chapter 7 describes MOPed in detail.

What's next?

I start by looking back at what has been done, both in language teaching and in computer-assisted language learning. Then I discuss design issues, introduce Dustin, discuss its architecture, and introduce MOPed, the authoring tool to build Dustin-like systems.

The next chapters discuss:

- Language learning in the classroom (Chapter 2 - Language Learning)

- How technology has been used in language learning (Chapter 3 - Technology in Language Learning)

- What I think should be done (Chapter 4 - Design Issues)

- An example (Chapter 5 - Dustin)

- Why Dustin is a good solution, why it is not. (Chapter 6 - The SBLL Architecture)

- A tool to build Dustin-like systems (Chapter 7 - MOPed -- An authoring tool)

Chapter 2
Language Learning

Introduction

In the previous chapter, I described the nature of the problem: How do we train people to communicate in a foreign language? Examining the history of language instruction gives us some insight. To begin with, as we see how methods evolved to meet changing needs, the importance of defining the goal of instruction becomes clear. As I mentioned earlier, the purpose of language instruction has not always been to train people to perform in real life. At one time, language learning was considered useful only as an intellectual exercise for the well-educated. Only recently have applied linguists begun to see the role of language instruction as that of preparing people to interact with others. The way we view language defines the goal of language instruction, and defining it precisely is crucial. Instruction may fail to prepare students to perform in real life not because of the method itself, but because of the goal underlying instruction.

As I analyze the evolution of language teaching methods, three points become clear: (1) what constitutes proficiency today, in other words, the goal of language instruction, (2) the principles underlying those methods that achieve this goal; and (3) the problems involved in implementing these methods. These main findings can be succinctly stated as follows.

(1) The goal of language instruction.

Teach students how to interact and perform social transactions using language.

The goal of language instruction is not to teach students the structure of languages. We must focus on communication, not on form.

(2) The principle underlying learning.

We learn language by using it as a tool to perform social transactions.

People learn best by using language in the process of interacting with others -- they learn by doing. Explicit instruction about language structure does not help.

(3) The main problems with language instruction.

(a) Limited interactions with native speakers.
(b) Limited exposure to the target culture.
(c) Limited individual attention and feedback..

Since the resources necessary to provide all these experiences are rarely available, teachers have to turn to textbooks and methods that simplify their lives but that do not help students learn to communicate.

I begin by looking at a number of methods of instruction: the needs they address, their methodology, and their problems. Later, I draw a picture of the current state of language instruction, highlighting trends and identifying primary problems. Understanding these gives us a good start towards finding solutions.

Language Teaching Methods

The following overview of methods of instruction begins at the end of the 19th Century with the Grammar-translation method, and ends at the end of the 20th Century with the Natural Approach. Over the last century, there has been a significant increase in cross-cultural interactions in the world, and language teaching has become concerned with communication rather than with intellectual exercise. During this period, methods have evolved from a structural, non-communicative approach to a very different interactionist, communicative approach. Each method addressed specific needs in particular contexts and contributed information that helped define the goal of language instruction today and the main problems in achieving that goal.

Grammar-Translation

In the 19th Century, given the limited opportunities for contact with foreigners, people studied foreign languages to either (1) read foreign literature -- mainly Latin, a language no longer spoken -- or (2) develop intellectually through the mental discipline involved in studying a foreign language (Richards & Rodgers, 1986). During that time, Grammar-translation combined the influence of German scholarship with methods used to teach Latin. In this method, language was viewed as a set of symbols and rules, and language learning was synonymous with memorization of these symbols and rules. Grammar-translation emphasized the study of grammatical rules and translation exercises, and typically students acquired extensive knowledge about the language, but no competence in using it interactively (Finocchiaro & Brumfit, 1983).

In Grammar-translation, the student (1) listens to explanations of grammar rules, (2) memorizes bilingual lists of words that contain the vocabulary needed for the day's exercises, and then (3) practices the grammar rules and the vocabulary on a given reading selection. These exercises consist of translating sentences to and from L1 (source language) and L2 (target language), with L1 being the medium of instruction.

Implicitly, Grammar-translation assumes that mastering languages depends on conscious knowledge of grammar. Heavily influenced by the idea that language learning is a good mental exercise, teachers present rules of grammar abstractly and then force students to practice their application in selected readings; students learn grammar deductively. The linguistic knowledge imparted includes types of sentences (i.e., declarative, interrogative), vocabulary organized by parts of speech (i.e., verbs, nouns, etc.), and endings (i.e., genitive, accusative, nominative). Accuracy is important, and translation exercises measure student performance.

Vestiges of Grammar-translation still inhabit contemporary college textbooks in the form of long word memorization lists, emphasis on grammar rules, and translation exercises. One of the main reasons these features have survived is that, although painful for students, they are convenient for teachers. Grading is easy. Though still practiced, however, Grammar-translation has no advocates; no one in the field defends either its utility or its theoretical foundations (Richards & Rodgers, 1986). The method addresses literary needs of literary people. Sessions are usually boring, and the pressure for accuracy often causes anxiety. No attempt is made to teach students to interact, and what they learn from Grammar-translation is of little use on the streets of a foreign city. Grammar-translation is a prime example of "learning about a language" rather than "how to use a language."

Direct Method

As written communication gave way to oral communication, due to increased opportunities for interaction among Europeans, the Direct Method emerged. The Direct Method evolved, in the 1920s, almost as a reaction to the Grammar-translation method. Instead of focusing on written language, it focused on spoken language; instead of translating everything, it presented everything in the target language. Instead of teaching grammar explicitly, it expected students to learn grammar inductively. However, although successful with certain populations, most notably through the Berlitz school, the Direct Method proved inadequate for use in schools.

In the Direct Method, teachers defer written language for months and even years, and teach oral skills first. They focus on pronunciation and grammatical accuracy, and are particularly strict about using exclusively the target language. As described in Richards & Rodgers, (1986), according to Gouin and colleagues, precursors of the method, people should learn L2 (second or other language) in the same way that children learn L1 (first language). The most important characteristic, they propose, is that "a foreign language [can] be taught without translation or the use of the learner's native tongue if meaning [is] conveyed directly through demonstration and actions." It was Franke, a psychologist, who provided the theoretical justification for a monolingual approach. For Franke, students learned best by using language actively, because there is a direct association between forms and meaning in the target language (Richards & Rodgers, 1986). In the Direct Method, everything happens in the target language -- translating is taboo. Teachers introduce every new word either through pictures, pointing, and mime or through associations of ideas.

Sentences are sequenced according to their grammatical structure, and teachers ask questions about the structure of the language to help students extract grammatical knowledge. The Direct Method emphasizes (1) aural-oral skills, (2) the exclusive use of the target language, (3) teaching of grammar inductively, and (4) accuracy -- errors are corrected in the classroom.

The method met with success with highly motivated, paying students who found language learning intrinsically motivating. However, although successful with this population, through private institutions such as Berlitz that could hire good teachers who were native speakers, the method proved problematic in secondary schools. It depended heavily on native speakers and on the teacher's skills, rather than on more readily available resources such as textbooks. In practice, it also proved counterproductive in its dogmatic rule that forbade the use of L1, leading to many wasted hours of mime and actions trying to explain a single new word. In overemphasizing the importance of using only the target language, Sauveur and other proponents overlooked important factors, such as extent of exposure, the artifacts in the environment, and the quality of the experience that were not captured by the simple absence of the native language (Richards & Rodgers, 1986). Also, the requirement that sentences be sequenced according to their grammatical structure imposed an unrealistic sequencing of input, making it hard for teachers to contextualize the use of language. After studying artificial sentences like "La plume de ma tante est sur le bureau de mon oncle," (Finocchiaro & Brumfit, 1983, p. 5) students had difficulty applying them to real-life situations. As Finochiaro describes it, "all statements used were related to the classroom. Teachers did not generally think of students using language beyond the classroom. Any connection with real life was expected to come later and was not the business of the school." Its theoretical foundations were weak, and the Direct Method was soon perceived as the product of amateurism (Richards & Rodgers, 1986).

Although based on good intuitions about naturalistic learning, Direct Method's dependence on intensive resources was incompatible with the means of most schools. Certainly, its extremist stance on sequencing material according to grammatical structures and its dogmatic position on forbidding translations contributed to its demise. However, its major problem was the lack of resources. The skilled teachers required by the method were costly and beyond the means of the typical school classroom. To work with large numbers of students, the Direct Method demanded a prohibitive amount of resources.

These practical issues, documented in the Coleman Report (Coleman, 1929), triggered Direct Method's decline. After studying the state of foreign language instruction in the U.S., Coleman published a report that attacked the importance given to spoken language. He argued that teaching conversational skills was impractical given the restricted time for foreign language instruction in schools, and considering the limited skills of teachers, instruction should focus on reading and grammar involved in simple readings. Conversation skills for secondary students in the U.S. was perceived as irrelevant, and between the two World Wars, language instruction reverted once again to focus on reading, writing, and grammar (Finocchiaro & Brumfit, 1983; Richards & Rodgers, 1986).

Audiolingual Method

When the United States entered World War II, the need for oral proficiency increased dramatically (Richards & Rodgers, 1986). For two years, the Army trained soldiers in German, French, Italian, Japanese, and other languages, to prepare interpreters and code-room assistants. Fifty universities provided eight-hour-a-day intensive courses. The intensity of contact during these courses proved to be effective and convinced applied linguists of the value of both intensity and the oral-based approach. When the war was over, the U.S. emerged as an international power. With its new role came a need to train Americans in foreign languages and to train foreigners in English.

For the first time in history, a method, Audiolingualism, would combine theories of language learning with those coming from other fields, namely linguistics and psychology.

Linguistics, led by Bloomfield and Fries, contributed a structuralist view of language (Finocchiaro & Brumfit, 1983). According to these structuralists, the study of language involved the identification and description of patterns of a language in an explicit and rigorous manner. In their view (Bloomfield, 1933; Fries, 1945), language is a system of structurally related elements for the encoding of meaning (phonemic, morphological, and syntactical systems), and language learning is the mastery of elements of language and rules to combine them. In other words, structure (i.e., phonemes, morphemes, words, phrases, and sentences) was the starting point of language learning. Also, in their view, language is basically oral, i.e., "speech is language." Language should be taught by (a) attention to pronunciation and (b) oral drills of basic sentence patterns. The sequence to be observed was (1) listening, (2) pronunciation, (3) speaking, (4) reading, and (5) writing. Fries had been working on instruction of Native-American languages to English speaking students, and one of the central figures in the method, a native speaker who worked as an 'informant', became a central figure in Audiolingualism.

Psychology contributed behaviorism. Skinner proposed an antimentalist view of learning in which (1) a stimulus elicits (2) a response, which can then be marked as good or not through (3) reinforcement. Translated to language training, the stimulus is the foreign language material, the response is the learner's reaction, and the reinforcement is the approval or disapproval from the teacher. Corrections served as proactive reinforcement, discouraging bad habits. For Skinner, verbal behavior was equal to other behaviors, and language learning was a process of mechanical habit formation. Speech had priority and analogy was considered better than analysis -- the practical implication was that grammar should be learned inductively.

Another innovative aspect of Audiolingualism was the use of technology. For the first time, technological artifacts other than textbooks played a major role in instruction. Tape recorders and audio visual materials contributed extensively and constituted an important part of instruction. Without native speakers to serve as informants, students used tapes of native speakers, with feedback coming from recording and listening to one's own output.

Audiolingualism has had a strong impact on language learning because it addressed a strong demand for oral proficiency and combined widely accepted theories of language and learning. Its main concern is oral proficiency; its foundation drill and practice -- mimicking and memorization (mim-mem). The student listens to a dialog, then repeats it, repeats it, and repeats it, until he has memorized the patterns in the dialog. Since the student is forming a habit, it is important to correct bad habits from the start. Correct pronunciation, rhythm, and intonation are emphasized. Following the dialogs, the student practices drills. Drills can be simply repeating a speech pattern, e.g., "I see the house," changing inflection, e.g., "I see the houses," replacing words, e.g., "I see the car," restatement, e.g., "She sees the house," and a number of other variations (e.g., completion, transposition, expansion, contraction, transformation, integration, rejoinder, and restoration). Since errors are thought to result from interference from the native language they are always corrected in the classroom. Teachers dominate the classroom activities; students have a reactive role. Addressing student's personal interests, e.g., talk about dating, is against the model and strongly discouraged.

Summarizing, Audiolingualism is a combination of structural linguistic theory, aural-oral procedures, and behavioristic notions of learning, with tape recorders and audiovisual materials playing central roles in Audiolingual courses. It relies on oral drills and practice of dialogs memorized through repetition. The linguistic syllabus contains items of phonology, morphology, and syntax that are taught in the order of listening, speaking, reading, and writing. The assumptions of Audiolingualism are that (1) foreign language learning = other learning, (2) we learn from experience, and (3) language learning is mechanical habit formation. Audiolingualism rejects the analytical model of the Grammar-translation method in favor of mimicry and memorization, and rejects the model of (a) exposure, (b) usage, and (c) absorption of grammatical structures of the Direct Method in favor of grammar as the starting point (Richards & Rodgers, 1986).

Audiolingualism began to fall in disfavor when Chomsky rejected its theoretical foundation, structural linguistics and behavioristic learning, in favor of the more mentalistic theory of transformational generative grammar (1959, 1966). Chomsky proposed the notion that language is an innate aspect of the mind and postulated a language processing module. Contrary to Skinner's view that language behavior is equal to other behaviors, Chomsky's view is that language is separate from other behaviors and so is learned differently; humans have a language acquisition mechanism, and sentences are not imitated and repeated but rather generated from competence. Chomsky showed that behaviorism and structural theories did not account for creativity and uniqueness of individual sentences.

Transformational Generative Grammar received wide support and destroyed the theoretical foundations of Audiolingualism, causing changes that are still rippling through the field of language instruction. Theorists attacked Audiolingualism as being unsound in terms of both language and learning theories. As one of the most welcome consequences of this new view, drill-and-practice decreased significantly in subsequent methods of instruction. To make matters worse, Audiolingualism was also found to be ineffective, with results below the levels expected; students found the procedures boring, and they had problems transferring knowledge acquired through repetition and drills to real-life situations.

The combination of technology and scientific theories in Audiolingualism was , however, an attractive innovation, and for a long time its overall impact camouflaged the weaknesses of the method. Today, pattern practice is still used to proceduralize knowledge, but it is now considered artificial and useless. Drills are boring and what students memorize often has little application in real life. Students parrot incomprehensible material, and while they become good at parroting, they develop no communications skills. Motivation is curtailed by the fact that students aren't allowed to discuss topics that are interesting to them -- instead they are forced to stick to the topic of the day; the emphasis on mastering patterns, and not on communication, makes interest fall rapidly. Grammatical sequencing forces unrealistic dialogs, and because production is supposed to be error-free, students usually feel anxious in the classroom. Although significantly better than Grammar-translation, Audiolingualism also falls short of preparing students to perform in real-life situations.

A need for change

The fall of Audiolingualism left a void, generating a renewed interest in teaching grammar deductively, i.e., teaching rules and then how to apply them. Chomsky's more mentalistic view of human language capabilities, and his notion that "competence precedes performance" provided a rationalization for a way of teaching that was much more amenable to the factory model of schooling that was becoming prevalent at the time. One method based on Chomsky's theory, the Cognitive-code Method, for instance, shifted back to teaching rules -- students learned about the language before, and separated from, learning how to use it. Such rationalization was, in fact, a distortion of what Chomsky had proposed. Chomsky did not defend that competence involved explicit knowledge of language structure, but only that the development of the necessary knowledge structures, conscious or unconscious, preceded performance. At any rate, despite its theoretical interest, Cognitive-code had a relatively small impact on second language instruction.

An increasing interdependence among European countries prompted instruction to focus on communicative proficiency rather than mastery of structures. In 1973, Wilkins proposed that, instead of describing language through traditional concepts of grammar and vocabulary, language should be described through a system of meanings that underlie the communicative use of language. Wilkins described two types of meaning: Notional categories (e.g., time, sequence, quantity, location), and Communicative Functions (e.g., requests, denials, offers). He later published a complete description of his view in "Notional Syllabuses" in 1976. Wilkins motivated a new definition of language compatible with the view of language as a tool for communication and language learning as the development of communicative competence. Language competence was redefined thus:

1) Language is a system for expression of meaning.
2) The primary function of language is interaction and communication.
3) The structure of language reflects functional and communicative uses.
4) The primary units of language are functional and communicative.

A number of methods emerged whose goal was to develop communicative competence. Psychologist James Asher (1977) created Total Physical Response (TPR), in which the student responds to imperative statements through actions. For example, the teacher says "Touch your face," and the student touches her face; "Touch your nose," and so on. Speech is delayed until comprehension has been internalized. At one point, when a student displays readiness to talk, he starts engaging in verbal interactions. The method has had some success in basic level instruction, and has received support from theorists who support the idea that acquisition is comprehension-based (comprehension precedes production). TPR's major advantage lies in helping students feel less threatened since they do not have to produce language for a few months.

Another method, Silent-Way, developed by Gattegno (1972, 1976), was based on Bruner's (1962) distinction between expository and exploratory (hypothetical learning). Gatteno's Silent-Way uses minimal modeling; the teacher models the pronunciation and intonation in the beginning of the class, and then guides students during the exercises by using colored rods that help indicate intonation, stress, etc. Despite its claims, in practice the method does not differ significantly from other methods such as Audiolingualism that emphasize accurate reproduction of sounds and sentences.

Total Physical Response and the Silent-Way did not necessarily follow a notional syllabus, which became closely associated with communicative competence, but adopted the new view of language competence described above. Other methods included: Counseling-Learning (emphasis is on creating a friendly atmosphere), English for Special Purposes (study of very specific content), and functional-notional approaches (based on Wilkin's notional-functional syllabus). One approach that combines a number of features from other methods and that has been receiving much attention in the field of language acquisition is the Natural Approach, proposed by Tracy Terrell in 1977.

Natural Approach

Terrell proposed a method based on naturalistic principles observed in second language acquisition studies (1977, 1981, 1982). Natural Approach develops communicative competence by using language in communicative situations without grammatical analysis, drilling, or grammatical theory. Unlike the Direct Method, the Natural Approach de-emphasizes teacher monologues, repetition, formal question and answer, and accuracy of production. It does emphasize (1) exposure, (2) emotional preparedness, and (3) listening before producing.

Terrell received the support of, and later joined forces with, Krashen, a theorist who had proposed an ambitious model of second language acquisition (Krashen & Terrell, 1983). This alliance solidified the position of the Natural Approach in the field of SLA. Krashen and Terrell reject the view of grammar as a central component, and argue that language teaching methods should be based on theories of learning, not on the structure of language. They emphasize meaning and the greater importance of words over grammar. Grammar is subordinate to lexicon. Language is a vehicle for communication of messages, and acquisition takes place only when people understand messages in the target language. Procedures used in the classroom do not differ from those of other methods, but the underlying theory changes the emphasis.

Krashen's theory of language learning, the monitor model, described in a series of papers and books (1982, 1985), provides the theoretical rationale for the approach. Krashen claims that the model is an empirically grounded theory of second language acquisition. The monitor model is extensively discussed in the literature (McLaughlin, 1978; Krashen, 1985). The model is based on five hypotheses, described by Krashen as follows:

1) The Acquisition/Learning hypothesis

There are two independent ways of developing ability in second languages. 'Acquisition' is a subconscious process identical in all important ways to the process children utilize in acquiring their first language, while 'learning' is a conscious process that results in 'knowing about' language. (p. 1)

2) The Natural Order Hypothesis

We acquire the rules of language in a predictable order, some rules tending to come early and others late. The order does not appear to be determined solely by formal simplicity and there is evidence that it is independent of the order in which rules are taught in language classes. (p. 1)

3) The Monitor Hypothesis

This hypothesis states how acquisition and learning are used in production. Our ability to produce utterances in another language comes from our acquired competence, from our subconscious knowledge. Learning conscious knowledge, serves only as an editor, or monitor. We appeal to learning to make corrections, to change the output of the acquired system before we speak or write. (p. 1)

4) The Input Hypothesis

The input hypothesis claims that humans acquire language in only one way -- by understanding messages, or by receiving 'comprehensible input'. We progress along the natural order (hypothesis 2) by understanding input that contains structures at our next 'stage' -- structures that are a bit beyond our current level of competence... We are able do understand language containing unacquired grammar with the help of context, which includes extra-linguistic information, our knowledge of the world, and previously acquired linguistic competence. (p. 2)

5) The Affective Filter Hypothesis

Comprehensible input is necessary for acquisition, but it is not sufficient. The acquirer needs to be 'open' to the input. The 'affective filter' is a mental block that prevents acquirers from fully utilizing the comprehensible input they receive for language acquisition.... This occurs when the acquirer is unmotivated, lacking in self-confidence, or anxious, when he is 'on the defensive', when he considers the language class to be a place where his weaknesses will be revealed. The filter is down when the acquirer is not concerned with the possibility of failure in language acquisition and when he considers himself to be a potential member of the group speaking the target language. (p. 3)

The principles underlying this model are that:

a) The goal of learning is to develop communicative skills
b) Comprehension precedes production
c) Production emerges when the learner is ready
d) Acquisition is central -- not learning
e) Affective filter must be low

The Natural Approach emphasizes oral communication skills based on student's needs. It is designed to create low affective filter, provide extensive exposure to vocabulary, and pay no explicit attention to grammar. Lessons are not organized around a grammatical syllabus. Activities are borrowed from other methods, such as the Direct Method, but during questions and answers the teacher tries to minimize anxiety by not requiring the student to speak until he is ready. Students are allowed instead to respond by using body language or, less compulsively, their native language -- emphasis is on communication, not form. Talking slowly and clearly, the teacher leads the student through interactions requiring simple Yes/No responses first, then progressing to word answers, and finally to full sentences. Activities focus on meaningful communication, not on form; teachers do not correct errors in form. The method does not introduce any novel procedure, but instead uses familiar activities in a framework that emphasizes comprehensible input in an environment that minimizes anxiety and maximizes self-confidence.

The purpose and nature of these activities are:

1) Supply comprehensible input to facilitate acquisition
2) Ensure that the learner does not feel anxious
3) Restrict grammar instruction
4) Do not corrected errors in acquisition, only in learning
2) Let teacher use only L2; let student use L1 or L2
3) Include grammar work only in homework
4) Talk about ideas, perform tasks, and solve problems

The learner does not *try* to learn, but engages instead in meaningful activities that require language. The teacher serves as a source of input, manages the environment and coordinates activities.

Unlike the cognitive or habit-drill approaches, the Natural Approach prepares students to communicate with native speakers in real-life situations. And unlike any single predecessor, the Natural Approach addresses a wide number of issues: readiness for production, the role of affect, the role of monitoring processes, sequence of learning, the role of non-verbal communication, and the distinction between learning and acquisition. It brings us back to the resource intensive approach of the Direct Method, except now it can count on technological resources that may compensate for the still limited availability of native speakers and exposure to the target culture. Unlike Direct Method, Natural Approach addresses a widespread need and rests on a more mature theoretical foundation.

The State of Language Teaching

The Natural Approach sits at the end of two major trends in language instruction. The first trend shows a shift away from teaching grammar towards teaching interactive skills. In the predominant view of language at the turn of the century, the structuralist view, language was a system of structurally related elements, and the goal of language learning was to master these elements. In this view, language competence is synonymous with grammar competence. Methods such as Grammar-translation, Cognitive-code, and Audiolingualism derive from this view of language.

Later, the structuralist view gave way to the functional view. In this view, language is a "vehicle for the expression of functional meaning," and the emphasis is on communicative rather than grammatical features of language. Wilkin's notional syllabus marked the beginning of a communicative-competence movement that influenced methods referred to as functional-notional methods. Proficiency, according to the functional view of language, is the acquisition of discourse competence -- the ability to understand and produce coherent text.

Recently, the notion of communicative competence has evolved to an interactional view, in which language is a tool for the "realization of interpersonal relations and the performance of social transactions between individuals" (Richards, 1990; Richards & Rodgers, 1986; Rivers, 1987a). Language is about performing transactions to have needs met, about getting a loaf of bread and sending mail back home, and about establishing and maintaining relationships. Proficiency, in the interactional view, is the ability to produce and recognize appropriate language in context. Instead of emphasizing any aspect of the language itself, the interactional view emphasizes how language is used in the target environment to perform social transactions -- the view promotes sociolinguistic competence. From this interactionist view of language, we can define the current goal of language instruction as follows.

Point 1 - The goal of language instruction is to

Teach people how to interact and perform social transactions using language.

Naturalistic Learning

The second trend in language teaching shows a shift away from academic instruction towards more 'naturalistic' ways of learning -- towards instruction that is more conducive to sociolinguistic competence. The main purpose of learning languages is no longer to read books in a foreign language or to seek intellectual development, but to do business and exchange ideas in an international community. When developing methods such as the Direct Method, Silent-Way, Natural Approach, Total Physical Response, and English for Specific Purposes, researchers attempted to capture the qualities of natural settings (e.g., type of interactions, physical surroundings, cultural idiosyncrasies) and based their methods on notions of naturalistic learning (e.g., how children learn, what happens during immersion, how our bodies interact with language).

In developing 'naturalistic' methods, theorists assume that the process of acquiring language is the same for L1 and L2, even though the conditions are different. When learning L2, the learner's internal condition is different because he has already acquired a great amount of knowledge, cultural background, and motor skills and doesn't need to relearn what he already knows. The external conditions are also different in that those with whom the learner interacts expect him to act his age, imposing constraints on the interactions they have -- motherese (i.e., simplified language) is restricted between adult speakers. However, whether we are learning L1 or L2, we learn by interacting with people in the process of achieving goals. We acquire knowledge about the language inductively, from examples, rather than through the memorization of decontextualized rules. These naturalistic methods are all based on the principle described below.

Point 2 - The principle underlying naturalistic learning

We learn language by using it as a tool to perform social transactions.

The Problems

This idea of capturing 'naturalistic' language use and learning and trying to bring them into the classroom is not new; it has inspired methods since the nineteenth century. In France, C. Marcel (1793-1896) proposed child language learning as a model for language teaching; the Englishman T. Prendergast (1806-1886) observed that children use contextual and situational cues to interpret utterances; and Gouin (1831-1896) developed a language teaching method based on his observations of children's use of language (Richards & Rodgers, 1986). Today, the shift towards naturalistic instruction is gaining strength because we are getting better at capturing those qualities of natural settings that are essential for language acquisition.

However, 'Naturalistic' methods, and all other methods, face a pervasive problem: the teacher. Despite the evolution of theories and the wider spectrum of variables taken into account, all methods become heavily dependent on this one central figure: the teacher. In fact, it is not exactly the teacher who is the problem, but the lack of resources that causes problems attributed to the teacher. In the hands of a good teacher, the Natural Approach is likely to help students prepare for real-life interactions; without a good teacher, and good teachers who are native speakers are hard to come by, learning is hindered. The main problems can be summarized as follows.

Point 3 - Main problems:

(a) Limited interactions with native speakers.
(b) Limited exposure to the target culture.
(c) Limited individual instruction and feedback.

If the purpose of language instruction is to help the student develop sociolinguistic competence, interactions with native speakers are a crucial part of the learning process. Wilga Rivers argues that "If communication of messages in the target language is the goal, then interaction must be present from the first encounter with the language" (1987b, p. xiv). We learn by doing, or in other words, we learn to use language by using it. So, authentic use of language, which forces the student to use what he knows, to create messages, and organize knowledge in his head, is an essential part of the learning process. This means that the student should learn by interacting with hotel receptionists, cab drivers, scientists, business executives or whomever she is likely to meet in real life. And to be effective, this practice must address immediate interests of the student. Instead, what usually happens in the classroom is that one person, the teacher, centralizes these experiences and students rarely get to practice what they feel like practicing.

Also, when interacting, people take into account the context and extralinguistic cues associated with the message. The physical surroundings and the precise context of a social transaction help people understand and organize the language being used. For example, if I ask someone in a supermarket to show me where the milk is, his explanation is likely to refer to elements in the surroundings (e.g., "aisle 1, with the eggs. It's easier if you go down this way"). Authentic conversations always happen within a larger context that supplies elements essential to the interaction. In the classroom, on the other hand, students are limited to practicing 'situated' dialogues (e.g., asking for directions to the YMCA) in 'unsituated' physical contexts (e.g., not in New York, but inside a classroom), which forbids allusion to concrete referents. Again, limited resources makes bringing this contextual information into the classroom almost impossible.

Compounding this problem, students are often forced to engage in activities that may be of interest to the group but are not always relevant to the individual. If a student is preparing herself to go to school in England, she should expose herself to the speech patterns that she will encounter there. Students like to participate in activities that address their concerns; they want to practice in situations similar to those they will encounter in real life. The problem is that letting each student choose a target culture requires much more than a single teacher can give.

In summary, although we know what students need to learn languages, classroom instruction lacks the resources needed to prepare students to perform in real life. The resources needed are (a) teachers, (b) native speakers and (c) physical surroundings, which are necessary for the student to learn languages by engaging in interactions with native speakers in interesting activities that are situated in and contextualized by its real-life surroundings, while receiving individualized feedback and instruction.

Conclusion

Table 1 shows a summary of the methods mentioned in this chapter. These methods evolved to address changing needs, incorporating new resources and better theories of language and language learning. Today, as we saw, language instruction is concerned with teaching students how to communicate with others. To do so, it must give students: (1) individualized practice (2) with native speakers (3) in authentic situations (4) that are interesting to them.

However, classroom language instruction consistently fails to produce real-life competence for the following reasons:

- Instruction is heavily dependent on highly skilled, hard to find teachers
- There is a shortage of native speakers to interact with
- Students have no access to extra-linguistic cues and the target culture
- Teachers cannot provide individualized instruction and feedback

Table 1
Methods of Language Instruction

Method	Influences & Characteristics	Originator/ Date
Grammar-translation	Read literature	Up to 1910
Direct Method Berlitz	Oral	Gouin
Audiolingual	Based on content Structural linguistics Behaviorism	Fries, 1948
Cognitive-code	Role of abstract mental processes Based on Chomsky's TG	Chomsky, 1959
Counseling-learning	Atmosphere is crucial, focus on condition Humanistic technique, ameliorate feelings of intimidation	Curran 72,76
Functional-Notional	Theory of language	Wilkins (76) Finochiaro & Brumfit (83)
Silent-way	(Condition) Feel secure	Gattegno 72,76
Total Physical response	Speech & Action. Derives from learning theory (proc & cond)	Asher 77
ESP - English for Specific Purposes	Focus on individual's needs	Robinson, 1980
Cooperative Learning	Social/Affective strategy	Slavin 1980 Dansereau et al. 83
Natural Approach	Importance of extralinguistic clues in language learning	Terrell 77, 82
Interactional	Interactionist view of language. Vehicle for interpersonal relations and performing social transactions	Richards & Rodgers (86) Rivers (87)

If we want to teach students how to interact in the target culture, we must solve these problems. Students need these resources (i.e., teacher, native speaker, physical surroundings) so that they can have all the experiences (i.e., interact, receive feedback, and receive instruction) in the right context (i.e., interesting activities, real-life surroundings) in order to acquire language. Unfortunately, regardless of how good (or bad) the theories are, language instruction consistently stumbles upon a common problem: lack of adequate resources. Native speakers, authentic situations, and exposure to the culture are never easy to come by. Consequently, given their limitations, teachers fall back into modes of teaching that are not conducive to communicative competence. Any solution must address these limitations and seek ways of providing the necessary experiences.

The next chapter shows how teachers have been using technology to address this problem.

Chapter 3

Technology in Language Learning

Introduction

To give students individual practice and exposure to native speakers, two commodities whose short supply have been afflicting classroom instruction, there has been a parallel development involving the use of technology in language instruction. Although not always explicitly stated in most methods, technology has been playing a significant role in exposing students to the target culture. An increasing reliance on materials such as audio-visuals, video tapes, multimedia, and more recently computers, brings some of the missing resources and experiences into the classroom and language laboratory. Chapter 2 discussed what has been done in the classroom; I'll turn now to how technology has been used. This will make evident in what ways technology perpetuates bad practices, and begin to show in what ways it can be used to solve existing problems.

Up until recently, technology was non-interactive, thus unsuitable for addressing many of the issues raised in the previous chapter. Computers, however, provide interactivity, and consequently, can provide individualized instruction and feedback, letting students take control and pursue their interests. Computers can help compensate for the shortage of teachers and native speakers, while addressing each student's interests and exposing them to the target culture.

This chapter traces the increasing use of technology in language learning, which, not surprisingly, parallels very closely the trends in language instruction. Each new application of technology, usually accompanying a particular method, contributes insights into what works and what doesn't. From tape recorders used in Audiolingual methods to multimedia simulations used in naturalistic methods, each has features that give us a sense of direction. The trend that becomes apparent, as we follow the development of technological solutions to language learning problems, seems to point towards building simulations in which students learn by interacting with people. Extrapolating, I will argue that simulations should not only replicate situations, so that they can be used as tools in naturalistic methods, but that they should incorporate naturalistic notions themselves. More than serving as simple tools, simulations can be implementations of innovative methods of language learning. The next sections show how (1) audiovisual material, (2) computers, (3) multimedia exploratory environments, and (4) multimedia simulations have been used in language instruction. The progression helps illustrate the trend towards such simulations.

Audiovisual

Language teaching has been relying heavily on records, tapes, slides, films, and videos. During the sixties, for example, tape recorders introduced not only a new type of presentation but also a new type of feedback: Students could record and listen to themselves. This technology was at the core of the Audiolingual method of instruction. Later, TV and videos brought to the classroom extra-linguistic elements of conversations that were not captured by tape recorders (e.g., gesticulation, physical surroundings, facial expressions), and this ability to capture extra-linguistic cues suited the emerging 'naturalistic' methods of language instruction.

For the past thirty years, audiovisual material has been exposing students to foreign languages and cultures. Videos have not only captured essential aspects of foreign environments but have also delivered entire language courses. Today, on a regular basis, TV networks broadcast a number of language courses. For example, engaging teachers, native speakers, and authentic surroundings merge in courses for Japanese, *Let's learn Japanese*; Italian, *Bon Giorno Italia*; and French, *French in Action.*

French in Action

The most complete of these courses, *French in Action*, developed at Yale University, combines textbooks, cassette tapes, and 42 video tapes that contain the core of the course. In the videos, an enthusiastic and captivating teacher guides the student through the experiences of a group of French people going about their lives in France. Shot in France with native actors, *French in Action* introduces the student to characters, such as the attractive Mireille, and follows them through the day. In one scene, for example, Mireille goes to a newsstand to buy a newspaper. She is greeted by the newsstand lady, "Bonjour mademoiselle Mireille;" they have a brief interaction; Mireille gets the paper; says good-bye, and leaves. After this vignette, *French in Action* shows other people greeting, then shows Mireille at the newsstand again, pausing for the student to practice on his own. After that, the teacher makes some comments, drawing the attention of the student to important points, and the story continues. "The objective of *French in Action* is total language teaching through planned immersion -- the presentation of French language and culture in a way that simulates the experience of actually being in France... Above all, it makes [students] aware that the acquisition of a language does not merely entail learning grammatical structures but depends on a complex system of verbal and non-verbal communication, gestures, looks, attitudes, behavior, intonation, and cultural conventions and assumptions (Yale, 1987, p. vii).

French in Action incorporates many of the ideas in current models of language teaching. Its interesting story, well-designed scenarios, and the charisma of the teacher and actors are hard to match under ordinary classroom conditions. Besides, *French in Action* packs an enormous amount of cultural information in its 42 video tapes. The major problem with *French in Action* is that the student doesn't engage in interactions. In the practice modules, the student repeats lines and records utterances, using tape recorders as in the Audiolingual method, but does not engage in any interactions. Exposure is there, but the interaction essential to learning is missing. This is a limitation of the technology itself. Video technology is limited by its non-interactive nature, which forces students into passive roles. And for language learning, passive exposure is ineffective. Language acquisition, as Chapter 2 showed, is a byproduct of interactions.

Computers

One potential solution to this interactivity problem is to use computers. Unlike videos, computers can exchange information with the student and give feedback that is essential for language acquisition. Microcomputers, which became widespread in the early eighties, seemed to provide the interactivity missing in audiovisual, and were thus a welcome innovation. The first attempts at using computers, however, were disappointing. Unfortunately, computers didn't mesh with multimedia back then, and instead of using computers to complement videos, designers used them to implement outdated structural and behavioristic models of instruction. Instead of building on currently available technology and adding interactivity to videos, they added interactivity to textbooks. As a result, the first Computer-Assisted Language Learning (CALL) systems did little more than provide vocabulary lists and administer multiple-choice tests. These early systems helped teachers grade, but didn't help students learn. Instead of adapting computers to address language learning problems, developers adapted their solutions to technology.

As an example of this reverse thinking, one of the reasons Alan and Pamela Maddison defended transformational grammar as a useful approach was that it was 'suitable to computers' (1987). Transformational grammar, they argued, could be used to teach students to "generate a large number of acceptable utterances and how to understand them." They explain that "if the rules are adequate, the computer can be used to generate kernel sentences and carry out transformations; and to match sentences generated by the students." (Maddison & Maddison, 1987, pp. 20-31). They were defending pattern practice based on the fact that computers can use transformational grammar to implement them. They were adapting needs to the constraints of technology.

This type of thinking led to the development of computer-based word lists, sentence lists, dictionaries, and translation programs, and systems that focused on grammar, verb conjugation, and other structural aspects of language that ostensibly suited the computer. Multiple-choice tests and fill-in-the-blank exercises, typical of classroom tests and textbooks, were also computerized. In general, the only beneficiaries at this stage were teachers, who no longer had to do the grading. These systems perpetuated bad practices, and students gained very little from them.

The strong initial influence of structuralism, behaviorism, and technological limitations led many teachers to see computer-based instruction as synonymous with outdated paradigms. Many of those genuinely interested in helping students develop communicative competence were discouraged by early CALL programs. Fortunately, a few went on to devise new and better ways of using the available technology. I look at three cases next.

Supplementing classroom activities

At the University of Delaware, Braun and Mulford (1987) introduced computer-assisted instruction as part of a radical restructuring of the entire first-year French curriculum. They threw away grammar drills, reliance on textbooks, and the language laboratory "with its mindless repetitions," and replaced them with active learning of oral French with inductive acquisition of grammar, a writing workshop, and a computer classroom.

In the classroom, Braun and Mulford used a modified version of TPR (Total Physical Response), obtaining good results after short 40-hour terms. During the first few weeks, students act out commands without producing speech. Grammar accuracy is acquired gradually, and their method usually leads to better comprehension than Audiolingualism. When students finally begin to produce, they find the transition to production exciting and are surprised at themselves. From then on, vocabulary introduced through TPR is used in role-playing activities. At the end of the 14-week term, students have a basic stock of adverbs, prepositions and articles, conjunction, and noun-adjectives, in a total vocabulary of about 400-500 words (Braun & Mulford, 1987). Grammatical notions such as 'qui,' 'que,' negations and interrogations by intonation are also acquired. What is more important, they 'acquire' rather than 'learn' (refer to chapter 2); they feel confident, and display good competence.

Outside the classroom, homework helps students review classroom activities; the writing workshop helps them to think about the things they learned in class through questions, reviews, quizzes, dictations, and completion of stories; the computer classroom helps them with vocabulary and verbs.

In the computer classroom, two computer programs work on (1) verb forms, and (2) vocabulary. In the verb form lessons, the less interesting of the two, the computer asks the student to conjugate a verb in a certain tense (e.g., Verb: "Parler," Tense: "Passé Composé") and then provides feedback according to the student's answer. The more interesting vocabulary lessons involve sight, hearing, and touch. Unfortunately, these are not all available at the same time; the student has to choose one format: picture, audio, or word arrangement. In the picture format, a picture appears on the screen, and the student has to fill in the corresponding word in a sentence completion exercise. In the audio format, a sentence appears on the screen with a word missing while the complete sentence is played over the headphones. The student types in the missing word. Although this latter format was expected to be the most promising, it was the word arrangement format, in which students organize and memorize words, that proved most popular. In word arrangement, sixteen words appear on the screen, and the student reorganizes them anyway she wants (e.g., moving them within a window), usually placing them on the screen according to an imaginary story with the words serving as signposts. Once done rearranging, the student sees the words replaced with their translations into English. When she is ready, the program begins the recall exercise. Only the initial letter of the French words reappear, and the student has to retype them.

Despite its simplicity, this program helped students reinforce the vocabulary used in class, serving as a useful complement to classroom activities. When properly positioned within a larger context, even simple text-based applications may be effective to address specific language learning needs. Such programs might be useful as sub-modules of larger computer-based language learning environments.

A microcomputer game in French culture and civilization.

In many cases, the impact of a CALL program depends more on how it is used than on its design. In 1983, Betje Klier developed a game, Poker Pari, to be used in a contest among high-school students who gathered annually at the Texas French Symposium (1987). Klier had noticed in class that "students regarded as a treat any activity dealing with culture and civilization." Klier decided to exploit their interest in culture and civilization along with game elements that high school students liked, such as card games, quiz shows, video games, and computers.

Poker Pari gives points for correct single-stroke answers to multiple choice language, culture, and civilization questions. Thirteen topics comprise a deck of cards. The computer, an Apple II Plus, selects seven topics at random and presents questions one at a time. The student has the option of rejecting two topics, and the system imposes no time constraints.

Listed below are the topics selected.

01 - Grammar, elementary
02 - Grammar, advanced
03 - Vocabulary, elementary
04 - Vocabulary, advanced
05 - History, government, and education
06 - Literature (authors and their works)
07 - France (geography, cities, provinces, and products)
08 - Monuments and masterpieces
09 - Outstanding persons and their works (artists, musicians, etc.)
10 - Francophone countries
11 - Paris
12 - Quotes, proverbs, and idioms
13 - Kings, queens, and castles
14 - Wild card: anything goes

Poker Pari announces the game and asks the student if she wants instruction or a hand. Easy or Hard options with different scores (5 and 8) are available for each topic. Questions are not flagged, so that subsequent hands may include the same questions. This feature was initially criticized, but later considered important to encourage students to retain knowledge since it could help in the future. A few questions appear in English, but most appear in the target language. Some examples follow (Klier, 1987).

"ELLE A MAL AUX DENTS."

1. She had bad breath.
2. She curses frequently.
3. She has a toothache
4. She has dirty teeth.

"SI VOUS AVEZ MAL AUX CHEVEUX," you have

1. A headache
2. A bad haircut
3. Bowed legs
4. A hangover

Or "Voltaire: TOUTES LES GENERALISATIONS SONT FAUSSES"

1. ET RASONNABLES.
2. SAUF LES MIENNES.
3. SOUS MICROSCOPE.
4. Y COMPRIS CELLE-CI.

"NUL N'EST PROPHETE"

1. A Rome
2. En son pays
3. Sans argent
4. Qui finit bien

The most interesting aspect of this game, as it turns out, is the way it is used. Klier decided that students would compete in teams, and that proved to be a good idea. Students found playing as a team more fun, and were motivated to work with their friends to prepare against their opponents, the other schools. The system served as a catalyst for learning. It pulled students together in a cooperative learning environment, in which students discussed and helped each other understand and memorize the topics in the system.

Over three years of personal interviews, Klier found that the objectivity of judging, the possibility of getting easy questions, the potential of avoiding topics not desired, and the novelty of the activity motivated students to prepare for the contest. Students said that playing in a team (a) intensified their feelings of success, (b) helped them remember answers that they associated with experiences with team members, and (c) made it more fun. Students asked to play "unofficially" after their game scores had been posted, because they wanted to see if they could achieve a higher score.

Although interesting, Poker Pari's game qualities do not qualify it as a good language learning environment. The amount of information it exchanges with a student is severely limited by its text-based, multiple-choice nature. Nevertheless, Poker Pari exemplifies one of the best ways in which we can use technology: as a catalyst. Peer acceptance and pressure are major factors in learning, particularly in language learning, and Poker Pari threads students together in collaborative activities.

The Dark Castle - An adventure in French

In some cases, even text-based, multiple-choice applications manage to move beyond drill and practice. In 1981, after seeing ELIZA (Weizenbaum, 1966), a conversational simulator, and 'Colossal Cave,' an adventure game, Saunders (1987) developed *The Dark Castle,* an adventure in French that trains students in narrative reading.

Saunders wrote a 360-page story starting from a picnic in the countryside, going through forests, boat trips, bridges, tunnels, and finally arriving at a castle with staircases, rooms, corridors, and strange inhabitants. He then implements the story on a microcomputer, with the following goals:

1) To motivate reading by providing an interesting stimulus and allowing a degree of control over the story.

2) To support pupils' reading by helping them to keep going when they have problems, thus ensuring progress

3) To promote accuracy by encouraging group discussion and reference to the program's dictionary.

4) To show that reading a foreign language can lead to enjoyment and satisfaction

Saunders found groups of three to five students to be ideal because each student could see clearly, contribute to the discussion, and take turns at the keyboard. Since all interactions were single key commands, the program was very simple to use. Teachers provided help with the language, and an on-line dictionary provided much of the information needed for students to understand the adventure. Students also benefited from working and exchanging information with people with different ability levels.

Saunders' students were harsh critics. They found Saunders' story too long and lacking exciting elements; they thought that the beginning dragged too long before the first choice and that the ending was bad -- multiple endings could have been better; and they wanted more illustrations. Another problem Saunders noticed was that instead of using French, students discussed in their native language. Nevertheless, despite students' complaints, Saunders showed that computers did not have to be limited to testing and drills. The Dark Castle obviously needed improvements (e.g., adding videos, animation, graphics, sounds, and a better interface), but it began to point beyond drills, towards computer-based simulated environments.

Multimedia Exploratory Environments

Things began to improve when language teachers became more competent with computers and tools like Hypercard. They began to use graphics, sound, and text combined with intuitive interfaces to give students access to words and phrases in more useful ways. At the same time, multimedia was becoming accessible, and hypermedia systems began to offer computer-driven interactive video. Interactive Video Disc (IVD) allowed a good degree of control over the material, giving students the ability to address individual needs and, in the classroom, the IVD exploratory environments enabled teachers to adjust the material dynamically to address immediate interests.

After the initial crop of text-based CALL programs, it was becoming clear that language learning systems had to integrate audio and visual information. Visual information was shown to assist in comprehension, inference of meaning, and inference of connections between sentences (Doughty, 1991). Videos also depicted authentic language usage, provided visual and aural information, and allowed access to extra-linguistic cues and interesting scenarios. Combined with computers, videos enabled the creation of interactive multimedia systems that enabled (a) user involvement and participation, (b) self-paced learning and user control, (c) audiovisual teaching and learning, (d) immediate feedback, and (e) tracking and affordability (Slaton, 1991).

Instead of just presenting information in videos, interactive multimedia allows students to select desired topics to address their own needs. Multimedia systems re-purpose video discs and organize them into maps and lists of topics, allowing students to use them as navigational tools to explore the target environment.

The Zarabanda Notebook

One of these multimedia navigational tools is The Zarabanda Notebook (Underwood, 1991). The Notebook is based on a 25-episode soap opera called Zarabanda, produced by BBC in 1973 for teaching Spanish. Zarabanda is the story of Ramiro, a mechanic, who goes to the city to seek his fortune. Shot in real locations with native speakers, Zarabanda is rich with linguistic and cultural information.

The goal of the Notebook is to organize the information in Zarabanda so that students can see and hear information in the repurposed video at both the language and the story level. Developed in Hypercard, The Notebook has a number of maps: (1) Ramiro's room, (2) The Village, and (3) Maps for each scene. In the room map, the user can click on objects or other maps. Clicking on the book, for instance, starts an introductory video clip; clicking on Ramiro's girlfriend's picture starts a brief textual and graphic presentation of the cast of characters for the first episode. Clicking on camera icons shows still-frame of the characters.

In the Village Map, clicking on TV icons offers the student a choice between previewing or watching the scenes. The preview is a map that shows how scenes are sequenced in time and space. The student chooses to watch only a few scenes or watch the entire sequence. When viewing the scenes, buttons allow the student to control what he wants to see in a number of ways:

1) View whole scene
2) Stop
3) Show script in English (Spanish is on the screen)
4) Listening practice
5) Visual dictionary
6) Go back
7) Go next
8) Go to the MAP
9) Go to Ramiro's room

interaction is very different from participating in one; controlling information is not the same as exchanging information. So, the major problem remained unsolved: To learn how to interact, one must interact. In social interactions, people are never passive; instead, they use language as a means of establishing and maintaining relationships by constantly exchanging information. Interactions involve feedback, clarification of statements, and extra-linguistic exchanges that are essential in social transactions. This song-and-dance, necessary in performing social transactions, is never necessary when simply controlling information in exploratory systems.

To help students develop communicative competence, what is needed are programs that give students not only conversations to watch but also conversations in which to participate. When Krashen says, in his input hypothesis, that we acquire language by being exposed to comprehensible input, he does not mean to say that by watching French TV all day we will become proficient in French. Being exposed to comprehensible input must occur in contexts in which we can make sense of what we hear and in which we negotiate meaning by interacting with people, even if not using language. What we need is to simulate the target environment. By simulating it, we give students a chance to participate in face-to-face interactions with foreigners.

Athena Language Learning Project (ALLP)

At MIT, a number of programs have been developed that focus on developing communicative competence, some of which focus on interactions (i.e., they simulate conversations). The goal of MIT's Athena Language Learning Project is not mastery of the grammatical and syntactic code, but the ability to understand language in a culturally authentic and task-centered situation (Lampe, 1988; Morgenstern, 1986; Murray, 1987; Murray, 1990). In the short run, the goals are to implement simple interactions based on matching and anticipated responses. The long-term goal is to build full simulations that will occur within stories that provide a natural link to classroom activities.

MIT stresses communication and interaction in what they call Language Learning through Interaction via Software. It is firmly grounded on the interactional notion of competence, in other words that "language learning takes place in an interactional context in which the learner expresses, interprets, and negotiates meaning with other interlocutors" (Morgenstern, 1986, p. 25). "Language is not 'presented' as an abstract, static, and closed system, but is experienced as dynamic interplay." Dialogs are essential. "Language is seen as a negotiable system of meaning, expressed and interpreted via the social interaction of reader and text, or between speakers culturally coded situations rather than on a closed system of formal lexical and grammatical rules." (Murray, 1990, p. 22)

One of the aspects that the Athena Language Learning Project is working on is the natural language processing module. The quality of a simulation depends on effective interpretation of the input accompanied by appropriate responses. And effective interpretation depends on how well the system processes input (Morgenstern, 1986). Most systems currently available have limited input capabilities through multiple-choice or keywords. Artificial Intelligence techniques may give the illusion that the program understands the input (typed, not speech), so that learner's develop command of the language by interacting with simulated interlocutors.

This emphasis on communication and natural language processing have been guiding the development of a number of prototypes at MIT. Prototypes developed or being developed include versions for French (Direccion Paris: Part I A la rencontre the Philippe, Part II: Dans le Quartier St. Gervais), Spanish (No Recuerdo), Japanese (Good-bye this year's love), and German (LINGO). Two of these systems, *A la rencontre de Philippe* and *No Recuerdo*, include interesting features that make both worth mentioning. *A la rencontre de Philippe* does not use natural language, but it implements interesting exploratory features; *No Recuerdo* is not yet fully implemented, but it will use natural language to implement simulations of interactions.

Direccion Paris: A la rencontre de Philippe

Direccion Paris, a project directed by Gilberte de Furstenberg, includes two modules: *Dans le Quartier St. Gervais* and *A la rencontre de Philippe.* The former is a navigational tool analogous to Zarabanda, discussed earlier. The latter is an exploratory system that includes some simulation of interactions.

"A la Rencontre de Philippe is a fictional story about a young writer who must solve his romantic problems or find a new apartment in Paris. Filmed on location, the story features numerous options for action and opportunities to use maps, answer phones, and movie maps of authentic Paris apartments" (MIT, 1990, p. 5). The student helps Philippe, who has been thrown out of the apartment by his girlfriend, locate an apartment in Paris.

The system includes a map of the area and shots of apartments in various locations, including floor plans and route to the apartment. The student can see and hear language in action. She interacts with fictional characters, answers questions, and even listens to phone messages recorded in a simulated answering machine. When watching a scene, the student can stop, reverse, replay, and skip portions of the video, which can be accompanied by subtitles in L1 or L2. Flagging words in the subtitle gets a glossary. Also available are "cultural notes" that provide backup information on idiomatic expressions and historic locations.

The menu choices in *A la rencontre de Philippe* are not right-or-wrong quizzes, but occasions for the student to intervene in the story and have an effect on what happens next. The student interacts with Philippe by selecting from menu answers. She goes to Philippe's apartment, walks around, listens to messages, travels around in Paris, and can even use the phone to dial other characters in the story -- they are always out and the student has to leave a message. The story ends in one of seven endings. *A la rencontre de Philippe* provides three ways of accessing information: (1) Episode, (2) Linguistic function, and (3) Characteristics of speech (e.g., ffff). These different indices allow students to zoom into different aspects of communication (gesticulation, idioms, etc.) using a point-and-click interface.

The system has a number of interesting features. The REVIEW window is a navigation tool that allows the user to go back and forth within a dialog, with two types of transcript -- according to the level of proficiency of the student. The notation used in these transcripts (e.g., T[u] [s]as plus qu'a , where [u] & [s] aren't really enunciated) helps with pronunciation. *A la rencontre de Philippe* uses two sound tracks, one providing a clearer speech, and buttons that assist in comprehension and give students control (e.g., suggestion, help, examples).

The system was designed to be used in the classroom, and the various views are useful in that context. The map-like simulations are rich and authentic, and provide the flexibility to walk around in the simulated space. The exploratory aspects of *A la rencontre de Philippe* are very good. On the other hand, the simulation aspects are still limited. The system does not accept free input in French, and the level of interactivity does not allow conversational simulations. As of today, *A la rencontre de Philippe* is mostly an exploratory system, but in the future it is expected to include more flexible interactions with the people in the simulated environment.

No Recuerdo

No Recuerdo, directed by Morgenstern, is an interactive fiction with simulations of conversations (Morgenstern, 1986). It combines natural language processing and interactive video disc, letting students interact with people in the simulations through typed input. "No Recuerdo is the fictional story of Gonzalo, a Colombian scientist who lost his memory while working on a secret [virus]. The adventure was videotaped in Bogota and features a maze-like series of plots and simulations that makes the student help Gonzalo get his memory back before the amnesia plague hits." (MIT, 1990, p. 5)

As the story unfolds, students engage in interactions with people, trying to get information that may lead to the location of the dangerous viral material. *No Recuerdo's* many branches lead different students to obtain different information that is later used in the classroom to corroborate a story. Engaging the student in the problem puts him in close cultural proximity, so that instead of participating as a tourist, a passive observer, the student becomes an active participant in the story.

Students need good discourse and listening skills to progress within the story, and uncover where Gonzalo hid the material that could cause an outbreak of amnesia in Latin America. They converse by typing sentences and receive responses in the form of a full video segment or a still picture of the interlocutor with audio or text superimposed. The goals of the program are listed below:

1) Vocabulary learning
2) Reading and listening comprehension
3) Cultural awareness
4) Practice with conversation strategies
5) Writing (because input is typed)

No Recuerdo purports to imitate real life conversations with native speakers, and according to current views of language learning this is one of the most important experiences students need (i.e., interactions with native speakers). The natural language interface offers the advantage that students have to recall, not just recognize, the knowledge necessary to perform a task. In addition, if the final version uses full video, instead of still pictures, it may also provide the extra-linguistic cues that students need to place information in context (i.e., exposure to the target culture).

The literature available describes *No Recuerdo* at a fairly high level, not covering specific solutions to various design problems that would be interesting to consider. For example, since the target audience that can benefit from a language training program that use typed input is restricted, it would be informative to know more about its target audience. Also, since the use of a natural language interface requires mechanisms to compensate for the fact that computers can't understand extra-linguistic cues -- for instance by providing buttons to express confusion -- it would be interesting to see the solutions implemented to address these problems. At this point, since *No Recuerdo* is not yet available, an analysis at this level is not feasible.

Nevertheless, *No Recuerdo* raises an interesting distinction. To learn a language through (a) practice and (b) exposure, the two experiences that interactive fictions provide, is analogous to learning auto mechanics by having broken cars to fix and auto mechanics to watch. The problem with this model is that it lacks the support system that is necessary to enable experiential learning. We need tools (i.e., manuals, specification sheets, wrenches, and screwdrivers) and guidance. We need somebody or something to give us some clue as to where to start, to instruct us, to assign us tasks at our level of competence, and to give us hints. Passively watching others fixing cars is not enough. We need to ask questions and we need feedback on our performance so as to be aware of what we don't know and what we need to learn. In language learning, having this support system is like having our mother helping us through interactions with others, telling us to go ask dad for a candy, showing us how to ask for it, giving us hints, correcting us, and explaining things about people to us. So the distinction between interactive fiction and this richer environment is that the former is a practice arena that assumes students receive the necessary support from a source outside the arena, and the latter is a complete learning environment, in which students not only practice and observe but also receive the necessary support to learn from experience.

The previous chapter showed that, given an interactionist view of language (i.e., teach people how to interact) and a naturalistic method of instruction (i.e., students learn by interacting with others), three problems need to be solved. Technology must provide students with:

a) Interactions with native speakers
b) Exposure to the target culture
c) Individual instruction and feedback

Systems like *No Recuerdo* are practice arenas. They assume that the necessary individual instruction and feedback are provided elsewhere. Now, if we want technology to address all three problems, then technology must also provide the individual instruction and feedback and the supporting tools necessary for experiential learning to occur. In this sense, Dustin, the system that I introduce later, is a learning environment. In Dustin, students engage in interactions and watch others while having access to supporting tools (e.g., dictionary, translation, recorder, transcripts) and receiving individual attention from a tutor.

Conclusion

The CALL systems described in this chapter can be classified along two dimensions: (1) whether the student has an active or passive role, and (2) whether it is based on structural or interactional approaches (see Table 2). Systems whose goal is to develop communicative competence vary from video-based instruction (e.g., French in Action), to exploratory systems (e.g., Zarabanda), to simulations of real life situations (e.g., Direccion Paris, No Recuerdo).

Table 2

Types of technology-based language learning systems.

	Structural	Interactional
Active	**Practice & Drill** • Word Arrangement • The Dark Castle	**Simulation** • No Recuerdo • A la rencontre de Philippe
	Q&A • Poker Pari	**Exploratory** • The Zarabanda Notebook • Dans le Quartier St. Gervais
Passive	**Memorize rules** • Early 'page turners', or computerized books.	**Exposure** • French in Action

Table 3, below, compares the systems introduced in this chapter. If our goal is to help students develop communicative competence, we must build simulations that let students practice, observe, receive individual instruction and feedback, and access tools that support them during interactions. Multimedia conveys essential visual and aural information, capturing the behavior of individuals in the real life environment. Multimedia-computer-based simulations can transport us to the streets of New York and provide the experiences missing in the classroom.

Table 3

Features of Dustin and other CALL systems.

Feature	French in Action	Poker Pari	Dark Castle	Zara banda	Direccio n Paris	No Recuerdo	Dustin
Interactive	√	√	√	√	√	√	√
Interactionist View	√			√	√	√	√
Naturalistic Approach					√	√	√
Exposure to the target culture	√			√	√	√	√
Interactions with native speakers					√	√	√
Individual instruction and feedback							√

Creating simulations of real life, however, is a difficult task. Not only does creating simulations require the contribution of various experts (i.e., language specialists, instructional designers, educational researchers, programmers, AI scientists), but it also requires solving three major problems. First, a number of difficult technical problems have to be solved (e.g., natural language interface, knowledge representation for simulations). Second, there are no authoring tools to help develop such simulations; we have to develop them ourselves. Third, and most important, designing simulated environments requires a careful analysis of what needs to be simulated and of how to put them together to assist in language acquisition. Simulations differ in content and architecture, and selecting the features of the environment, combining them with tools, and determining the processes and interface that promote learning is the most important and difficult task in creating learning environments. These design choices, ultimately, determine the success of the system. I discuss design choices in the next chapter.

Chapter 4

Design Issues

Introduction

Chapter 2 described the shift in language instruction from viewing language as a set of related symbols, the structural view, towards viewing language as a tool to interact with other people, the interactional view. Correspondingly, methods of instruction have been moving away from teaching language itself, removed from its use, towards teaching how to use language to interact with others in natural settings. Naturalistic methods try to bring real-life situations into the classroom so that students learn by interacting with others, as they would in real life. As Chapter 2 showed, however, implementing these natural settings presents major problems. Resources are not always available, and bringing native speakers and authentic, interactive situations into the classroom is rarely feasible. As a possible solution, Chapter 3 suggested that computer-based technology, incorporating audio and video, can bring these experiences to the student. Mirroring the trends in language instruction, computer-based language training has been moving away from passive presentation systems towards systems that simulate real life. Most existing systems implement limited interactions, usually simply allowing students to navigate and watch interactions. Ideally, however, authentic simulations can give students individualized exposure to the target culture and engage them in active interactions with native speakers. If properly designed, these simulations can help students develop language skills by interacting with others as they would in real life, acquiring language as a byproduct of performing social transactions in the target culture.

This chapter discusses the design of these simulations. It discusses the qualities that simulations must have when properly designed to assist in language acquisition. Since learning a language while in a foreign culture, which is the experience simulations try to replicate, involves a number of variables, looking at the elements involved in such experiences helps us determine what is important to simulate. When living in Japan, for example, learning is influenced by (1) the situations, (2) the learner, (3) their interactions, and (4) the resources available. Each of these elements introduces clues that guide the design of language learning environments. By carefully selecting those characteristics of each element that influence language acquisition and by carefully addressing the problems involved in transferring them to a computer-based simulation, I identify design principles for effective simulations. A good way to start is to take a good look at what happens when one moves to a foreign country.

What's so good about living in Japan?

"Living in Japan is the best way of learning Japanese"

If I move to Japan, Japanese immediately becomes the vehicle of every social transaction, the instrument of my survival, well being, and social acceptance. I need to eat, conduct business, earn money, interact with others, feel accepted in the community, express frustration, convey sadness and loneliness, and find companionship; all of this is heavily dependent on my knowing the language. I can use alternative means of communication: gesticulation, pointing, and mime, but I definitely need language to say something like "Express delivery please." I go at it with an insatiable thirst. I ask people to help me; I hang on to phrases as if my life depended on them; I buy books, dictionaries, tapes, and translating machines; I hire a private tutor; take classes; ask friends to teach me; watch TV; read the papers. People correct me or restate what they thought I wanted to say, showing me the right way of saying things; they get angry, making me realize that something is wrong; they frown, making it clear that I did not make sense. I expose myself to situations in which I have to interact with native speakers; e.g., finding myself a girlfriend who speaks the language. I drink, eat, and breathe the culture twenty-four hours a day, paying attention to its customs and social protocols, afraid of committing gaffes. Considering what I do, it is not surprising that living in Japan is an effective way of learning Japanese. The language is so important in my life that I almost can't help learning it.

Obviously a number of elements contribute to learning in the Japan-experience. The person is highly motivated to learn Japanese to go about his life. Language is crucial, and the drive to survive and prosper stimulates learning. The goals that the individual wants to achieve in the target culture affect his role in the environment. He spends twenty-four hours a day immersed in the target environment, hearing, seeing, and interacting with people who use only Japanese to communicate. The student is exposed to the context, in which language is used, including physical, social, and task settings as they occur in real life. He participates actively, engaging in interactions that require the use of language. These interactions force the student not only to recognize words but also to recall and adapt knowledge to the situation. He receives help. Native speakers, or tutors, correct errors or suggest better ways of saying things, and the environment itself is fortuitous in that it provides referents that assist in communication. The way people react, their behavior, provides the learner with essential feedback to adjust behavior. The student tries to understand patterns of the language, questions how words are used and what they mean in different contexts, in other words, he engages in analysis and reflection. When communication breaks down, due to insufficient competence or performance variables, he uses extralinguistic means of communication that help smooth social interactions. When at a loss for words, he resorts to tools like dictionaries, recorders, and translating machines. The social context and the interactions with others elicit emotions, affect, that influence learning, and the personality of the individual influences the degree of exposure and his willingness to try things out. Finally, different people use different learning strategies, because they learn in different ways. See Table 4 for a summary of the variables that influence language acquisition when living in the target culture.

Table 4

Variables that influence language acquisition in natural settings.

Variable	Description	How it influences learning
Motivation	Highly motivated. Personal needs have to be met; language is a powerful tool to satisfy them.	The more goals depend on language the more motivated students will be.
Goals	Survive, make friends, establish oneself, thrive.	Dependency on language proficiency affects other variables.
Immersion	Completely immersed, twenty-four hours a day.	Extent of exposure is thought to have a stronger impact than age.
Context	Native speakers, target culture and environment, real life situations.	Higher relevance of language used in real life for real life purposes.
Active participation	Engage in interactions continuously.	Forces retrieval and adaptation of existing knowledge structures.
Help	People help, friends help, tutor helps.	Non-threatening situations foster exploration and corrections help eliminate errors.
Behavior/Feedback	The environment is fortuitous, people give feedback all the time.	Crucial for fine tuning language skills.
Analysis and Reflection	Study the language, think about the culture and social protocols.	Detect patterns in the language, build abstractions that enable transfer.
Extralinguistic Communication	Gesticulation, mime, pointing, facial expressions.	Helps handle communication breakdowns. Eases interactions.
Tools	Use dictionary, tape recorder, translating machines.	Artifacts help us with specific problems.
Affect	Emotions influence learning and interaction with the environment.	Fear, threat, security influence openness to exploration and learning.
Personality	Traits determine student's relationship with the environment.	An outgoing person will have more exposure and willingness to experiment.
Learning strategies	Tricks used to learn the language itself.	Personal learning styles, cognitive strategies.

These variables cover a wide range of phenomena -- from tools in the environment to characteristics of the learner. Organized in groups, they address four important elements of real life experiences: (1) situations (e.g., context and behaviors), (2) student (e.g., goals, motivations), (3) interactions (e.g., sequence of events, freedom of choice), and (4) resources (e.g., tutor, books, etc.). See Table 5 for a summary of the aspects of the environment that each variable addresses.

Table 5

Variables that influence language learning and how they relate to aspects of the environment.

Variables	Shape
Context Behavior/Feedback	**Situations**
Motivation Goals Analysis and Reflection Affect Personality Learning strategies	**Student**
Immersion Active Participation Extralinguistic communication	**Interactions**
Help Tools	**Resources**

Next, I discuss each of these four elements (i.e., situations, student, interactions, resources) and the problems involved in simulating real life using current technology. I define a set of principles for the design of simulations, which I then use to develop the simulation described in the next chapter.

The Situations

If we capture the context in which interactions occur and the behavior of the people we are simulating, we provide the student with a chance to expose herself to the target culture, to comprehensible input, and to interactions with native speakers. How well we capture the social and physical setting in which something occurs and how well we capture the way people react to utterances determine how much the student can gain from the simulation.

Context (Social/Physical)

The first principle:

(1) Present language in context, as it is used in real life.

Language is always used in context. When our parents tell us that we are going to 'barbecue' for the first time in our lives, the activity is integrated with the fact that we are on vacation from school, that it is warm outside, it is a weekend, we grill meat, it smells all over the place, the sky is blue, and the dog is running around. The word 'barbecue' becomes associated with all these other things. One day, a few weeks later, when the sky is blue and the dog is running around, we might suggest to our parents: "Why don't we barbecue?" The social and physical context helps us organize new information.

The situated learning literature articulates well how the context helps us learn. Brown, Collins, and Duguid (1989), arguing for the importance of teaching knowledge in context, note that "knowledge is situated, being in part a product of the activity, context, and culture in which it is developed and used." Activity, concept, and culture are interdependent, they argue, and learning in context allows us to off-load part of the cognitive task onto the environment. We off-load part of the task by leaving it up to the environment (e.g., a blue sky and the dog running) to give us the cues with which to retrieve a piece of knowledge (e.g., barbecue). The environment provides the student with the conditions for applying and organizing knowledge, by showing how a piece of knowledge is used in a larger context (Collins, 1988).

This means that, in the simulation, I should recreate the social and physical context in which language is used. Since the environment provides the keys to index knowledge in our heads, I must present language in context. For example, if the task is to prepare American students for life in a Japanese university, I must simulate conversations that occur in Japanese universities, with Japanese students, speaking the way they speak in real life. Ideally, I should also expose him to the same physical surroundings, social setting, and tasks that he will face at the university.

Perception (Audiovisual)

(2) Provide visual and aural information.

Most of the contextual information described above is perceived either visually or aurally. In real life we observe the way people walk, the way they gesticulate, and the way they move their lips when they talk. The visual, aural, and contextual cues involved in the interaction all assist in learning.

An example that illustrates the importance of visual and aural cues in learning was given by T. Gallwey during a talk in Chicago (1991). Gallwey suggested the idea of teaching someone how to say the word 'uncoordinated' by saying: "You round your lips and open your mouth until your lips are about half an inch from each other, place the tip of your tongue behind your lower teeth where the teeth meet the gums, lower the middle of your tongue so that the inside of your mouth form a wide chamber, now release a puff of air and vocalize from the middle of your throat for about a third of a second. This will take care of the 'U' in 'uncoordinated'. Now immediately after vocalizing, raise the back of your tongue until it touches the back of the palate, completely obstructing the passage of air. It is this closing of the channel that stops the vocalization. We have now done the entire 'UN' part of the word." The complete description of the word 'uncoordinated' fills about one page. Such a long description makes the student aware of the complexities of uttering the word 'uncoordinated,' but it is not the way people produce speech; there is not enough time to think of all these rules. The behavior can be described verbally, but the production is not governed by verbal rules. Muscles do not use English. Ultimately, oral proficiency is learned through sight and sound.

Also, as shown experimentally, visual information is more effective than other types of memory . An example that illustrates its effectiveness was given by Alan Kay during a conference in Chicago. Kay points out that while flipping TV channels we can recognize a movie we have seen before after seeing only about a couple of seconds of it. And what is more impressive, we can also remember what is going to happen next. Visual information provides powerful cues with which to retrieve information. Whenever applicable, simulations should always provide visual and aural information.

Behavior

(3) People in the simulated environment must react authentically to student's utterances.

Although replicating the context of interactions is important, the value of simulations lies in replicating the behavior of people. The way people act provides feedback that is essential for language learning. An angry tone of voice, for example, tells us that we should change something or maybe apologize. People's reactions to what we say help us adjust our own behavior and correct our knowledge structures.

As Rivers put it, "Interaction involves not just expression of one's own ideas but comprehension of those of others. One listens to others; one responds (directly or indirectly); others listen and respond. The participants work out interpretations of meaning through this interaction, which is always understood in a context, physical or experiential, with nonverbal cues adding aspects of meaning beyond the verbal. All of these factors should be present as students learn to communicate: listening to others, talking with others, negotiating meaning in a shared context. ...communication there must be -- interaction between people who have something to share....Through interaction, students can increase their language store as they listen to or read authentic linguistic material... In interaction, students can use all they possess of the language -- all they have learned or casually absorbed -- in real-life exchanges where expressing their real meaning is important to them. They thus have experience in creating messages from what they hear..." (Rivers, 1987a).

The key to the interactive way of learning languages is the exchange of information, which involves negotiation of meaning. When I say "huh?" during a conversation, my interlocutor realizes that I didn't understand what he said and consequently tries to clarify what he meant. If the simulated people behave like real people, students will be able to adjust their knowledge structures through these negotiations of meaning. Since people's behavior provides important feedback on the student's performance, the reactions of the simulated people should be realistic and perceived as truly dependent on the student's input. Simulated people must behave as they would in real life.

The Student

If we manage to simulate the context of the interactions and the behavior of people, we have captured an important portion of the living-there experience. But, as important as this may be, it is not enough to guarantee learning. It is what the student does that promotes learning. The tasks that the student performs determine how much she'll learn, how motivated she'll be to learn, and how useful the simulation will be to address her real life needs. The level of proficiency she will achieve will depend heavily on her reasons for wanting to learn the language and the learning strategies that she uses. Simulations cannot directly control either of these. However, on one hand, simulations can define (a) the role of the student and (b) the goals that she will pursue in ways that address her reasons and, on the other hand, simulations can provide tools that assist with her learning strategies.

Role

(4) The role that the student plays in the simulation is the role that he/she is practicing for in real life. The student participates actively, not reactively.

A learner who is on his way to London, will be interested in learning to interact with people from London. If he is thinking of working as a mechanic, he will be interested in learning the language used by mechanics in London. Simulations have to place students in roles that are close to the roles they will play in real life.

Language acquisition is heavily influenced by the needs of the individual, and students learn by getting involved in performing tasks that are relevant to them. Active participation forces recall and adaptation of knowledge structures. Students engage in more than just recognition tasks; they learn by building knowledge structures (Papert, 1976, 1987). In the context of actively participating in interactions with other people in London, for instance, the English language furnishes the building blocks, and the student uses the pieces that he knows of the language to construct messages.

In order to involve the student in the simulated situations we must come as close as possible to replicating her role in real life. One way of doing it is by establishing goals that are closely related to her goals in real life.

Goals

(5) Goals must be meaningful to the student. Language is a means to achieving the goal, not the goal itself.

The role that a person plays in a social context is intimately tied with the goals she pursues. And to the extent that the simulation addresses her real life goals she'll be motivated to interact with the simulation.

In a study on motivation, Malone found that "the single feature of computer games that correlated most strongly with preference was whether or not the game had a goal" (1981). According to a study by Morozova, Malone says, motivating goals have the following qualities:

- *Using the skill being taught was a means to achieving the goal, but it was not the goal in itself*
- *The goal was part of an intrinsic fantasy*
- *The goal was one with which readers could identify*

Morozova's study shows that learners acquire skills better when they focus on achieving goals than they do when focusing on learning the skill itself. In language learning, this means to focus on the message not on its structure. In fact, the shift in language instruction away from grammar towards communicative skills is based on the fact that people learn language better when they are concerned with communicating messages instead of with the language itself.

In order to engage the student in the simulated activities, the goals in the simulation must be closely related to the goals of the student in real life. If the simulation helps the student acclimate to an American university, for example, then goals should include things like registering for classes, dropping/adding courses, getting on meal plans, moving into the dorm, and getting a student ID.

Interaction: Student & Situations

In real life, we are not always free to engage in interactions with others. Social norms, limited resources, and personality traits constrain what we can do. Although simulations do not replicate all the advantages of living in the target culture, they offer the advantage of relaxing these constraints that real life imposes on our interactions with others.

Before discussing how interactions should happen in simulations, it is important to note that we cannot control how much time the student spends playing with a simulation. Immersion studies conducted in Canada have shown that the amount of exposure to the target language (for which length of residence is an index) is one of the most important factors in determining the level of linguistic skill achieved (Harley, 1990). Unfortunately, we have very little control over this variable, because the student chooses the amount of time she wants to spend in the simulation. Unlike real life in a foreign country, once the student switches off the computer everything reverts to the source language and culture.

However, since simulations allow us to remove some constraints imposed by real life, simulations offer advantages that may replace quality and intensity for quantity of exposure. First, we can sequence the events in ways that optimize learning (sequencing, discussed next). Second, the student can choose what to do, and even do things repeatedly, which is not always possible in real life (exploration, discussed later). And third, a very important feature, simulated environments are non-threatening, giving students a chance to engage in interactions without the emotional constraints of real life (threat, discussed at the end of this section).

Sequence of events

- (6) Engage in interactions
    - If fail, then show an example
        - Try again
        - If fail, break the task down into smaller parts
    - If succeed, move on

Given that we can simulate the context, the behaviors, and involve the student in roles that are meaningful to her, we need to address one more problem: determine how things should happen in the simulated environment so that students learn.

For teaching to be effective, we must first prepare the student to acquire some new knowledge or skill, then help him acquire and organize the new information in memory, and finally make him use it (Schank, 1991). To prepare students, they must be curious to learn something new. Curiosity arises when a student realizes that his knowledge is incomplete, inconsistent, or unparsimonious (Malone, 81). In other words, they realize they need to learn something when they fail (Schank, 1982, 1991). Since it is not always clear what constitutes a failure, the system must prepare the student by establishing this condition for him, either by intervening in a conversation or by simulating authentic behavior that indicates a failure to communicate (e.g., the simulated person frowns or says "I don't understand you"). So, the first task a computer-based learning environment has is to place the student in a situation that is of interest to him, and wait for an indication that the student needs information he doesn't have. When the student realizes his failure, he will be interested in acquiring new information.

Once the student is ready to acquire new knowledge, we present him with an example. Since failure always occurs in the context of trying to perform some social transaction, the example is always of someone else successfully performing the same social transaction. As described previously, this has to be done in a way that conveys all the necessary cues to help him retain and organize the information (i.e., providing physical context, social context, and audio/aural information). Later, to ensure that the information is correctly assimilated, we force the student to use the knowledge in a situational simulation. Performance in the simulation provides feedback to the student and lets the system detect further failures. If the student fails again, then the task that he is pursuing might be too difficult. If possible, the system should break it down into more manageable portions, and reapply the sequencing described above.

Exploration: Let the student choose

(7) Let the student have control over what to do in the simulated environment.

The structure proposed above should not be taken as a motion for rigidity. One good feature of real life is that we can choose what to do. Unlike classroom situations that force us to study about cats and dogs when the book tells us to do so, real life allows us to walk into a cafe for a chat instead of going to the zoo. This sense of individual choice allows the learner to pursue his interests; simulations should replicate this feature.

In fact, simulated environments can offer more than would be normally available in real life. For example, the student can practice conversations repeatedly. In real life, one does not to go a cafeteria and order food six times just to practice ordering different things. Nobody orders breakfast, lunch, and dinner within a period of five minutes. Simulated environments make it possible for the student to not only repeat experiences but also to go through a large number of them without being constrained by social norms.

Freedom of choice enables individualized instruction. The student gets to do what she wants and when she wants to do it. No longer does she have to comply and pay attention to things that other students are interested in but that are irrelevant for her. I must provide the student with means of controlling the experiences.

Threat

(8) Do not keep scores. Do not evaluate. Do not keep records.

How the student feels, which is largely determined by personality traits, has a strong impact on learning. Introverts and extroverts, for example, relate differently to the demands of the environment. Simulations can work as a social buffer, enabling less socially adventurous learners to engage in conversations with strangers without feeling threatened. Exactly how affect influences language acquisition, however, is the subject of a long debate among second language theorists. A dialog between two theorists, Krashen and

McLaughlin, captures the state of the field when it comes to the role of personality traits and affect in language acquisition.

Krashen (1982, 1985) postulates that individuals learn language through exposure to comprehensible input. However, individuals exposed to the same amount of comprehensible input may acquire different levels of proficiency. Krashen proposes that this difference is due to an 'affective filter.' The learner must be 'open' to the input, and the affective filter is a mental block that prevents all input from being processed. This filter is usually 'up' when the learner is unmotivated, anxious, defensive, or lacking self-esteem; the filter is 'down' when the learner is not afraid of failing and he feels that he belongs in the target group.

McLaughlin (1987), on the other hand, argues that the problem with Krashen's affective filter hypothesis is that "there is no research evidence to support a causal relationship between these personality variables [self-consciousness, vulnerability, and insecurity] and language learning. Indeed, research on individual differences in second language learning has proven to be a methodological Armageddon. It is extremely difficult to show any relationship between personality factors and language learning." McLaughlin does not, however, deny that affect seems to play an important role. He uses the lack of supporting evidence to argue against Krashen's 'affective filter hypothesis' but acknowledges that "most researchers in the field of second-language acquisition would admit that affective variables play a critical role."

The consensus seems to be that affect does play a critical role. As Seliger puts it "Since language is used in social exchanges, the feelings, attitudes, and motivations of learners in relation to the target language itself, to the speakers of the language, and to the culture will affect how learners respond to the input to which they are exposed. In other words, these affective variables will determine the rate and degree of second language learning" (Bebee, 1988). Most researchers believe that extroverted, self-confident people are better learners. People with these characteristics are usually more willing, or 'open,' to trying things out -- they are less afraid of relating to other people. Consequently, they engage in interaction more often than shy people, who lack self-esteem.

Simulated environments are non-threatening. The nature of the instrument, simulations, helps bring 'high filter' individuals to feel at ease and comfortable playing with others. Regardless of how realistic our model of the world turns out to be, the student knows that the simulated agent's reactions need not be taken personally. The only case in which a simulation can become threatening is if it is used to evaluate the learner or keep records of interactions. We should avoid this. These negative aspects of real life need not be transferred into the simulation. The reason the student plays with the system is that she wants to interact with foreigners in a foreign language; the motivation is intrinsic. It is up to the student to choose what to focus on, what she thinks is important for her to know, and so on. Imposing external punishment or rewards on the student only lowers intrinsic motivation (Malone, 1981; Lepper & Greene, 1978). We should not keep information that can be used to evaluate the student.

Resources

(9) Provide tools for addressing communication problems (e.g., dictionary, transcript of conversations, etc.)

(10) Provide tools for analysis and reflection (e.g., recorder)

The only remaining elements that we need to capture in the simulated environment are resources, tools that provide cognitive assistance. In real life, a number of resources facilitate or assist in learning. Books, dictionaries, and people help us cope with the target culture. In simulations, a number of new tools such as transcripts of conversations, subtitles, and translations can be added. Also tools that help students reflect and practice language patterns, such as tape recorders, should also be provided. Simulated environments should include these types of tools. I discuss next, the most useful of these resources, a person who's willing to help, a tutor.

Tutor

(11) Help student perform social transactions

Simply being thrown in an environment where the language is used is insufficient for learning to occur. Immersion has its limitations. Universal, non-verbal means of communication may help us survive, but to learn a language we need to properly socialize, and for that we need help. The way we learn when we are young is by having our parents serve as mentors in the process of acquiring language. They are expert users of the tool and we learn from them by observing, trying with, and receiving help from them. They show us how sounds associate with things, actions, and abstractions. We test our hypothesis on them, and observe the effect of our utterances. They correct us when needed.

Apprenticeship, the most popular model of instruction until the 19th century (Collins, Brown, Newman, 1989), is based on the idea that skills used to be learned by working with an expert. The apprentice sits at the feet of the master, and learns through a process of observation, practice, and coaching. The student observes the master performing a task, tries to do it himself, and receives help from the master. Both the master and the task itself provide the necessary feedback for the student to monitor his performance.

In simulations, we need to give the student someone who can assist him in these social transactions -- a tutor. Next, I define three characteristics of the help provided.

(12) Do not correct grammar.

There is no evidence that correcting grammar is helpful or necessary in language acquisition (Terrell, 1977). Learners in natural settings acquire grammar even if incorrect speech is accepted in the initial stages of language acquisition. Moreover, it is almost impossible for a beginner to be able to carry on a conversation without errors. Therefore, the tutor should not focus on grammar. It should not correct grammar.

(13) Provide constructive feedback

Malone (1981) argues that to be educational, feedback should be constructive. "In other words, the feedback should not just reveal to learners that their knowledge is incomplete, inconsistent, or unparsimonious, but should help them see how to change their knowledge to become more complete, consistent, or parsimonious (p. 364). Although the tutor should not intervene at the grammatical level, it should intervene when the student is not saying the right thing at the right time. If the student asks for something without first greeting the other person, for example, the tutor should intervene and suggest that the student greet first. Other types of intervention can occur when aiding the student in making decisions that give him control over the environment (e.g., Are you hungry? Do you want to meet someone?) The student's choice will determine the course of his experiences in the simulated environment.

(14) Provide means for the student to interact with the tutor

It is not always possible for the tutor to detect what kinds of problems the student is having, a problem that has been consuming much of researchers' time through the creation of student models. Student models, or inferences based on student's input, enable us to detect cognitive problems, the interactions themselves will provide the necessary information to infer the state of the student. But to individualize instruction, a tutor needs both cognitive and affective feedback. If a task is too easy, the student will be bored. There is no way for the system, based solely on the input, to determine whether the student managed to accomplish the task because he is over qualified for it or if his knowledge is just about enough at that level of difficulty. In real life, the student helps by making faces, by showing boredom on his face. In the simulation, we need alternative ways of detecting boredom. This means that the tutor must handle common problems students face, and the system must provide the student with means of communicating the most common states to the tutor (e.g., bored, not knowing what to do, not knowing what to say).

Problems: Limitations of technology

Although simulations are ideal for language learning, ideal simulations are not yet feasible. Technology, as it currently stands, is still limited, and a number of problems have to be addressed before we can begin to build simulations. The major problem has to do with communicating with a computer, a problem that is best explained by an example.

In a real life French bakery, if you do not understand the attendant, all you have to do is frown and he will probably realize that you didn't comprehend what he said. He'll rephrase what he said or try some other way of communicating with you. In a simulated French bakery, you frown and the computer stays put. Conveying your confusion to a simulated agent in a computer is impossible through facial expressions or through gestures. Computers can't see, and they are doing poorly at listening. Humans communicate through facial expressions, body language, intonation, and other extralinguistic cues, and much of what we learn depends on the quality of feedback that we receive from the environment. Now, if the environment (simulated) is such that it cannot respond well to our behavior, how much can we interact with and learn from it?

The main problem with current technology for implementing simulations is that the channel of communication is very narrow going from the user to the computer. The student is able to absorb a large amount of information coming from the environment, through video and audio, but the computer has very narrow channels of communication to get information from the student. The question is whether the fact that computers do not process facial expression, speech, tone of voice, body language, gesticulation, focus of attention, and blankness in the eyes mean that we can't simulate interactions with people. If the agents in the environment can't detect the student's reactions, can they interact at all with the student?

Conceptual Feedback

> (15) Provide means for the student to express important cognitive and affective states to the simulated people and to the tutor. Provide means of communicating common states (e.g., Huh?)

Here's one solution. In a paper about reflection, Collins & Brown (1988) describe different levels of feedback that can be given to a student who is learning a swing in tennis. At the level with highest physical fidelity, i.e., replay, the student performs a forehand, the system video tapes him, and plays it back side by side with a video of an expert performing the same swing. The student has a very accurate view of what he himself is doing. At another level, i.e., abstracted replay, the computer records the critical features of the motion, a reflective material is taped to critical points such as the racquet and body joints, so that the student is able to focus on the important aspects of the swing and compare it to the expert's motion at a different level of abstraction -- the student is not distracted by the way he looked in comparison to the pro. This second type of feedback lacks physical fidelity, but is has conceptual fidelity. It extracts the information that is important in that context.

Now, turn this around. We may be unable to have physical fidelity when conveying information to the simulated environment, the simulated agents won't understand our crying. But we can have conceptual fidelity of feedback -- as long as the environment allows us to express what we need to express (e.g., through an "I am crying" button). Until computers become able to process body language and tears, we can resort to conceptual feedback for the purposes of interaction. The next question is what kind of conceptual feedback?

Lepper and Chabay (1985) argue that truly personalized instruction must be individualized along motivational as well as cognitive dimensions. This means that when interacting with a student, as when teaching mathematics, we need to know more than whether the student is getting the correct answer or not. We need to know if he is bored, we need to know whether the whole thing is above his head or not. Perhaps the student would prefer to be working on something else. Interactivity means having the student's states affect the environment. Again, computers are incapable of analyzing facial expressions and detecting if information seems to be going way over the user's head. However, we can use the same kinds of conceptual feedback mentioned above. If the student feels something is too hard, he can express this by using buttons.

The system must provide means for the student to convey these states to the computer. Since the system can't detect it from the input all the time, the student communicates it by pushing a button or some other active means. As discussed earlier, the student must be able to communicate both cognitive and affective states to the simulated people.

Natural Language

(16) Use natural language interface (assist/facilitate spelling)

The other main problem for language learning purposes is the limited speech recognition capability of current technology. Development of oral proficiency is heavily dependent on feedback from listeners, and in the 'natural' setting almost all interactions are based on oral communication. To provide the same amount of feedback and style of interactivity, language learning systems will have to incorporate speech recognition. This is a limitation of current technology that we need not rationalize. It would certainly be preferable to have full blown speech recognition capabilities.

So, the question is: Can current technology be useful in language learning? What can we do with limited speech recognition capability? Or, assuming that it is possible to interface using typed input in natural language (e.g., in the target language), can we use it effectively for language learning?

Oral proficiency develops from a combination of (1) input, (2) feedback, and (3) reflection. We hear a speech pattern a number of times until we grasp its meaning and sound. At one point, we try to use it in some context and receive feedback that means either that we didn't pronounce it correctly or that it didn't make sense in the context in which we used it -- or both. We then reflect upon the experience and make necessary adjustments. The part of the process that current technology does not allow us to provide is feedback on pronunciation. Other than that, computer-based environments can give the student a wealth of input (e.g. videos of people using language and interacting with each other in many different real life contexts); using a natural language interface (i.e., typed input) the system can provide feedback indicating that something said did not make sense in the specific context; and finally computers can provide innovative tools to help the student reflect on his oral performance by giving feedback (e.g., recording the student's speech and playing back, showing graphical representations of speech, i.e., spectrographs, and intonation, i.e., pitch tracking).

In order to determine in what conditions either limited speech recognition capabilities or natural language interface would be useful, we need to determine under what conditions feedback on pronunciation plays no significant role in language learning. There are two situations in which this is true.

The silent period

Research shows that in natural settings, learners go through a 'silent period,' during which they listen very carefully to input and produce very little output (Richards & Rodgers, 1986). During this period, learners interact with others by using simple expressions such as 'yes,' 'no,' and pointing. In the long run, people who use this learning strategy acquire better oral proficiency than people who engage in speech production early on. This phenomenon seems to be attributable to the fact that such learners develop more acute aural discrimination before they engage in production. Current speech recognition technology can easily support a vocabulary of a few words (e.g., 'yes, 'no,' 'what?', 'no speak English'). At ILS we tested a system, brand x from company ?, robust enough to implement a simulation that would be perfectly suited to those students who are in this phase of the acquisition process. In other words, simulated environments can be very effective for students in the silent-period.

The Classroom Proficient

A certain class of people has an interesting skill. They have what is called 'context-reduced proficiency' (Harley, 1990). Having this skill, also referred to as 'academic proficiency,' means that they are able do well in school tests, and perform well in the classroom -- they are the product of the typical classroom language instruction. These people know how to read and write, they can conjugate verbs, and perform complex syntactical analyses. However, very often, they do not know how to interact with native speakers in real life situations. Since they can write in the target language, natural language can be a useful way of providing these individuals with the exposure to native speakers in real life situations, let them interact with them, and receive feedback on the appropriateness of their sentences. The simulated environment will not provide feedback on pronunciation (except through the tools), but all other aspects of language usage can be exercised through the computer-based system. If they are not concerned with acquiring native-like pronunciation, learners who are academically proficient benefit from playing with a simulation.

Conclusion

Table 6 summarizes the desirable characteristics discussed in this chapter.

Table 6

Desirable characteristics of a computer-based language learning environments

Variable	Desirable Characteristics	Shape
Context/Perception	• (1) Present language in context, as it is used in real life. • (2) Provide visual and aural information.	**Situations**
Behavior/Feedback	• (3) People in the simulated environment must react authentically to student's utterances.	
Role	• (4) The role that the student plays in the simulation is the role that he/she is practicing for in real life. The student participates actively, not reactively.	**Student**
Goals	• (5) Goals must be meaningful to the student. Language is a means to achieving the goal, not the goal itself.	
Sequence	• (6) Engage in interactions • If fail, then show an example • Try again • If fail, break the task down into smaller parts • If succeed, move on	**Interactions**
Exploration	• (7) Let the student have control over what to do in the simulated environment.	
Threat	• (8) Do not keep scores. Do not evaluate. Do not keep records.	
Tools & Tutor	• (9) Provide tools for addressing communication problems (e.g., dictionary, transcript of conversations, etc.). • (10) Provide tools for analysis and reflection (e.g., recorder). • (11) Help student perform social transactions. • (12) Do not correct grammar. • (13) Provide constructive feedback. • (14) Provide means for the student to interact with the tutor.	**Resources**
Conceptual Feedback	• (15) Provide means for the student to express important cognitive and affective states to the simulated people and to the tutor. Provide means of communicating common states (e.g. Huh?).	**Problems**
Natural Language	• (16) Use natural language interface (assist/facilitate spelling).	

Chapter 5

Dustin - A Language Learning Environment

From Principles to Practice

Based on the principles outlined in the last chapter, I designed an architecture for computer-based language learning systems which I call Role-Playing in Social Simulations (RPSS). When using RPSS systems, students practice for real-life performance by rehearsing in a simulated environment. In this simulated environment, they interact with native speakers while playing the roles for which they are preparing themselves in real life. Students receive guidance from a tutor, who determines the tasks that they have to accomplish and provides help when needed. They have access to a number of supporting tools and can control how they navigate in a space of tasks and examples.

For the sake of clarity, instead of describing features of the RPSS architecture out of context, I will explain them in the context of a session with an implementation called Dustin. Dustin is a version of this architecture that was implemented to prepare foreign employees of a large consulting firm for their first visit to the firm's training center in the United States. I will describe Dustin and show how it implements the principles outlined in the previous chapter. Before I start, however, a few words about the components of simulations and the situation being simulated will help place Dustin in context. This context will also help organize the discussion about the RPSS architecture in the following chapter.

The Components

Any simulation can be divided into three major components: (1) what it simulates (real-life situation), (2) how it simulates it (mechanism), and (3) how it looks to the student (interface). Figure 12 depicts these three components and how they relate to

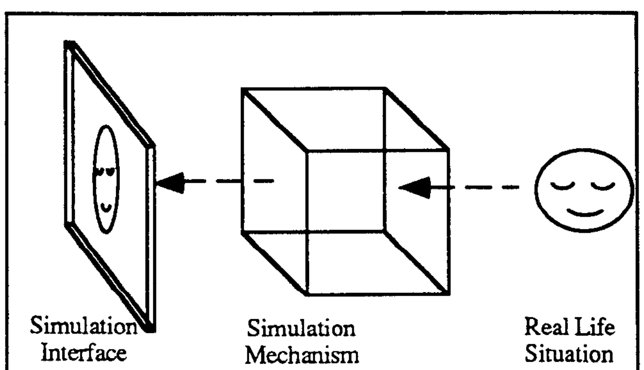


Figure 12. Simulations can be divided into three major components.

each other. In the description that follows, the (1) interface and the (3) real life situation will become apparent to the reader. For now, the (2) mechanism, which I describe in chapter 7, will remain transparent.

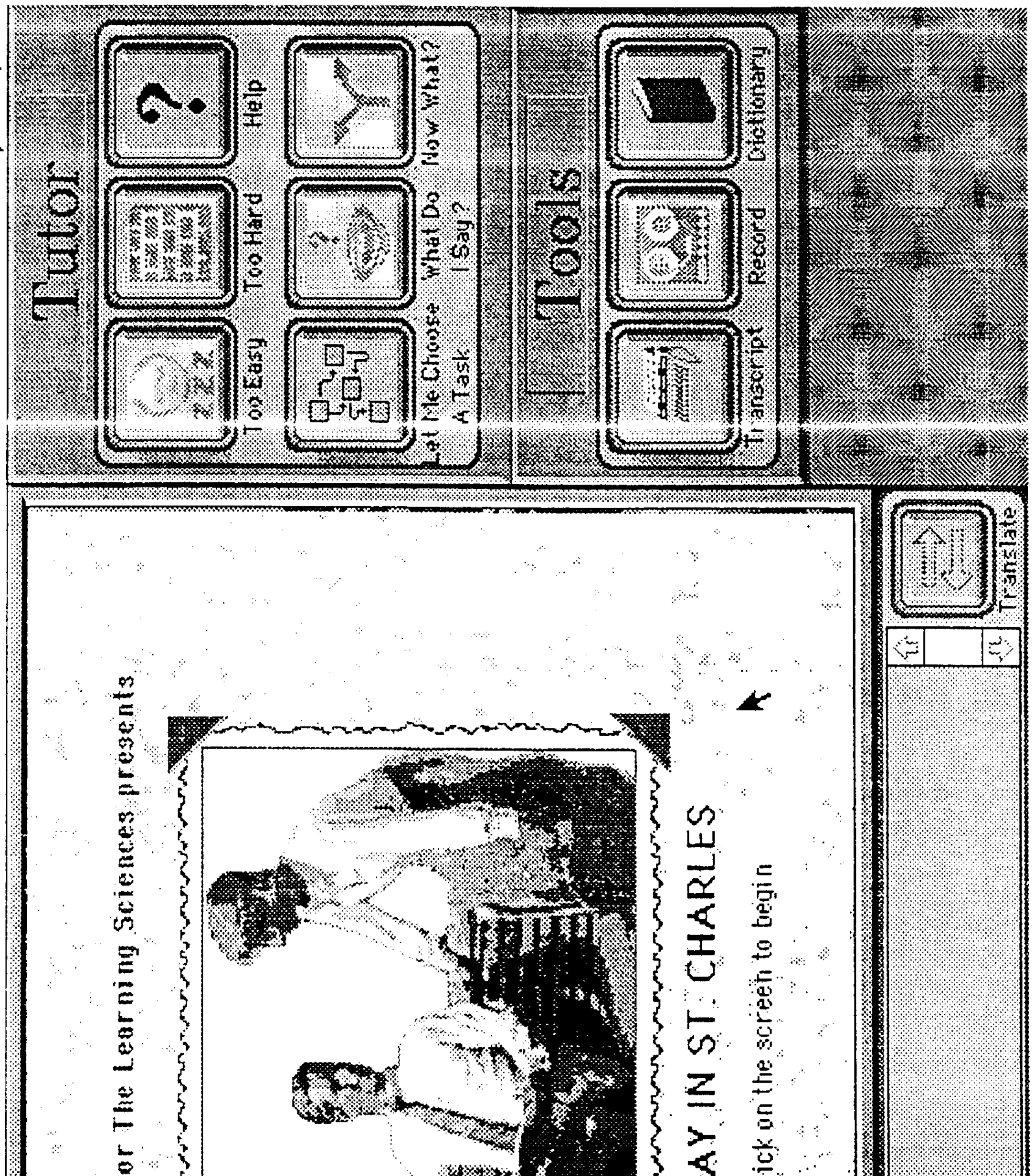
The Real-Life Situation

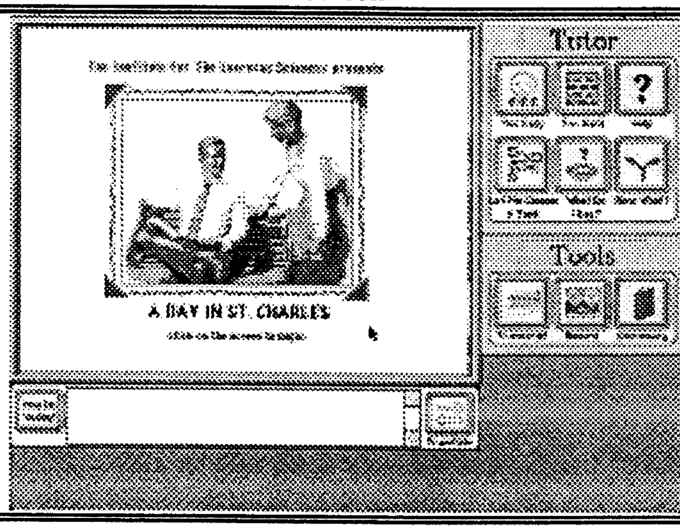
Andersen Consulting employees from around the world come to the firm's educational center in St. Charles, IL for training. Many of them, those who are in the United States for the first time, face tremendous difficulties with the language and culture when they get there. Typically, these newcomers have had training in English as a second language, but have had little or no exposure to native speakers. They can read and write fairly well, and usually perform well in tests. However, once in the United States, they have difficulties performing simple tasks such as going through immigration and checking into a hotel, difficulties that derive mostly from their lack of exposure to the target environment.

Dustin is a simulation of this target environment. It simulates the situations that these newcomers confront during their first day in the United States. The simulation begins at O'Hare, where they have to go through immigration and find transportation to St. Charles, and continues through a number of activities during a day in St. Charles, where they have to check in, meet their roommates, get food, go to classes, and engage in tasks that trainees usually face in St. Charles.

Dustin

Newcomers use Dustin on a one-on-one basis before coming to the United States. The version described here was developed for, and installed in, the firm's office in Madrid. To place yourself in the proper context, as I describe a session with Dustin, imagine yourself in Madrid, as an employee who has just been hired by the firm. You know English from a few years of English classes in high school, have passed a test in English proficiency (grammar/vocabulary), and are on your way to St. Charles for the first time. You are sitting in your office in Madrid.

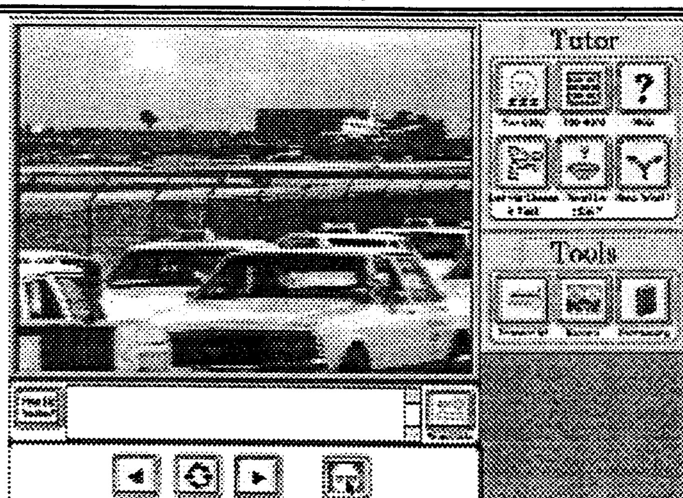

Dustin View Preferences
The Institute For The Learning Sciences presents
A DAY IN ST. CHARLES
click on the screen to begin
Tutor
Too Easy
Too Hard
Help
Let Me Choose A Task
What Do I Say?
Now What?
Tools
Transcript
Record
Dictionary
Translate
What Did You Say?

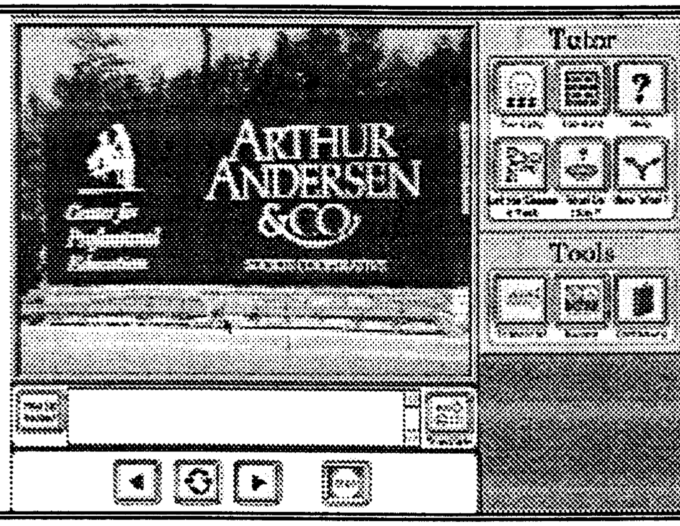
System	Screen	Student
		Student clicks on the screen

On the left column are system's actions; on the right column are student's actions. The picture in the center is a snapshot of the screen the student sees when using Dustin. These boxes should be read from left to right; the sound transcribed on the left column (system's action) is played before the student's action shown on the right column.

At this point, the student clicks on the central screen for the system to start.

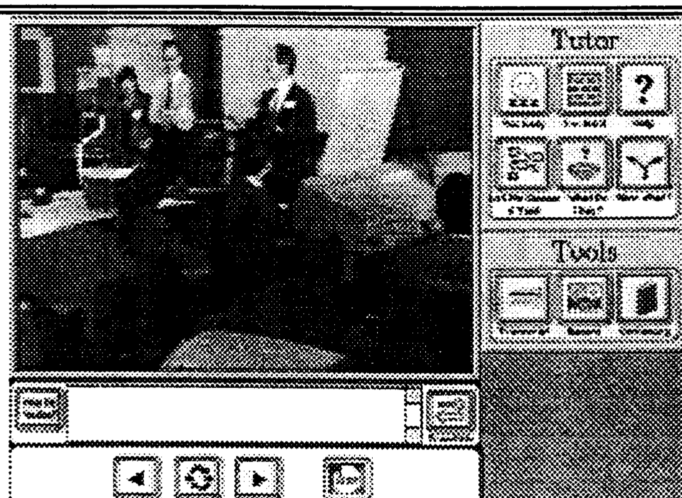
As each new feature of Dustin appears, I will indicate the design principle that the feature addresses. Principles appear in boldface and are numbered according to their numbering in the previous chapter.

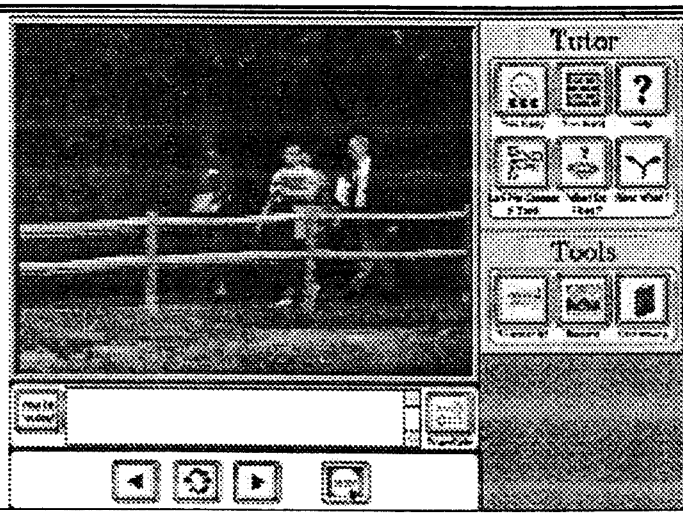
System	Screen	Student
Welcome to the United States. You have just arrived in Chicago, and are on your way to the Arthur Andersen Center for Professional Education in St. Charles.		

System	Screen	Student
This training center, nestled along the Fox river just West of Chicago, will be the center of operations during your stay.		

(2) Provide visual and aural information

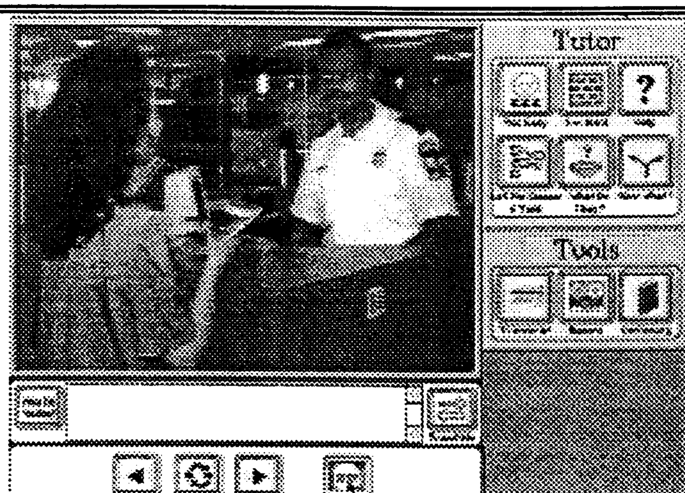
The window on the upper left corner (showing the Arthur Andersen sign) is the window into the simulated environment. It is here that the simulation provides the visual information, accompanied by audio.The physical surroundings, extra-linguistic cues, and behaviors are captured and conveyed through this window.

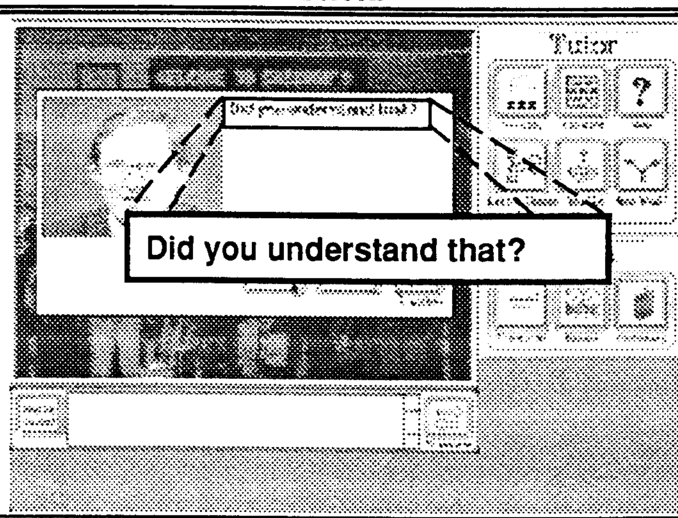
System	Screen	Student
After you check in, you'll have a chance to meet new people, learn new skills, and become better acquainted with the St. Charles environment.		

System	Screen	Student
You'll meet Maria Almeida and John Harrison, two Andersen employees who will lead you through various experiences if you need help.		

(1) Present language in context, as it is used in real life.

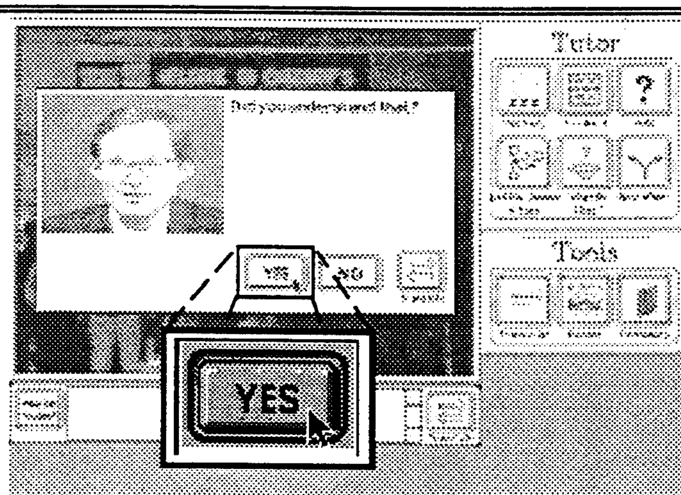
The system places the student in the same context she will face when arriving in the United States. The physical surrounds and the people she will see in the simulation are those she will see in real life. Every situation involving the use of language mirrors how it is used in real life. Soon she will be introduced to her first task: go through immigration at O'Hare Airport.

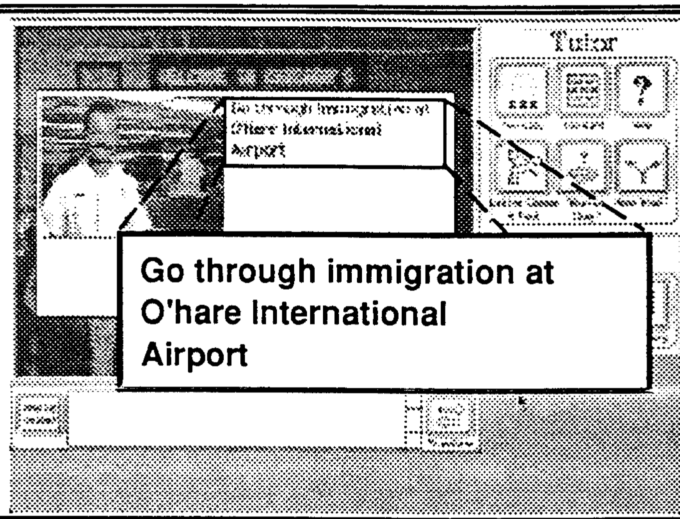
System	Screen	Student
But first, you must go through immigration at O'Hare International Airport.		

System	Screen	Student
Tutor Intervenes "Did you understand that?"		

Users in Spain would normally set the interface to Spanish. The interface language is an option in the menu bar through which the user can change the labels of the buttons and the introduction to her native language. The rest of the simulation is always in English, regardless of the choice of interface language. When the interface is set for Spanish, only the introduction and the buttons are in Spanish; simulated agents always speak in English.

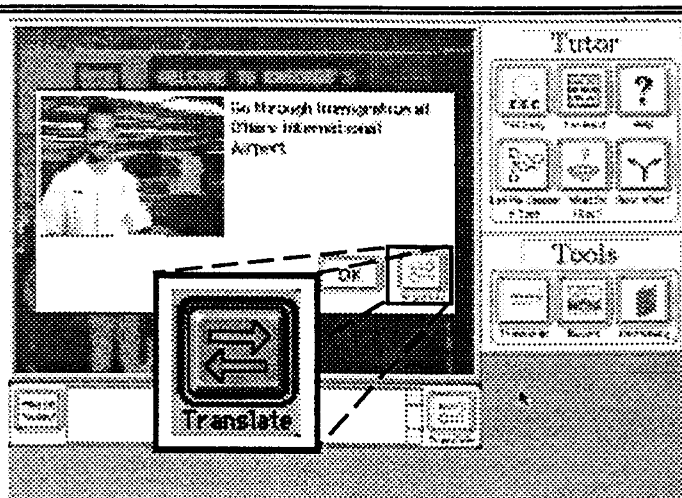
Tutor interventions always appear in the middle of the screen. The tutor will appear often to direct the student, to provide hints, to offer choices, and to give feedback.

System	Screen	Student
Note: If the student says NO, the system plays the Introduction in Spanish.		**Student clicks on YES button**

System	Screen	Student
Go through immigration at O'Hare International Airport.		

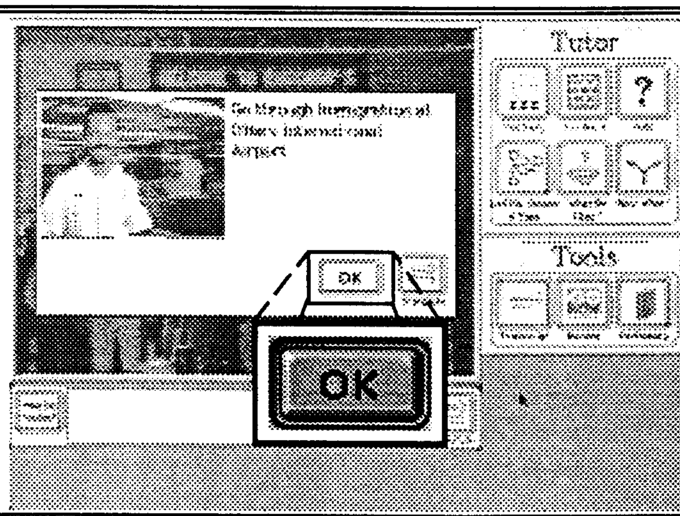
(4) The role that the student plays in the simulation is the role that he/she is practicing for in real life. The student participates actively, not reactively.

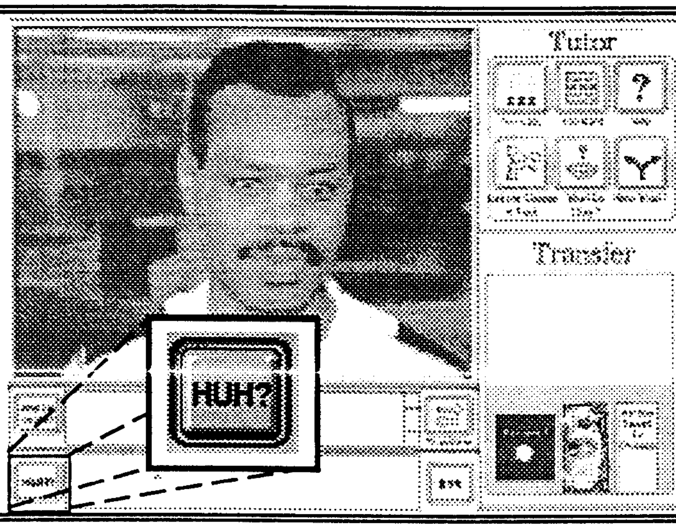
The tutor places the student in the proper context by assigning her tasks, tasks that always involve interacting with someone. The goals in these interactions are very well defined, and basically the same goals that she will pursue when she comes to the United States for training. Going through immigration is the first task -- the student has to interact with the immigration's agent in order to enter the country.

System	Screen	Student
		Clicks on TRANSLATE button

System	Screen	Student
		

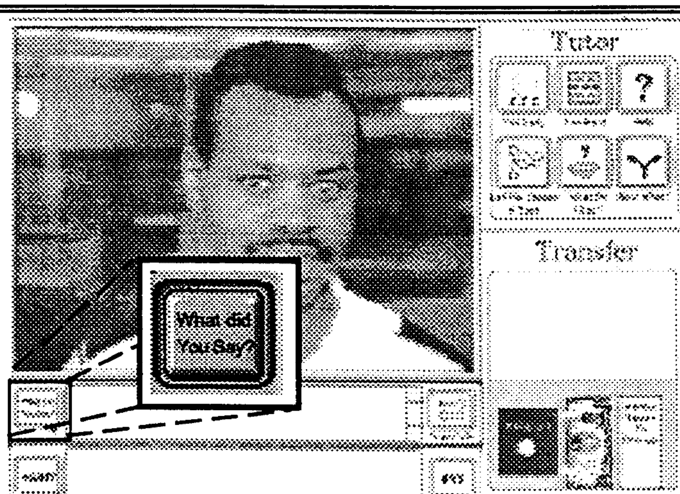
Translations are always available. The student is not allowed to 'turn on' translations, since we want to encourage reliance on the target language, but they are always, at most, a couple of steps away.

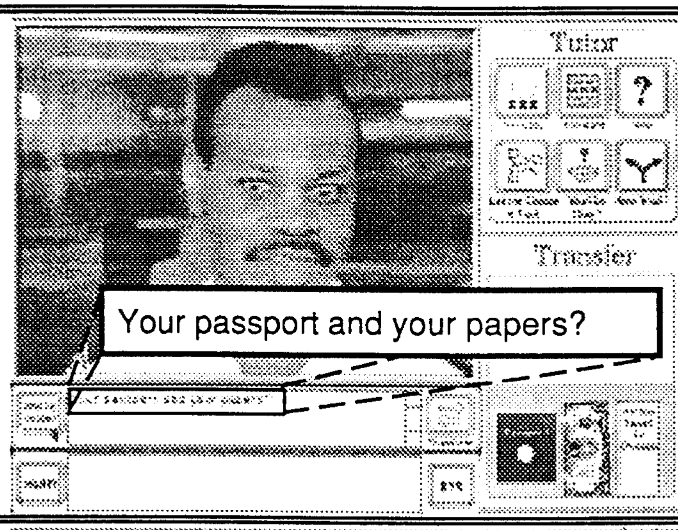
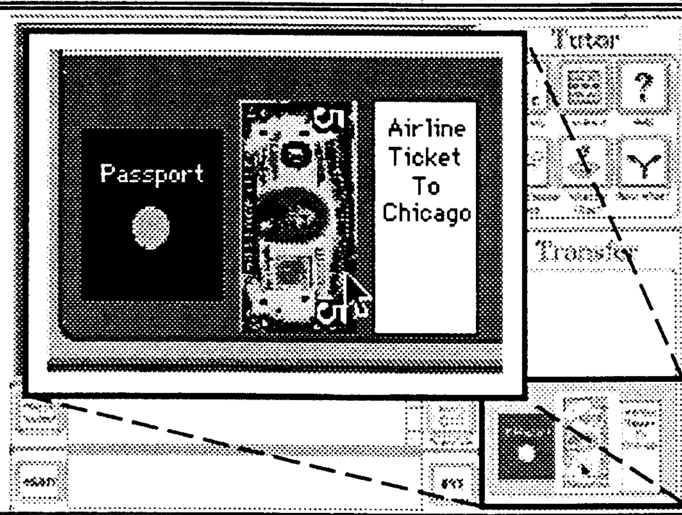
System	Screen	Student
		OK

System	Screen	Student
Can I see your passport and papers please?		**HUH?**

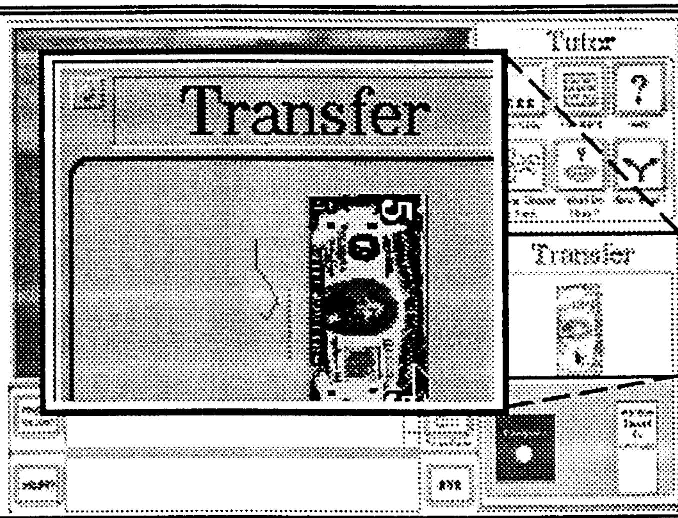
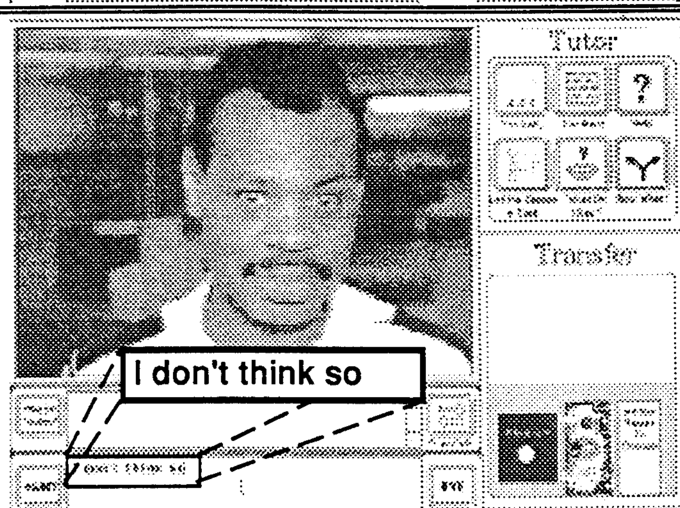
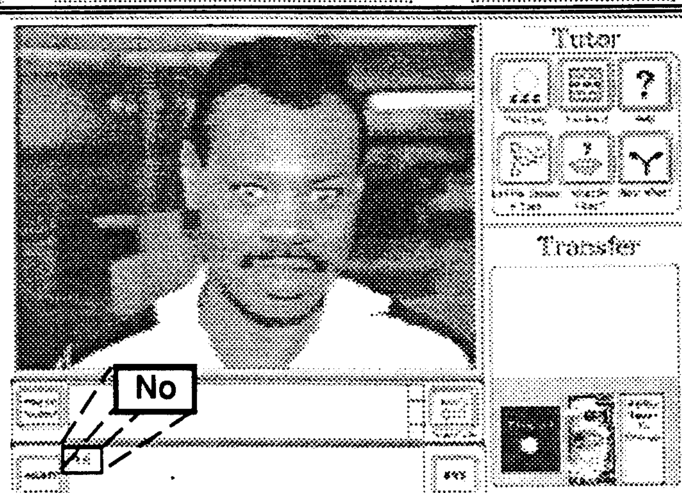
(15) Provide means for the student to express important cognitive and affective states to the simulated people and to the tutor. Provide means of communicating common states (e.g. Huh?)

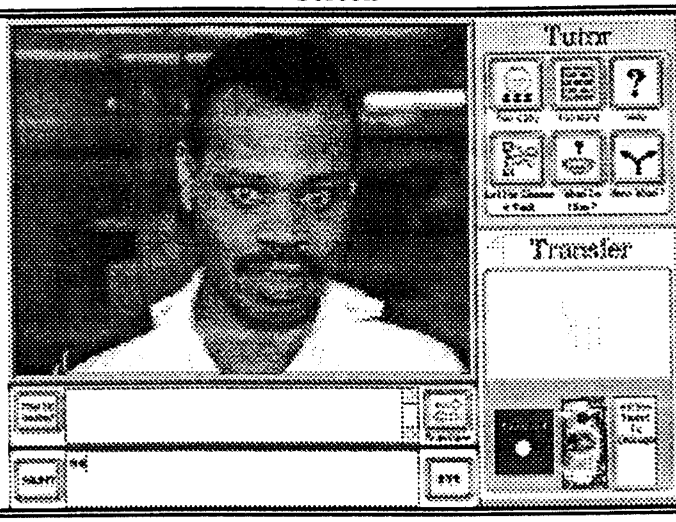
The student expresses confusion through the HUH? button. One of the problems discussed in the last chapter was that in real life we can just frown to let people know we are confused. In simulations, the student has to translate her frowning into an input that is understandable to the computer. The solution adopted in Dustin is to provide buttons through which the student communicates these states. When interacting with simulated people, the student can either use the HUH? button to express lack of understanding, like frowning, or the BYE button, which indicates that as far as the student is concerned, she's done with the task. Other buttons allow the student to communicate cognitive (e.g., Now What?) or affective states (e.g., Too Easy -> I'm bored).

System	Screen	Student
Your passport and your papers?		**What did you say?**

System	Screen	Student
Note: Another tool: the text for the sound just heard appears. Translation is also available. Tools are discussed later.		
Note: The student can transfer objects to the agent.		

Although most interactions are language-based, some are best represented by actions such as handing over a passport. In Dustin, whenever the student has to pay for something or transfer an item, such as the passport, he has the option of physically dragging the object to the simulated agent, instead of saying "here it is," for example. This type of tool is very effective in creating TPR (Total Physical Response) systems in which students react to input by performing actions. Object manipulation in computer-based environments can cover a wide range or interactions and serve as a metaphor for a wide number of tasks. Simulations of this kind might prove useful for beginners who are developing comprehension skills in a TPR style of apprenticeship.

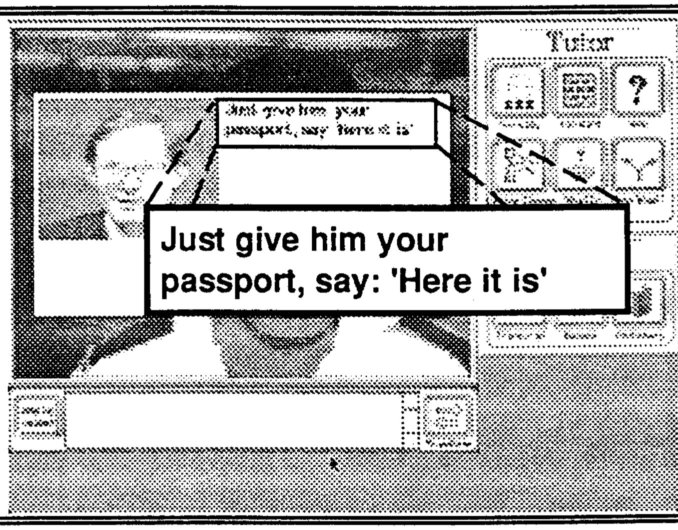
System	Screen	Student
		Drags a five-dollar bill.
No, just your passport will be fine now.		Student types: "I don't think so."
I need your passport!		"No"

System	Screen	Student
I think we are having a little problem here. Maybe we need to have you sit down and talk to someone, ok? (getting annoyed)		

(3) People in the simulated environment must react authentically to student's utterances.

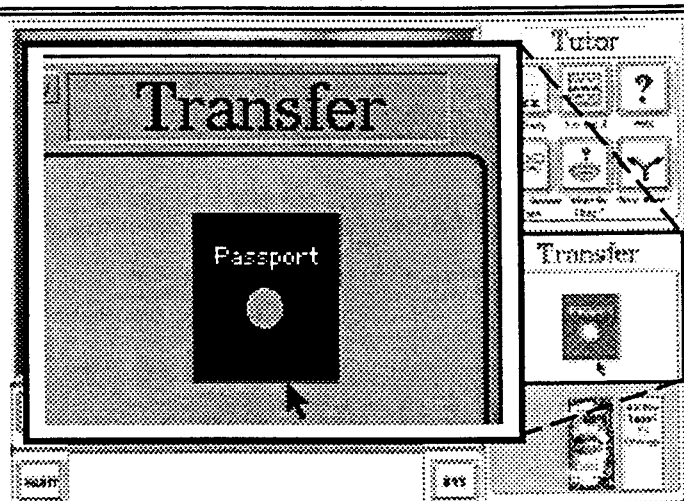
The student was being uncooperative in this task, and the simulated agent reacted as he would have reacted in real life. First, the student tried to give him money, then refused to give him the passport, and later persisted in refusing. Immigration agents are always polite and usually refer problem-passengers to their supervisors -- it is not unusual that passengers who have had a few too many drinks behave innapropriately -- but, like any human being, they react emotionally to annoying questions, and become impatient if the conversation is not progressing satisfactorily. Dustin captures such reactions and uses them accordingly in the simulated interactions.

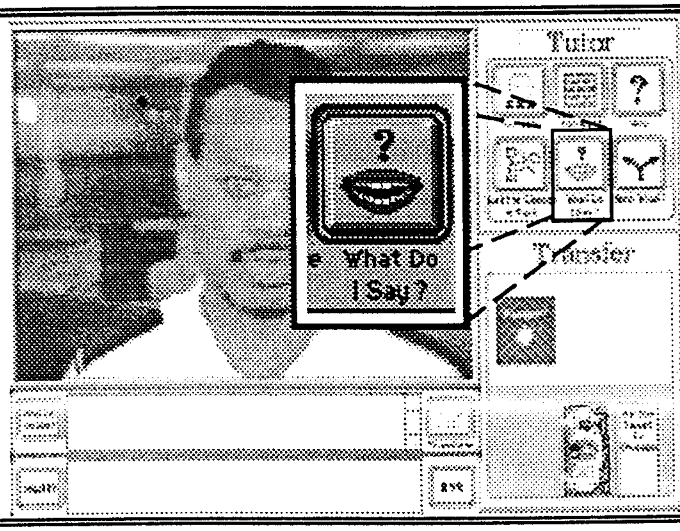
One reason this is important is that, when interacting with others, people often engage in non-standard behavior as a way of determining what is acceptable and what is not. During one of our test sessions, with 25 foreign employees, misbehaving was very common among people playing with the simulation. They refused to do things, they tried to pick up women in the simulation, and often tried to push people to their limits. One reason they do that is that they want to test the limits of the system, but another, and more interesting one, is that simply knowing what is right is not enough. People like to test the limits of acceptable conduct as a way of developing adequate models of how they should interact. Simulations have to account for this type of behavior.

System	Screen	Student
Just give him your passport, say: "Here it is."		OK

(11) Help student perform social transactions.

The student's uncooperativeness is getting her into trouble. The tutor, one of the most important resources available to the student, helps her refocus on the task at hand. The tutor will reappear a number of times playing different roles. In this case, it is simply guiding the student so that she accomplishes the task that she was given (i.e., go through immigration).

System	Screen	Student
I need your passport.		**Drags the passport.**

System	Screen	Student
What is the purpose of your visit today?		**What do I say?**

(14) Provide means for the student to interact with the tutor.

The student interacts with the tutor to ask for help. This button, What do I say?, allows the student to communicate a state that would otherwise have to be inferred from the actions of the student, an inference that is not always feasible. 'What do I Say?' is like shrugging one's shoulder with a question mark in one's face: "What am I supposed to say now?" The tutor obliges. Other buttons serve similar purposes.

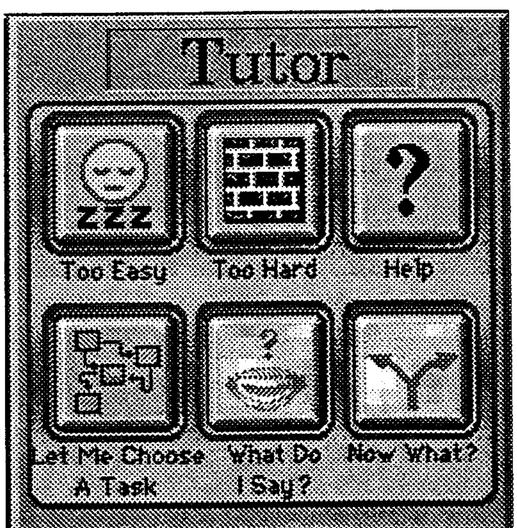
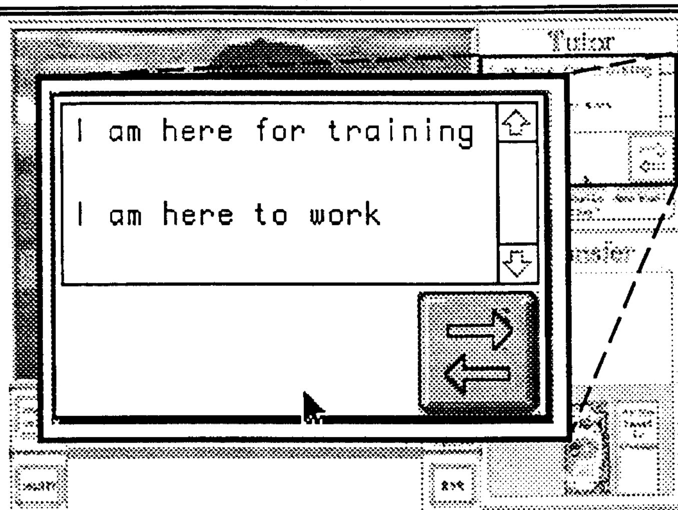
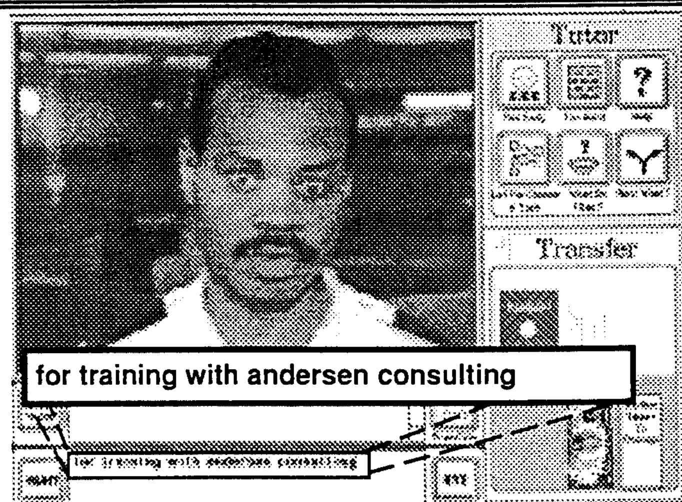


Figure 13. Tutor buttons in Dustin

Too Easy/Too Hard: The system selects the next task depending on the circumstances under which the student presses these buttons. It may be a simpler task, watching an example of the current task, a hint, or passing control over to the student.

What do I say?: The tutor suggests things to say -- tutor suggestions may include inappropriate utterances.

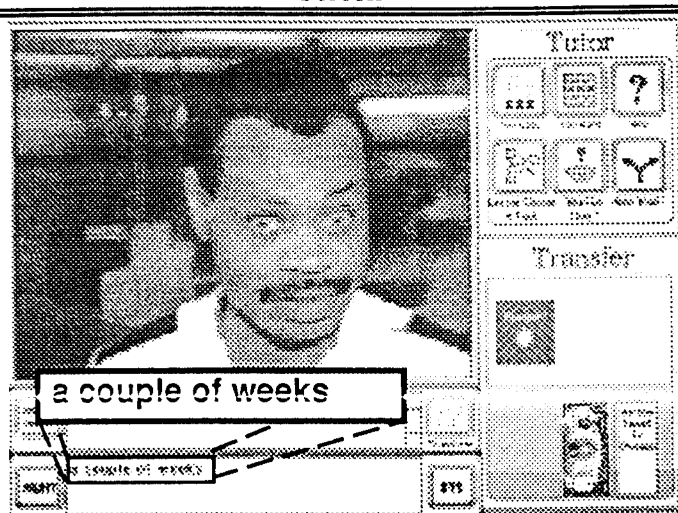
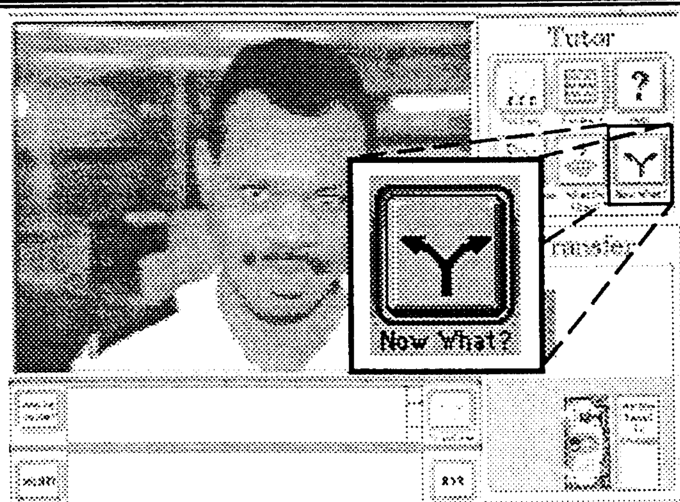
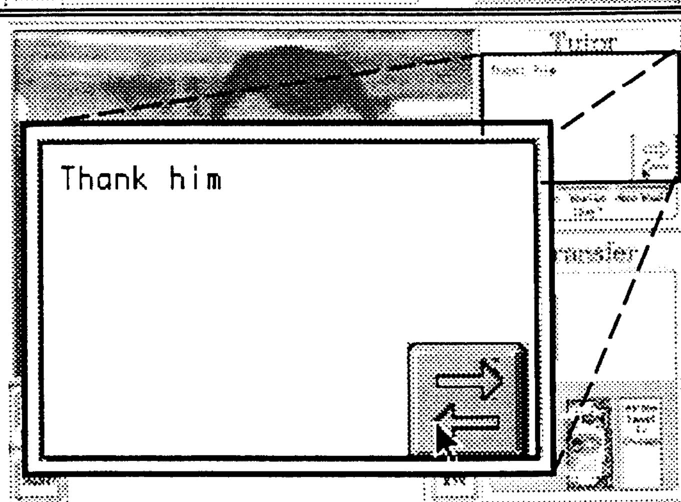
Now What?: Instead of telling the student how to achieve a goal, it explains to the student what is expected of her. It serves as a reminder of what the current task is.

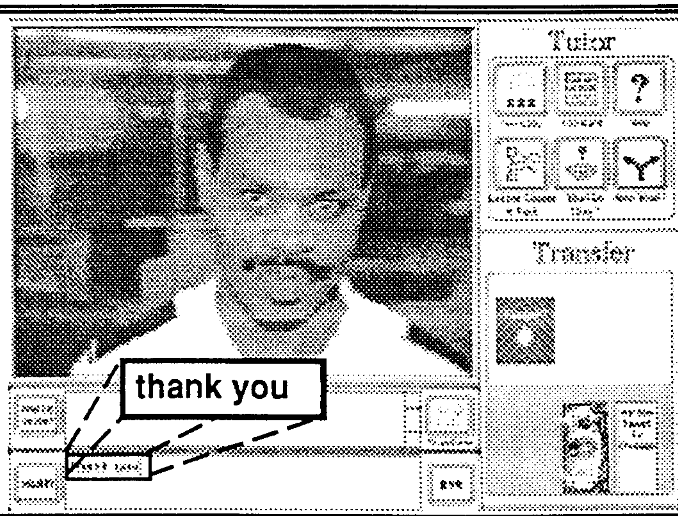
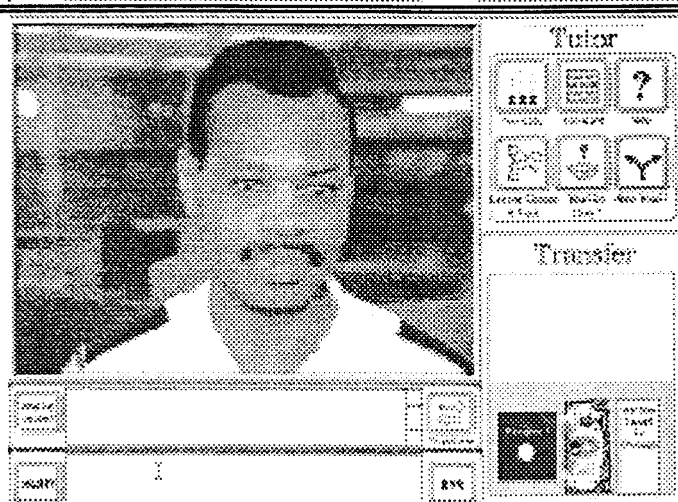
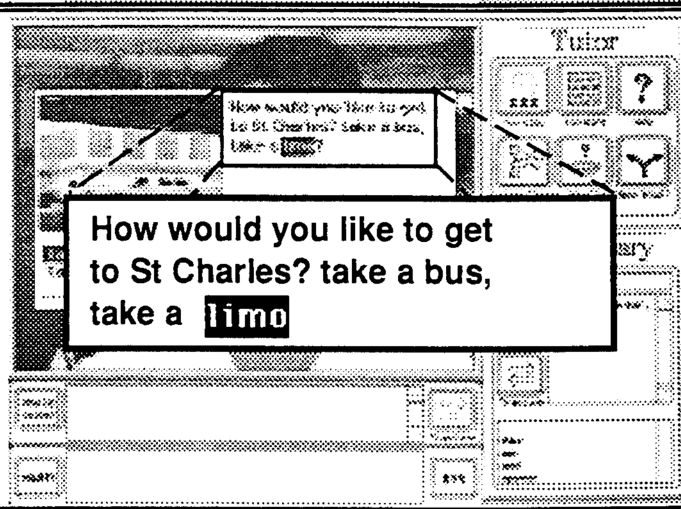
System	Screen	Student
		
		"For training with Andersen Consulting."

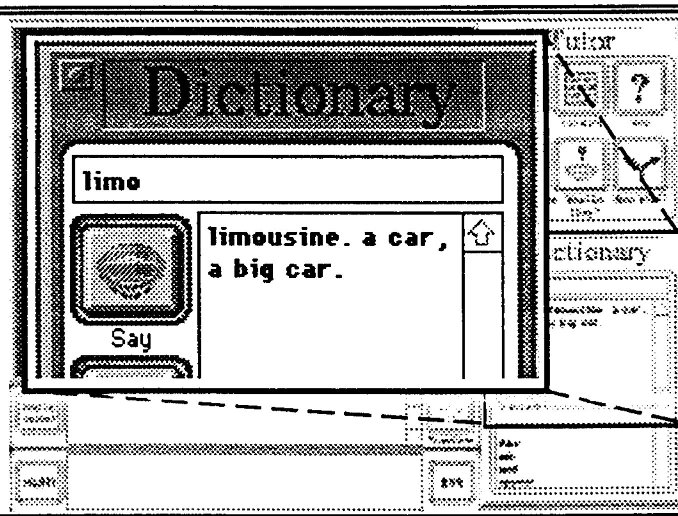
(16) Use natural language interface (assist/facilitate spelling).

In Dustin, the major means of interacting with simulated agents is through typed input in natural language -- the student types in the lower box on the screen. The input is checked for spelling -- a spellchecker offers alternatives for mispelled words -- and then processed by the simulation mechanism. The mechanism, explained in chapter 7, determines how the agent will respond. Using natural language forces the student to recall, instead of simply recognizing, the knowledge needed to perform a social transaction.

The use of typed English was possible in Dustin because employees who are coming to the training in St. Charles have had training in English as a second language. They all know how to write in English, and what they need the most is practice with native speakers and exposure to the target environment.

System	Screen	Student
Exactly how long will you be in the United States?		"a couple of weeks"
Aha, that's good. Ok, your papers appear to be in order, you're going to get your luggage from carrousel number two.		**Now what?**
		

System	Screen	Student
		"Thank you"
You're welcome. Bye now.		
How would you like to get to St. Charles? Take a bus, take a limo?		Doesn't understand the word "LIMO." Double-clicks on the word to get a dictionary.

System	Screen	Student
Note: Dictionary allows Translation & Sound playing.(Say). -		

(9) Provide tools for addressing communication problems (e.g., dictionary, transcript of conversations, etc.)

Many of the resources available in real life can be easily implemented in computer-based systems. In the example above, the student doesn't understand a word and decides to look it up, and simply double-clicking on the word calls up the dictionary entry for that word. When sound is available, the student can listen to how a word is pronounced.

Two other tools appeared earlier: subtitles and translations. When interacting with the tutor, the student can request translations; when interacting with simulated agents, the student can request subtitles for utterances heard and also translations for those utterances. In figure 2, the button "What did you say?" displays the utterance in text form, and "Translate" translates it into the source language.

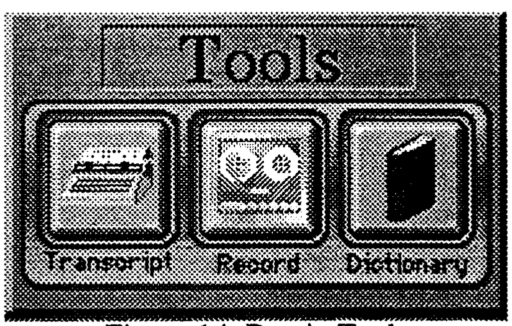


Figure 14. Dustin Tools

(10) Provide tools for analysis and reflection (e.g., recorder)

A recorder can be used to record and compare one's pronunciation with those of the native speakers in the simulation. While watching an example, the student can record his own voice then replay the line in the example for comparison.

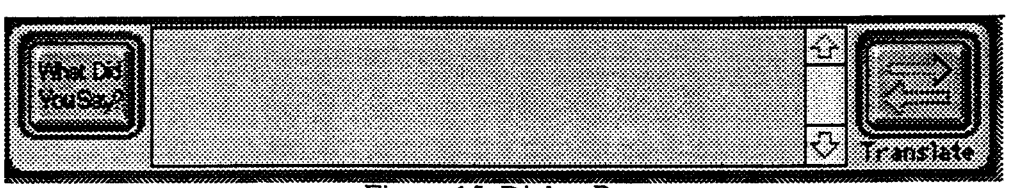
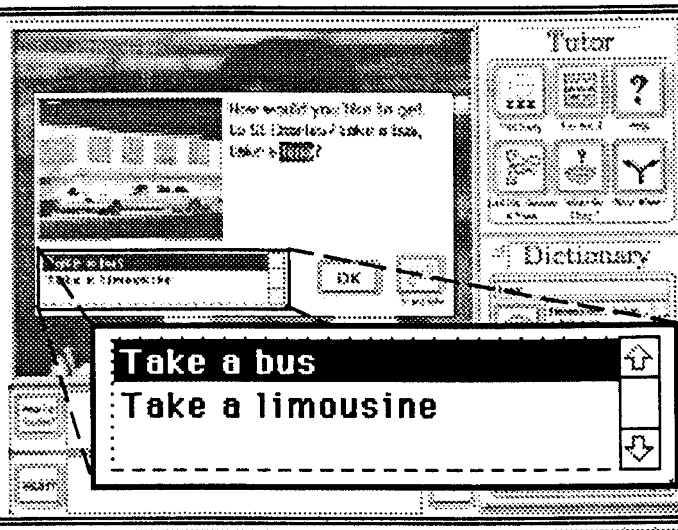
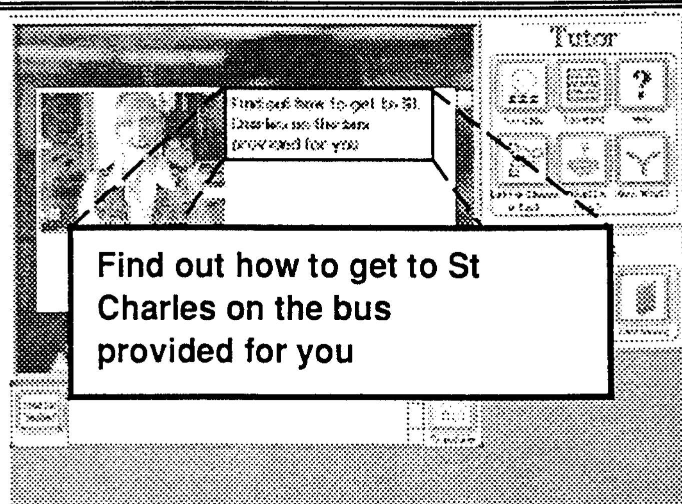
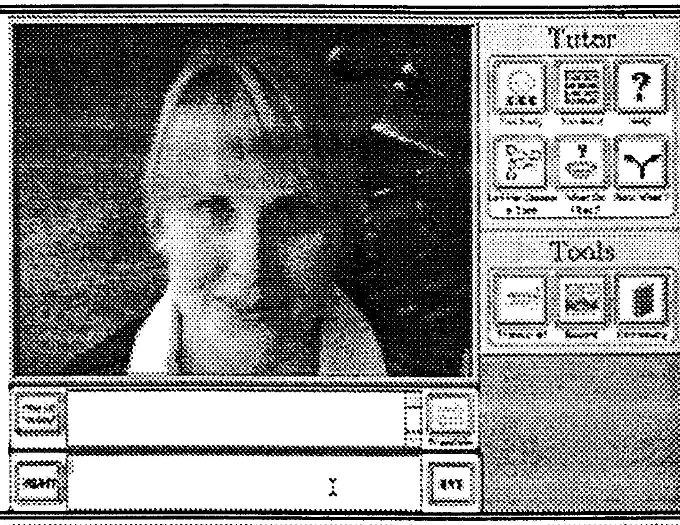
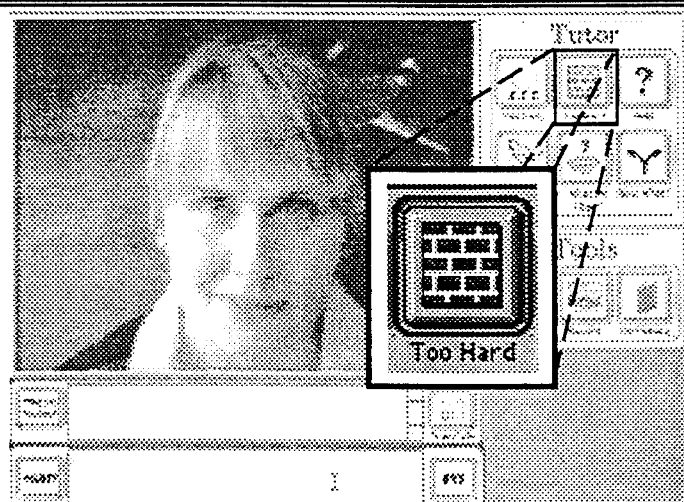
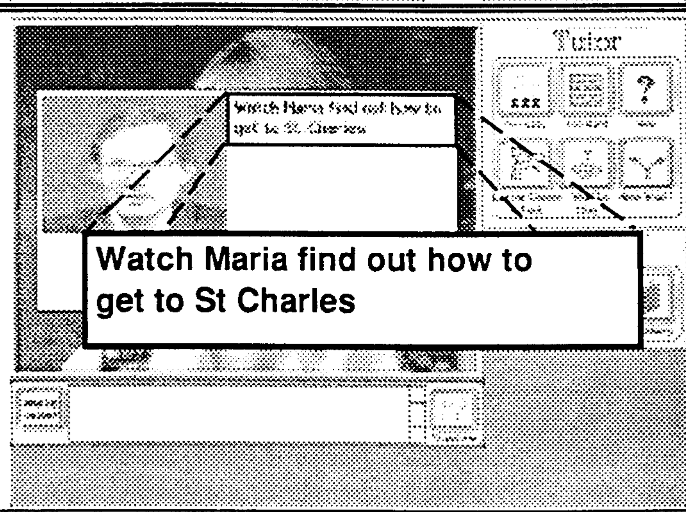


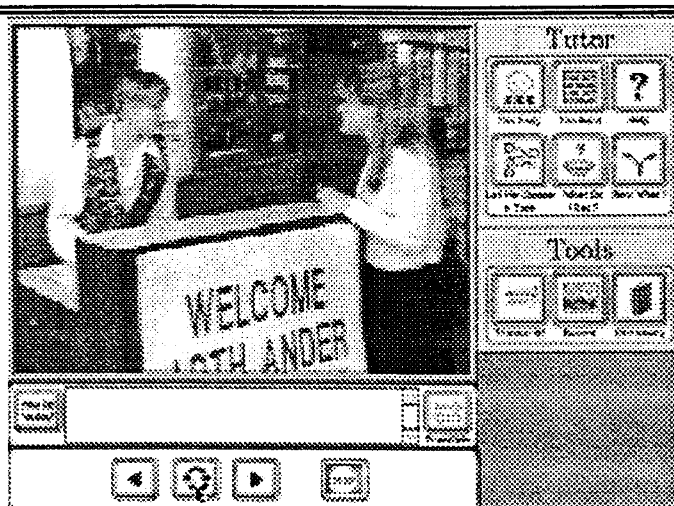
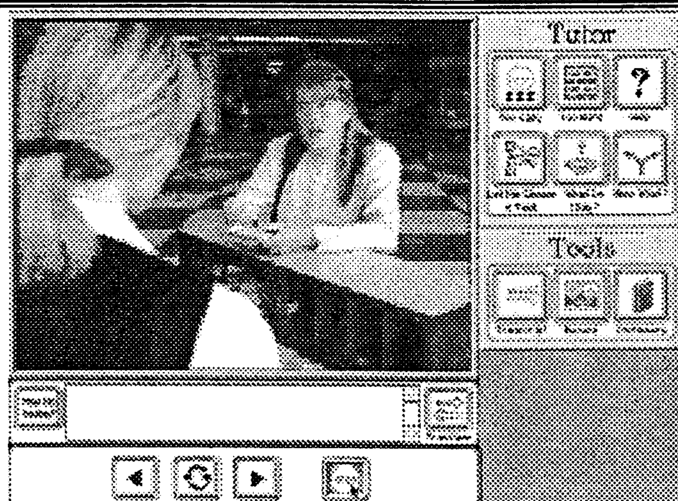
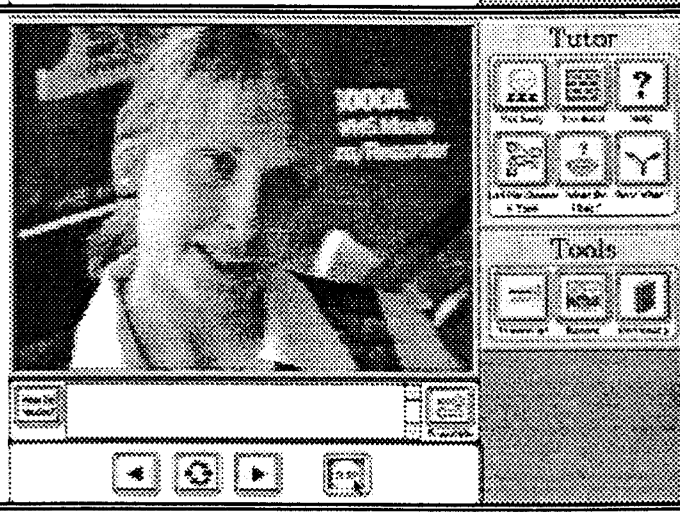
Figure 15. Dialog Box

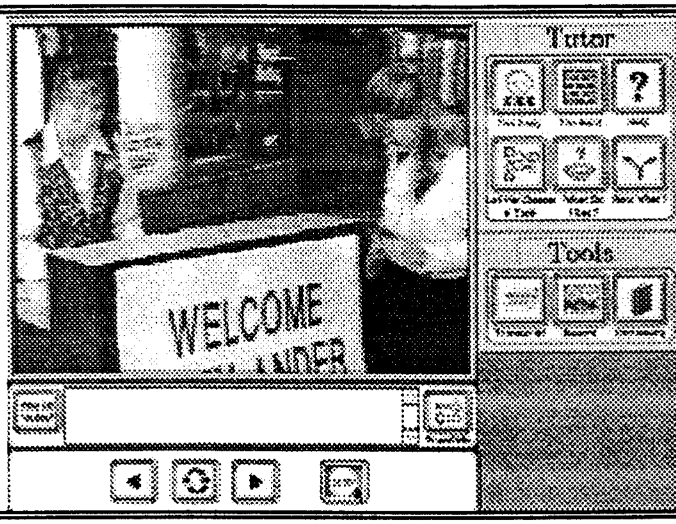
System	Screen	Student
		Take a bus
Find out how to get to St. Charles on the bus provided for you.		OK

(5) Goals must be meaningful to the student. Language is a means to achieving the goal, not the goal itself.

The system always tells the student to perform tasks that will be relevant to the student's needs in real life. Here, the student has to find a way of getting to St. Charles. All the tasks that the student pursues in the simulation depend on language and occur in the student's real life.

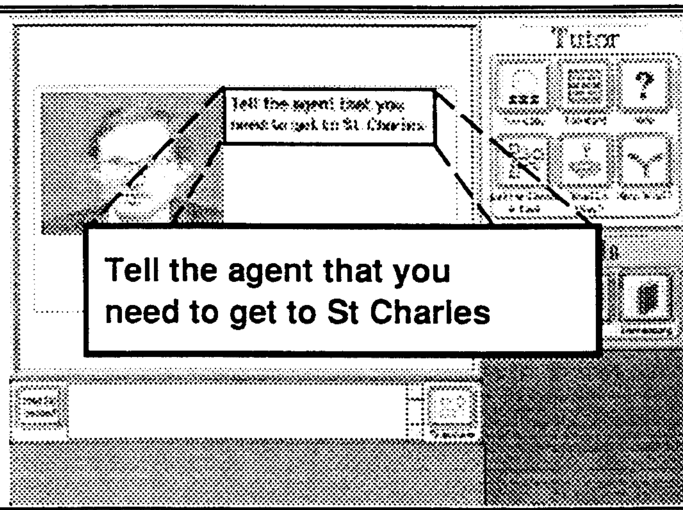
System	Screen	Student
May I help you?		**Student is confused.**
		**Decides to say this is "Too Hard"**
		**OK**

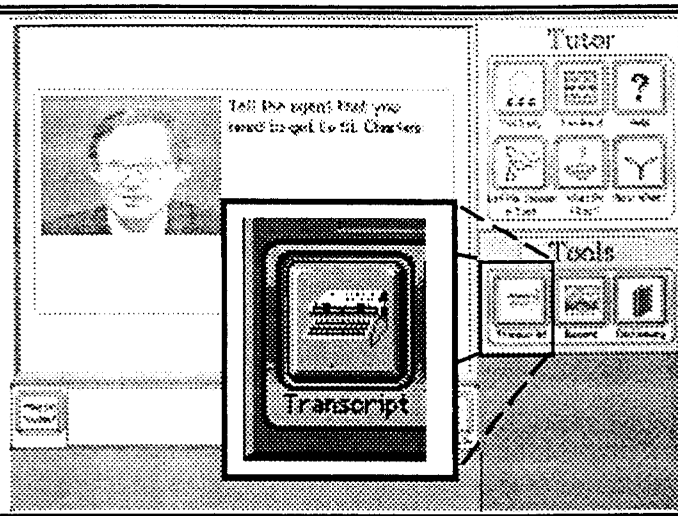
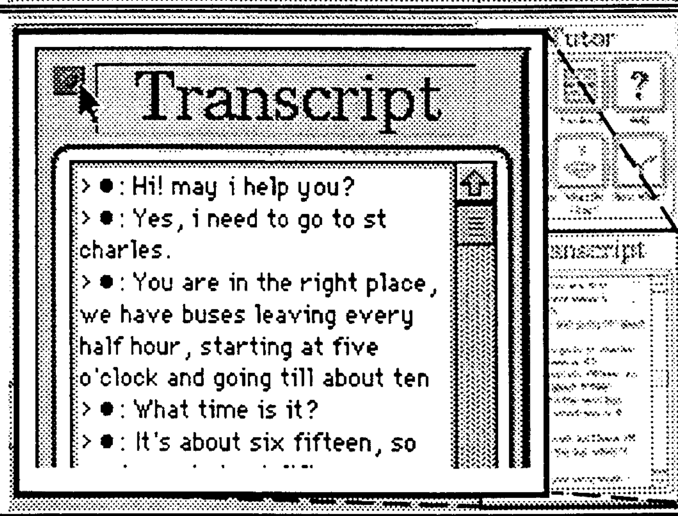
System	Screen	Student
Hi, may I help you?		
Yes, I need to go to St. Charles		Note: The student can <u>PAUSE</u>, <u>GO BACK</u>, <u>REPEAT</u>, and <u>SKIP</u> lines in the dialog.
You are in the right place, we have buses leaving every half hour, starting at five o'clock and going until about ten.		

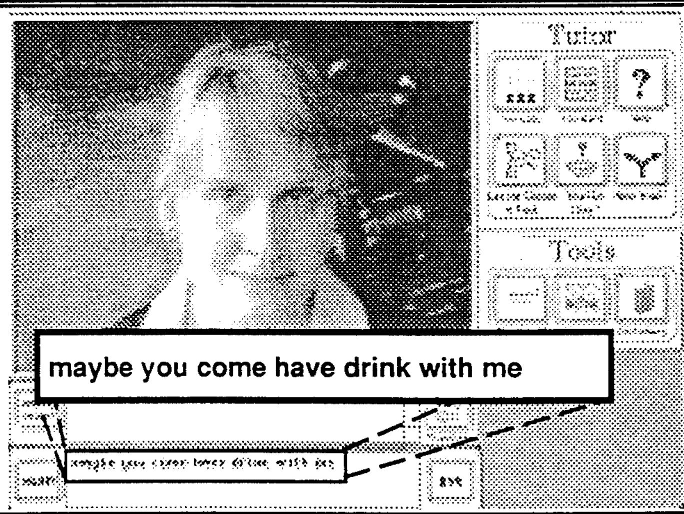
System	Screen	Student
• • • Thank you very much.		

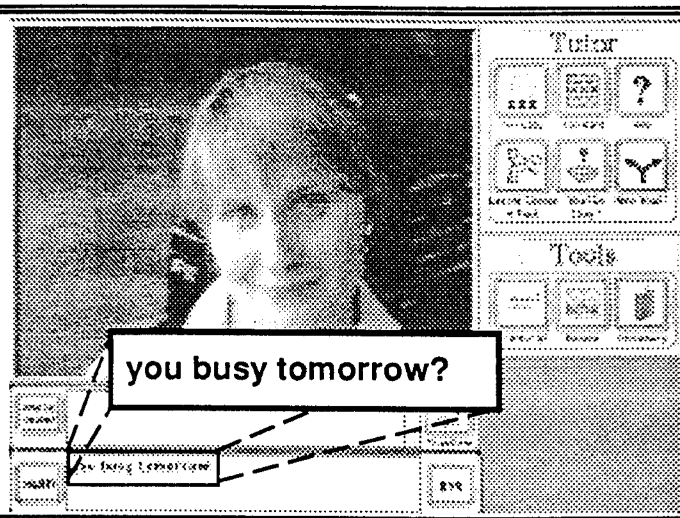
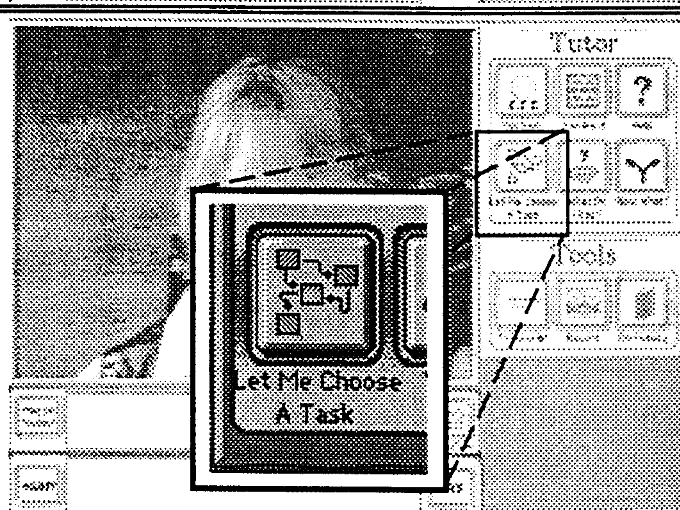
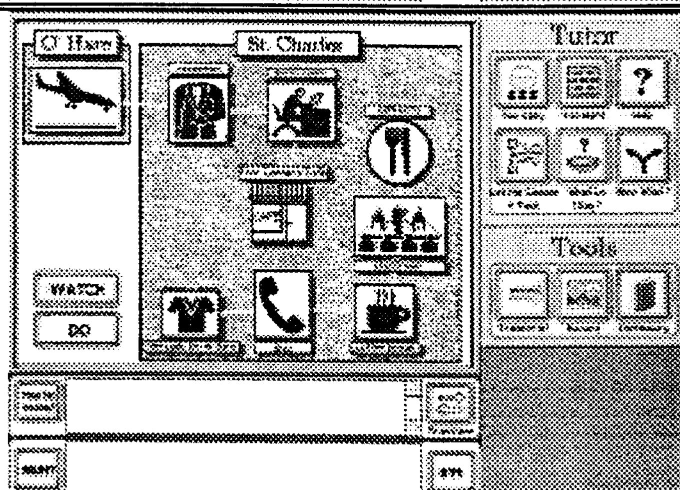
(6) **Engage in interactions**
- **If fail, then show an example**
  - **Try again**
  - **If fail, break the task down into smaller parts**
- **If succeed, move on**

The student got a chance to observe someone else performing the task. If the student can't perform a social transaction, Dustin shows her an example of someone else performing the same transaction, which is what happenned here. The student watches Maria trying to get a bus to go to St. Charles. She can pause, go back, repeat, and skip lines in the dialog. Once she is done watching the example, the simulation will throw her in the simulation again. Dustin follows the sequence of events outlined in principle 6.

System	Screen	Student
		OK

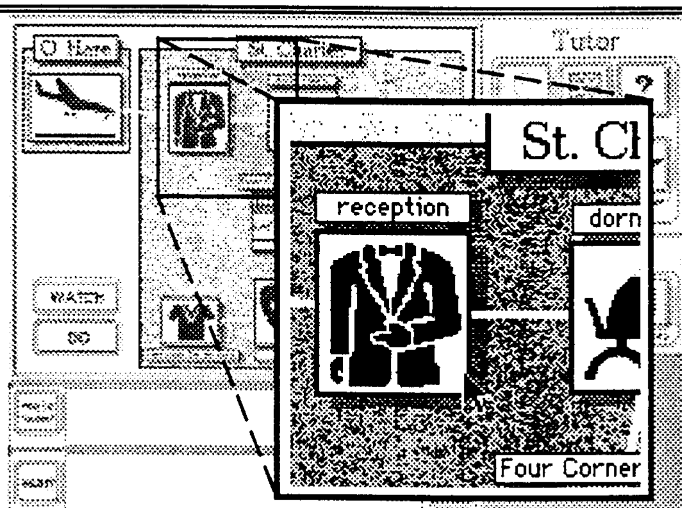
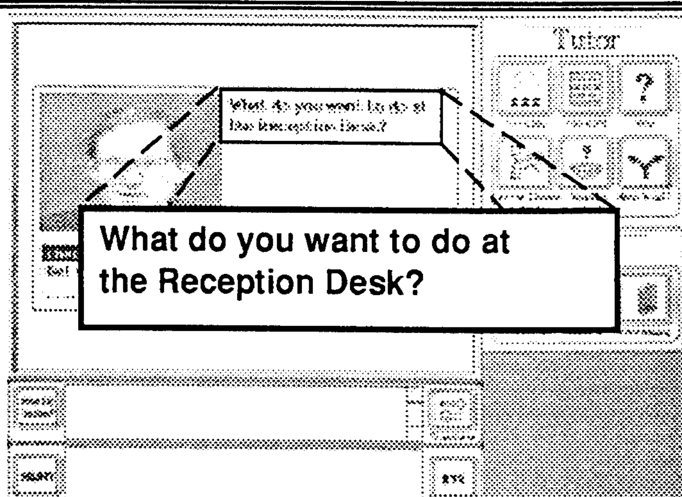
System	Screen	Student
		**Wants to see a transcript of the conversation.** **Transcript**
		**Closes the transcript and hits OK.**
Do you need help?		**"I want to go to st charles"**

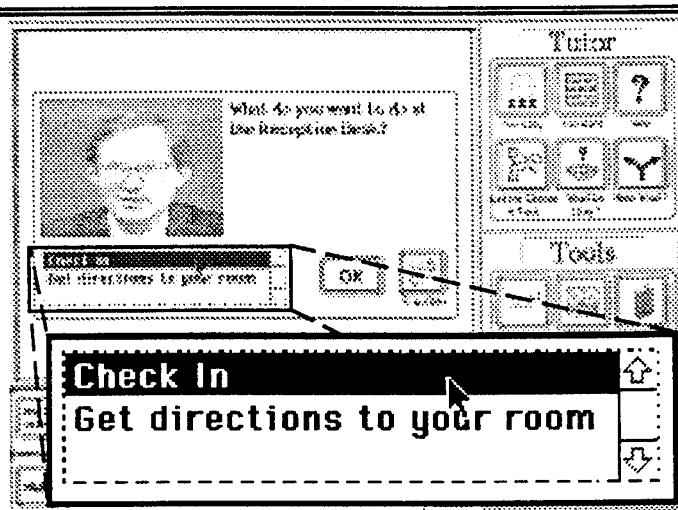
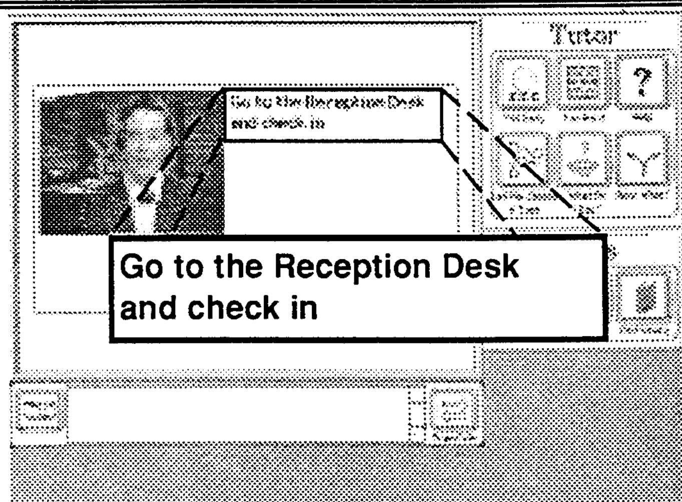
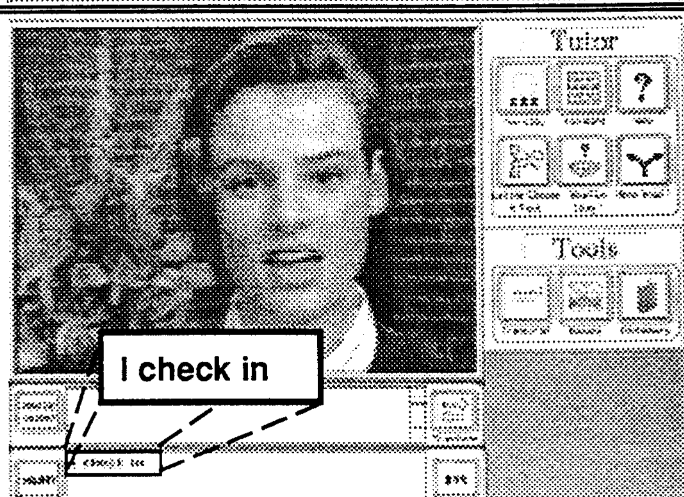
System	Screen	Student
You are in the right place, we have buses leaving every half hour, starting at five o'clock and going until about ten.		Student types: "When do they leave?"
Next bus leaves in in about 15 minutes. That's at six-thirty.		"maybe you come have drink with me" Note: Very common and predictable behavior when we tested the system.
I am sorry, but I am very busy right now.		"you want to dance tonight?"

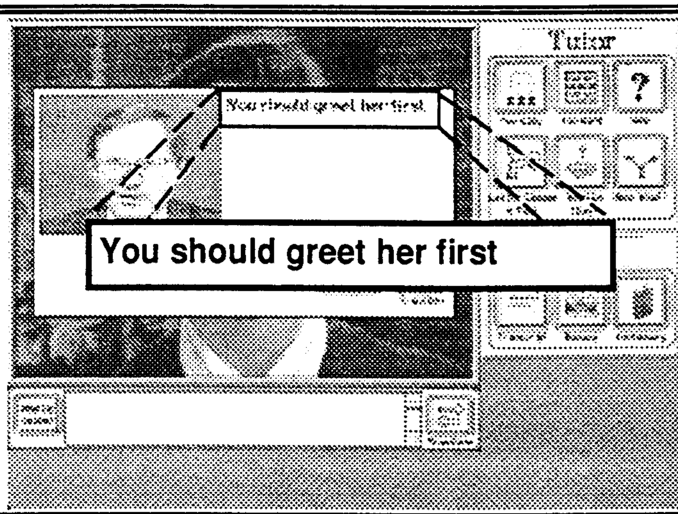
System	Screen	Student
If you have no further questions, I'd like to help the next person now.		"you busy tomorrow?"
I'm busy for the rest of my life!		The student decides to try something else. Let me Choose a Task
Note: The student can jump to any scenario.		

(7) Let the student have control over what to do in the simulated environment.

If the student is not satisfied with what the system is telling her to do, she can choose a task. She can jump to any scenario and either watch someone else or do the task herself. This is an important feature, as discussed earlier, because it allows students to address their own interests. The task selector in Dustin is organized by physical location. Once the student chooses a location, the tutor gives choices for tasks to perform in that particular scenario.

System	Screen	Student
		Chooses the RECEPTION
What do you want to do at the Reception Desk?		

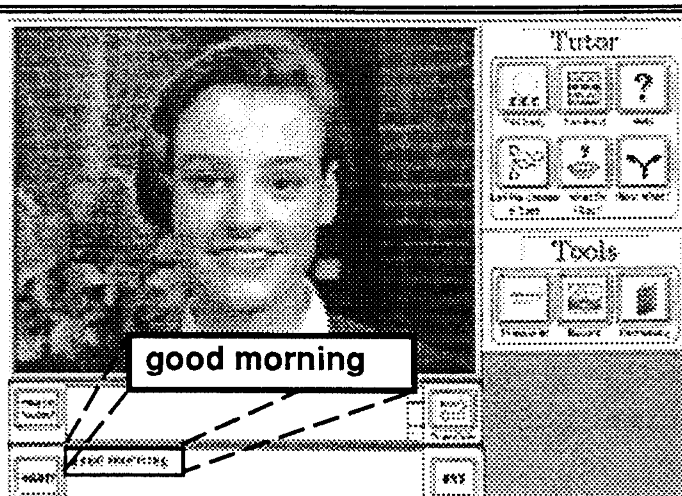
System	Screen	Student
		Chooses to <u>Check In</u>
Go to the reception Desck and check in		<u>OK</u>
Hi!		"I check in"

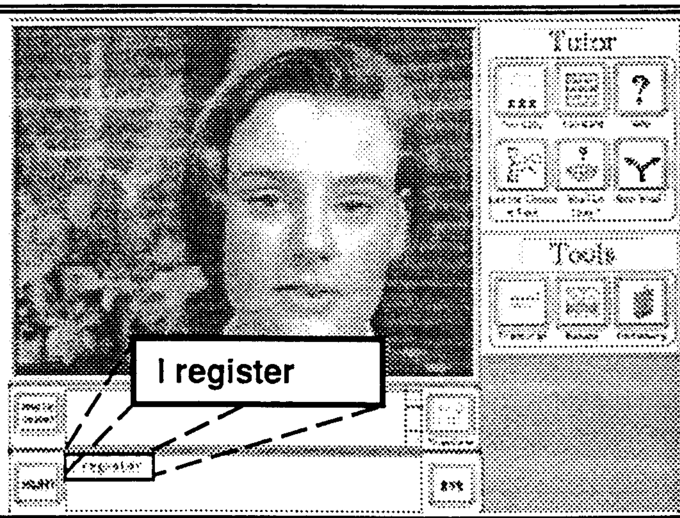
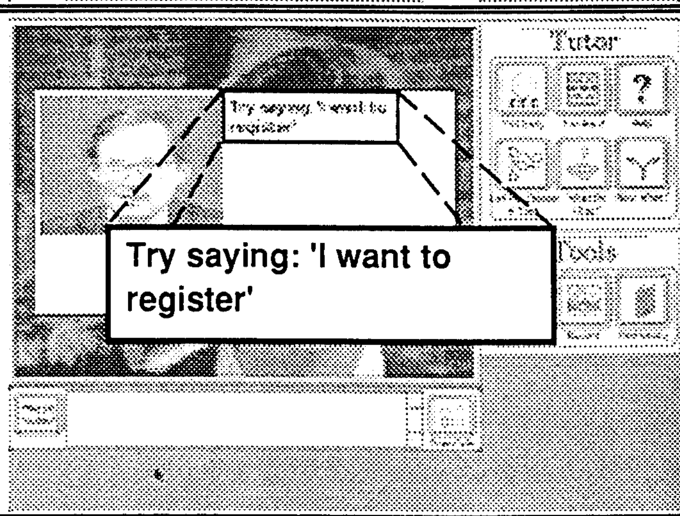
System	Screen	Student
You should greet her first.		OK

(13) Provide constructive feedback

The student didn't exactly do something wrong, but people respond better if we greet them first. The tutor provides this information in order to sensitize the student to the fact that she skipped an important part of the social protocol. Once the student does that, the receptionist is noticeably more friendly.

Not conforming to social protocols may lead to uncomfortable interactions in real life. When the student forgets to greet someone, or uses language that comes across as rude, the tutor intervenes with some explanation of what is the correct thing to say in such circumstances.

System	Screen	Student
Hello!		"Hi"

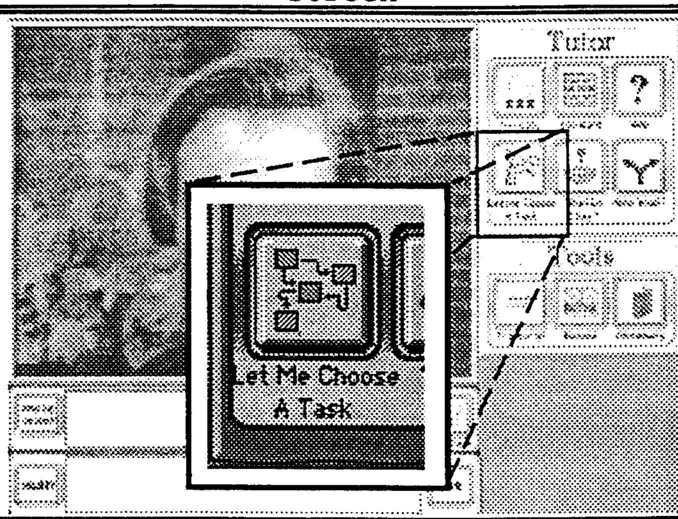
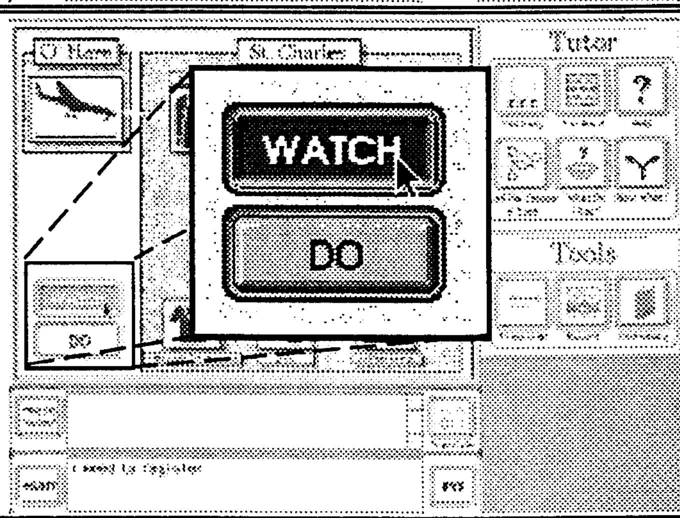
System	Screen	Student
Can I help you?		"I register"
Try saying, I want to register.		OK

(12) Do not correct grammar

Instead of criticizing what the student said, "I register," which would focus attention on structure, the tutor simply suggests a better way of expressing the desire to register. Instead of correcting grammar, the tutor suggest a correct way of conveying the message.

(8) Do not keep scores. Do not evaluate. Do not keep records.

The system does not keep scores, does not evaluate, and does not keep records. Students play with Dustin because they are interested in performing well when they get to St. Charles. They need no extrinsic forces to motivate them to do well.

System	Screen	Student
Your last name?		**Decides to explore.** **<u>Let me Choose a Task.</u>**
		**Student can either PERFORM (DO) tasks or WATCH other people performing them.** **<u>Watch</u>**

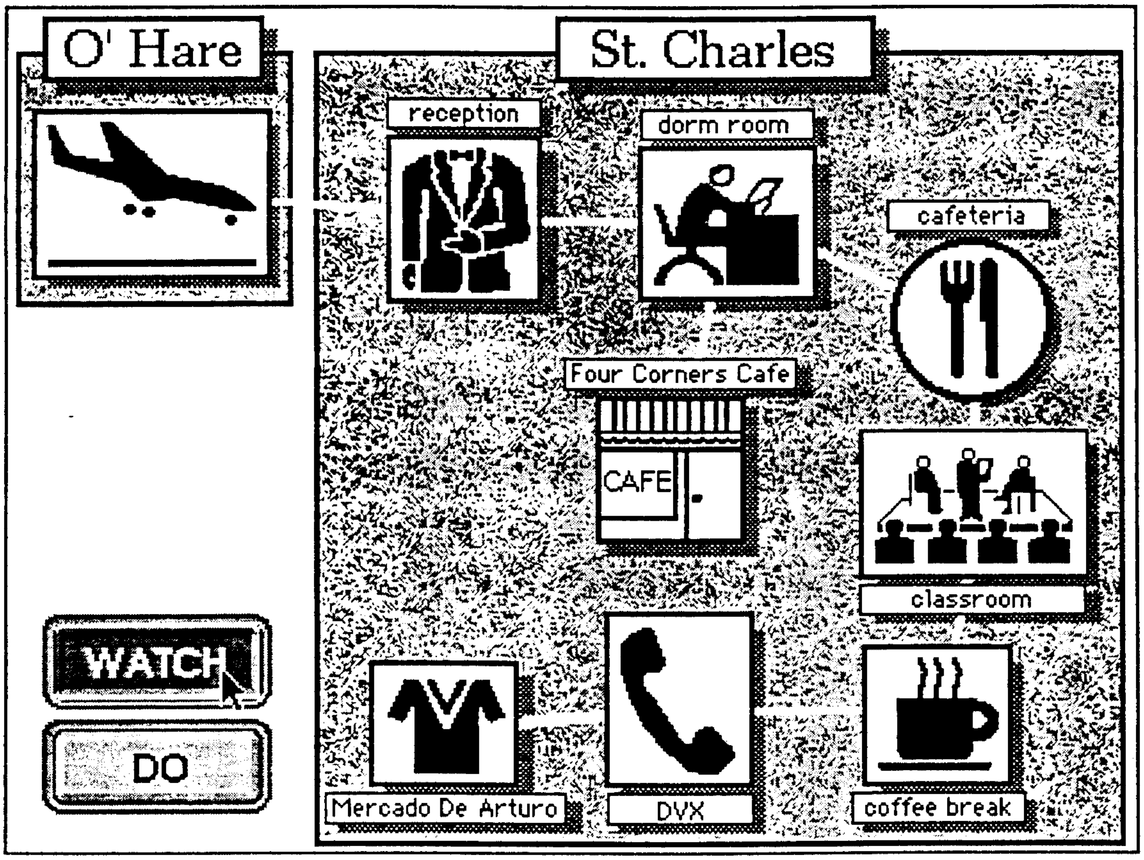


Figure 16. Dustin lets the student select a task.

There are sixteen tasks to choose from in nine scenarios.

Going through Immigration
at O'hare

Getting a bus to go
to St. Charles

Requesting a Limo to go to St. Charles

Checking In at St. Charles

Asking for directions

Meeting Roommate

Inviting roommate to eat

Getting breakfast at the cafeteria


Getting a sandwich at
Four Corners Cafe


Listening to Instructions
in the CAPS class


Helping a friend who
arrived late for class


Meeting people


Leaving a DVX message


Buying a T-shirt at
Mercado de Arturo

Conclusion

Dustin implements all the principles outlined in the previous chapter. Its simulations, for example, reflect closely the real-life situations that students face when in the United States. The immigration's officer, to mention a concrete case, is a real immigration's officer and it is not unlikely that, upon arrival, a visitor might interact with the very same person. The same applies to most of the people in the simulation; they work at the places simulated and their simulated behaviors very closely resemble their real-life behaviors.

Regarding the desirable characteristics associated with the specific domain being simulated (principles 1 through 5), Dustin was tailored to address each one of them directly. Language is (1) presented in the context of interactions that occur during the first day a newcomer spends in the United States. The student gets to (2) see and hear all the relevant information involved in these interactions. Within reasonable bounds, people in the simulations (3) react authentically to student's utterances, and the (4) role that the student plays in the simulation is exactly the same she plays in real life. The student participates actively in the pursuit of (5) goals that are relevant to their personal needs, and use language in the context of achieving these goals.

At another level, many of the features of Dustin are independent of the particular domain it simulates. The interface and the engine that runs the interface, for example, are completely independent of the real-life situations that a particular implementation addresses. Also, the underlying design, or architecture, embodies ideas, such as teaching strategies, that are domain-independent. I describe and discuss these general features of the underlying architecture in the next chapter.

Chapter 6

The RPSS Architecture

What exactly is Dustin?

Dustin implements (a) an interactionist view of language, focusing on communication and social transactions, and does so in a (b) naturalistic environment, in which students learn a language by using it as a tool to perform social transactions. Dustin gives learners (1) interactions with native speakers through simulated dialogs, letting students engage in interactions that are always situated in the context of achieving some goal in the target culture. Dustin gives learners (2) exposure to the target culture, letting them explore the same environment they will face in real life. Audio and video give students access to extra-linguistic information and the interface allows them to communicate through extra-linguistic means (e.g., handing the passport). Dustin gives learners (3) individual instruction and feedback. The tutor intervenes whenever the student is having problems, and stands by (i.e., tutor window), ready to provide him with information necessary to complete tasks. If the student is having problems, Dustin provides remediation, working according to the student's needs and interests.

Dustin also observes the principles outlined in Chapter 4, taking into account characteristics of the (1) situation, (2) student, (3) their interactions, and (4) resources that contribute to learning. Table 7, reproduced from Chapter 4, lists these principles.

Table 7

Desirable characteristics of computer-based language learning environments

Variable	Desirable Characteristics	Shape
Context/ Perception	• (1) Present language in context, as it is used in real life. • (2) Provide visual and aural information.	**Situations**
Behavior/ Feedback	• (3) People in the simulated environment must react authentically to student's utterances.	
Role	• (4) The role that the student plays in the simulation is the role that he/she is practicing for in real life. The student participates actively, not reactively.	**Student**
Goals	• (5) Goals must be meaningful to the student. Language is a means to achieving the goal, not the goal itself.	
Sequence	• (6) Engage in interactions • If fail, then show an example • Try again • If fail, break task down into smaller parts • If succeed, move on	**Interactions**
Exploration	• (7) Let the student have control over what to do in the simulated environment.	
Threat	(8) Do not keep scores. Do not evaluate. Do not keep records.	
Tools & Tutor	• (9) Provide tools for addressing communication problems (e.g., dictionary, transcript of conversations, etc.). • (10) Provide tools for analysis and reflection (e.g., recorder). • (11) Help student perform social transactions. • (12) Do not correct grammar. • (13) Provide constructive feedback. • (14) Provide means for the student to interact with the tutor.	**Resources**
Conceptual Feedback	• (15) Provide means for the student to express important cognitive and affective states to the simulated people and to the tutor. Provide means of communicating common states (e.g. Huh?).	**Problems**
Natural Language	• (16) Use natural language interface (assist/facilitate spelling).	

A method of teaching

What is important about Dustin is that it combines these essential elements into a cohesive model of instruction. Dustin combines a (1) tutor, (2) simulations, (3) examples, and (4) tools in a model of instruction that, instead of transmitting information to the student, which characterizes traditional instruction, helps students learn by doing. Dustin engages the student in interactions and helps him learn in the process of performing situated tasks. Instead of feeding information, Dustin coaches the student through the process, establishing an apprenticeship relationship with the student (Brown, Collins, Duguid, 1989; Collins, 1988; Collins, Brown, Newman, 1989).

The important implication of this apprenticeship model is that the function of teaching ceases to be one of transferring knowledge and becomes one of engaging the student in situated tasks while providing the support and modeling necessary to promote experiential learning.

Language learning is a prime example of skills that, in real life, we learn through apprenticeship. Our parents coach us in the process of interacting with others. They show us how to say thanks, they correct our mistakes, and they explain things to us. This model of instruction, the apprenticeship model, has given way to the traditional schooling model, not because of ineffectiveness, but because of its resource-intensiveness. Apprenticeship requires individualized attention and access to the target environment, and these resources are in short supply. Fortunately, as technology brings the resources necessary to language learning, so does it bring those resources necessary to implement this more effective, though resource-intensive, model of instruction (Collins, 1988).

In this chapter, I discuss Dustin's architecture, Role Playing in Social Simulations (RPSS) in more detail. I start by describing its components and its model of instruction, and then discuss what's good and what's bad about Dustin, also comparing it to other systems. Finally, I suggest some future applications of Dustin-like systems.

Dustin's Components

Dustin, or the RPSS architecture, has four major components: (1) tutor, (2) simulation, (3) examples, and (4) tools. The student is always either (a) receiving guidance from the tutor, (b) participating in a conversation, (c) watching other people, or (d) consulting some tool. Figure 17 depicts these major components and their functions, which I describe in more detail next.

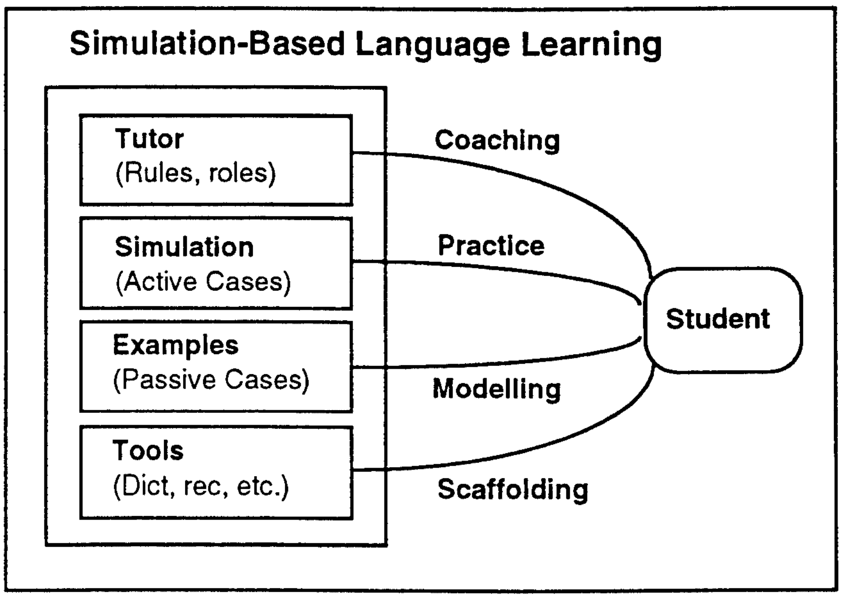


Figure 17. Components of the RPSS Architecture and their functions.

Tutor

Throughout our lives, other people help us acquire language. Our parents, teachers, colleagues, and peers are constantly coaching us. In Dustin, the TUTOR embodies this coach figure. The tutor assigns tasks and makes sure that the student follows norms of good communication. After giving her assignments, the tutor guides the student, helping with hints and providing feedback.

The tutor came into existence as a byproduct of the demands imposed by principles such as "Provide constructive feedback," and "Help the student perform social transactions," that called for a guiding figure. However, to understand the importance of a coach, we need to reiterate the importance of goals. As discussed in Chapter 4, goals are one of the most important keys to situate knowledge, because having a clear goal helps learners organize knowledge and understand feedback. Instead of relying on external sources to establish these goals, like *Direccion Paris* and *Zarabanda* that depend on goals set by the teacher, Dustin provides the goals to the student. Since goals are an important part of the context and subgoals must be clear even when a high level goal is known, Dustin motivates each task with clear goals, helping the student organize information by setting the conceptual context in which the task occurs.

Examples

We need exposure to meaningful input to learn (Krashen, 1985; Schank, 1991). In Dustin, the tutor guides the student to see relevant scenes or the student roams around eavesdropping, so to speak, on other people's conversations. Through observation, learners develop a conceptual model of the task before executing it. The idea that people learn from examples, or modeling, is defended by both second language theorists (i.e., Krashen's input hypothesis) and AI theorists (i.e., Schank's case-based reasoning). In the AI view, cases are contextualized by goals and the circumstances in which they occur -- tying back to the role of the tutor. All the examples in Dustin show people using language in the context of pursuing a goal which is described in advance by the tutor. This way, the student indexes the information in the example by the goals that it helps to achieve -- the context and the goal become indices for retrieving information.

Simulation

Simulations are at the core of Dustin's architecture; all other components support the student during interactions. The purpose of Dustin is to prepare learners to perform in real life, and it does so by engaging students in interactions with those people they will encounter in real life. The people in the simulations react authentically to student's inputs, and act very much the way they do in real life. Since most of the people in the simulation perform the same functions in real life, it is very likely that a learner will interact with the same person when she gets to St. Charles. By interacting in the simulation, the student uses her knowledge and receives essential feedback to fine-tune her skills. She becomes accustomed to the target environment, and since the tutor provides a very clear goal for each setting, she learns exactly what to do in each situation.

Tools

Learners use dictionaries, recorders, and other tools that help them develop communicative skills. Computers are particularly good for implementing memory aids, translations, guidebooks, and a variety of feedback tools. They empower students by complementing their knowledge, and computer-based simulations can accommodate a large number of these cognitive tools. Dustin implements a number of them (e.g., dictionary, transcript, translations, and a recorder).

Dustin's Model of Instruction

Dustin's model of instruction, apprenticeship, resembles the way children learn with their parents. Parents serve as coaches (e.g., "go ask your aunt if you can have a cookie"), provide examples (e.g., "say, may I have a cookie?"), serve as test bed for interactions (e.g., "try saying it to me, let me see if you can"), and often assist the child in interactions with others (e.g., "no, not ookie, Cookie").

The apprentice receives personalized attention and engages in tasks that are always compatible with his level of competence. The expert helps at critical points and serves as a source of information. The typical process of learning in the apprenticeship model (Collins, 1988) involves observation, coaching, practice, and scaffolding (i.e., holding the learner's hands until he is ready to perform alone). In Dustin, the student acquires skills by working with a master (tutor -> coaching) who provides the necessary guidance in performing tasks in a specific context (simulation -> practice). When the apprentice needs information, the tutor exposes him to the necessary information (examples -> modeling), and Dustin holds the apprentice's hand until he can do it on his own (tools -> scaffolding).

This type of apprenticeship, originally used to teach physical skills but adapted to teach cognitive skills, is resource intensive and rarely cost-effective (Collins, 1988). Providing individual attention and practice to each apprentice requires resources that are rarely available, not unlike the current problem in language instruction. However, as in language instruction, technology allows us to go back to resource-intensive models of teaching (Collins, 1988; Collins, Brown, Newman, 1989), enabling us to provide personalized attention and to expose students to tasks at their level of competence.

This approach of coaching instead of transmitting information, the cognitive apprenticeship model (Collins, Brown, & Newman, 1989), has been used successfully by researchers and educators to teach reading (Palincsar and Brown, 1984), writing (Scardamalia & Bereiter, 1985), and math (Shoenfield, 1983, 1985). A brief description of the characteristics of the cognitive apprenticeship model reveals the parallel between this model and RPSS' model. The most salient characteristics of apprenticeship are (1) situated learning, (2) modeling and explaining, (3) coaching, (4) reflection, (5) articulation, and (6) exploration (Collins, 1988). <u>Situated learning</u> means that learners acquire knowledge used in real tasks; Dustin engages the student in exactly those situations that he'll face in real life. <u>Modeling</u> is providing examples, showing either a process or an expert performance; Dustin has at least one example for every task it tells students to perform, allowing students to observe a native speaker and integrate what happens with why it happens. <u>Coaching</u> is giving personal attention; Dustin's tutor helps address individual difficulties at critical times, guides the student through social interactions, and provides feedback when appropriate. <u>Reflection</u> allows the student to study what he did, compare to others, abstract from his knowledge, and compare his knowledge to abstractions; Dustin provides a recording tool, allowing students to compare their speech to that of others in the simulation. <u>Articulation</u> helps the student make implicit knowledge explicit -- making knowledge more available and comparing structure and encouraging insight; Dustin help here is limited, and I discuss this limitation later in this chapter. Finally, <u>exploration</u> allows students to try out different things, helping them learn how to set achievable goals, form and test hypothesis, and make discoveries; Dustin lets students have control over the tasks that they perform and observe.

The cognitive apprenticeship model is of interest in the context of RPSS systems because it provides a metaphor that integrates RPSS' components, functions, and ideas. While some of the theories presented earlier explained parts of Dustin, this metaphor connects Dustin's components. The interactionist/naturalistic view explains the need for interactions in authentic situations (i.e., practice); Krashen's input hypothesis and case-based reasoning explains the need for examples (i.e., modeling); work on intrinsic motivation shows the need for clearly defined goals and constructive feedback (i.e., coaching). The apprenticeship metaphor clarifies the role of the tutor as an expert-guide who provides help, and reveals the need for scaffolding (i.e., tools) to help the student walk on his own. And what is more important, it connects all these components and functions.

In AI terms, Dustin's way of teaching contrasts with methods that teach rules rather than cases. Much of language instruction is based on structural teaching strategies that try to teach rules apart from the cases from which they derive and apart from the context in which they are used. The context from which a rule derives is what indexes the rule; Dustin teaches these cases, not the rules -- relying on inductive acquisition of grammar. Word games, syntactic structures, conjugation of verbs, and agreement are just a few of the labels under which decontextualized rules are organized and taught in classical language instruction. Structural strategies encourage the creation of a mapping between the source language, one's native language, and the target language, the language one is trying to learn. Dustin uses language as a tool for achieving goals in specific situations, and instead of encouraging the creation of a mapping function between two languages it encourages the creation of a mapping function between a goal to be achieved in a certain context and the language that is necessary to achieve that goal.

What's good about Dustin?

I mentioned earlier that living in Japan is one of the best ways of learning Japanese -- a perfect example of learning in a naturalistic setting. I described what happens when we move to a foreign culture, pointing out how a number of elements contribute to learning while living in Japan. Comparing the Japan-experience and the simulated St. Charles-experience, shows how Dustin implements many of the same elements that contribute to learning in naturalistic settings. Below, paraphrasing the description of what's good about living in Japan from Chapter 4, is a description of what's good about using Dustin.

A number of elements contribute to learning in the Dustin-experience. The person is highly <u>motivated</u> to learn English to go about his life when he gets to St. Charles. Language is crucial, and the drive to perform well and learn stimulates learning. The <u>goals</u> that the individual wants to achieve in St. Charles motivate his role-playing in the simulation. He spends a period of intense <u>immersion</u> in the target environment -- hearing, seeing, and interacting with people who use only English to communicate. The student is exposed to the <u>context</u> in which language is used. This includes physical, social, and task settings as they occur at O'Hare and in St. Charles. He <u>participates actively</u>, engaging in interactions that require the use of language. These interactions force the student not only to recognize words but also to recall and adapt knowledge to the situation. He receives <u>help</u>. Simulated native speakers and Dustin's tutor correct errors or suggest better ways of saying things, and the environment itself is fortuitous in that it provides referents that assist in communication. The way people react, their <u>behavior</u>, provides the learner with essential <u>feedback</u> to adjust behavior. The student tries to understand patterns of the language, and questions what words mean in different contexts; in other words, he engages in <u>analysis and reflection</u>. When communication breaks down, due to insufficient competence or performance variables, the learner uses <u>extra-linguistic</u> means of communication (e.g., the HUH? button, dragging objects) that facilitate social interactions. When at a loss for words, he resorts to <u>tools</u> like dictionaries, recorders, and translations. The social context and the interactions with others elicit emotions, <u>affect</u>, that influence learning. Finally, different people use different <u>learning strategies</u>, because they learn in different ways. Table 8 shows how Dustin compares with the living-in-Japan experience, according to the variables that influence language acquisition. Table 9 summarizes Dustin's features.

Table 8

Comparing the real-life Japan-experience and the simulated St-Charles-experience.

Variable	Description	Dustin Vs Japan
Motivation	Highly motivated. Personal needs have to be met; language is a powerful tool to satisfy them.	Same
Goals	Survive, make friends, establish oneself, thrive.	Same
Immersion	Completely immersed, twenty-four hours a day.	Shorter, but more intense exposure.
Context	Native speakers, target culture and environment, real life situations.	Same
Active participation	Engage in interactions continuously.	Same
Help	People help, friends help, tutor helps.	Same
Behavior/Feedback	The environment is fortuitous, people give feedback all the time.	Same Limited feedback on speech.
Analysis and Reflection	Study the language, think about the culture and social protocols.	Same
Extra-linguistic Communication	Gesticulation, mime, pointing, facial expressions	Conceptual feedback, object manipulation.
Tools	Use dictionary, tape recorder, translating machines.	Same
Affect	Emotions influence learning and interaction with the environment.	Non-threatening environment.
Personality	Traits determine student's relationship with the environment.	Better. Facilitates interactions for introverts.
Learning strategies	Tricks used to learn the language itself.	In principle, same.

Table 9

Features of the RPSS architecture

Feature	Description
View of language	Interactionist view
Approach	Naturalistic
Interface	Natural language
Exploration	Student-driven Tutor-guided
Practice	Simulation
Assistance	Tutor Contextualized help
Extra-linguistic communication	Conceptual feedback Video Audio Object manipulation

Who's using it?

During the development of Dustin, a number of ILS professionals, students, Andersen Consultants, and St. Charles trainees tested and helped us debug and improve various aspects of Dustin. To test Dustin with Americans, we developed RPSS prototypes to teach Japanese, Lithuanian, and Spanish. These prototypes taught us a few hard lessons about using typed input -- the target audience that can benefit from natural language interfaces is very well delimited. Over the duration of the project, eighteen months, we collected data from users through questionnaires, video tapes, and interviews. The data collected guided us through each of the many iterations that comprised the development of the final version. In the Summer of 1991, after about twelve versions, Dustin was finally installed in the Madrid office of Andersen Consulting.

ILS is now working on a French version of Dustin for travelers going to France -- Paris to be more precise. The idea is to implement Dustin on a portable unit, so that travelers can use it during their flight to Paris. This French version implements some changes based on more recent reports and feedback we received from users.

What do they report?

Dustin helps. Twenty-five foreign trainees from Japan and Spain, who tested an early version of Dustin, thought that Dustin would have helped them improve their ability to interact with people in O'Hare and St. Charles and that the exposure to the St. Charles environment prior to their coming to the United States would have put them more at ease in that environment. Exposing themselves to the people that they eventually meet in real life, listening and interacting with them in the simulated setting, generates a sense of familiarity with the place and the personnel in St. Charles. The informal data collected from these subjects show that Dustin helps students familiarize themselves not only with the target environment but also with the speech patterns of native Chicagoans. Trainees get used to the particular accent of those people working in St. Charles and O'Hare, and altogether, Dustin prepares trainees to face the impact of the first twenty-four hours in the St. Charles culture.

Although the comments received so far have been enthusiastic, this preliminary data has also pointed out some problems, not solved in Dustin, which I discuss next.

Situated Learning & Transfer

Dustin is still too small to promote transfer. The implementation of Dustin described here is large enough to guarantee its scalability -- which means that implementing a larger version would pose no technical problems. However, to compare and extrapolate the embedded knowledge to other situations, learners need a larger number of similar, but not identical, examples. Students benefit from exposure to variations on themes (e.g., different ways of greeting, different people greeting, people greeting in different situations), and Dustin has too limited a number of variations. Trainees reported that, although useful to prepare them to interact with people in St. Charles, Dustin was too limited to help them interact in other settings. Dustin needs more cases.

This is a problem that has been pointed out in apprenticeship (Collins, 1990). Students acquire knowledge in a particular context and have difficulties transferring the knowledge to other domains. To address this problem, besides being exposed to more cases, students may need help abstracting from the particular situation. Collins suggests that this can be done through articulation, and reflection. <u>Articulation</u> involves an effort to integrate pieces of information gained from particular situations, thinking about how what one knows in one context relates to the knowledge needed in other situations (Collins, 1990). For example, following a scene in a Japanese restaurant, the tutor could ask "Why do you think he is taking off his shoes?" The current version of Dustin does not encourage or demand this type of thinking from the student. <u>Reflection</u> involves looking back at what one did, trying to analyze one's performance or compare it to others'. To help the student reflect, Dustin could ask the student questions like "Why did he get upset? Did you miss something?" Here again, Dustin does not explore this type of reflection. It would not be hard to implement these features, and in fact, the idea of adding cultural notes explaining such things, suggested by a number of users, would come as a perfect model after articulation and reflection questions.

Exploring learning strategies

Dustin does not explore learning strategies. Not exploring particular learning strategies is actually more a problem of missed opportunity. Learning strategies are special tricks that people use to learn a language (e.g., using visual imagery to understand new information, classifying words according to their meaning, associating phrases with body language), and technology offers a great potential for supporting learning strategies with tools that would not be available otherwise. For example, imagine someone who likes to stand in front of a mirror and observe himself as he imitates someone, associating speech and intonation with facial expression. To explore this strategy, we could provide tools for him to video tape himself, allowing him not only to see himself as he would in a mirror, but also to play back and compare his performance to that of others. Dustin supports some learning strategies by allowing students to have control over the environment, letting them explore it according to individual preferences. However, Dustin does not provide any tool that directly explores any particular learning strategy.

Exploring the social context

I have not explored the social context in which Dustin is used. It is a mistake to think that how a system will be used in a larger social context is a problem that lies outside the scope of its design. As I showed when describing *Poker Pari* and as related research also shows (Scardamalia, Bereiter, McLean, Swallow, & Woodruff, 1989), how a system is used in a larger context may well be the best feature of a given system. I have designed Dustin as a stand-alone system, making it less threatening than the typical real life environment. But in doing so, I missed the opportunity to explore competition, cooperation, argumentation, and other social forces that promote learning. The potential to explore these social factors was obvious during sessions in which more than one user tested Dustin at the same time. They consistently engaged in discussions about the scenes, talking about how to handle simulated situations, generating hypotheses together, laughing, and encouraging each other to try things out. To properly use RPSS systems, I must consider these socio-psychological factors. Peer pressure and other social factors are extremely important in learning, and so far, Dustin leaves out this significant element of the learning process.

I want to belong

Finally, Dustin may not be good for learning that depends heavily on affect. One instance of this dependence occurs when students want to learn a language to feel accepted in the community. Research shows that the attitude of learners towards the target culture and language influences language acquisition. Dulay, Burt, and Krashen (1982) showed that the level of proficiency achieved is influenced by whether a learner wants to simply use language for utilitarian reasons (e.g., a Hispanic uses English to go shopping, but Spanish for social interactions), to integrate in the target society (i.e., participate in the life of the target culture), or to identify himself with the social group (i.e., belong to the target culture). The desire to belong, as the research shows, leads to higher proficiency. I believe that the RPSS architecture helps those with utilitarian motives. These people seek to develop repetitive, well-scripted interactive skills, and in these situations, RPSS systems can simulate most of the interesting cases. On the other hand, there are limitations in using RPSS systems to promote integration and identification because these involve complex interactions that depend heavily on affect. Unlike real people, simulations cannot truly satisfy the learner's need for affection.

RPSS and other CALL systems

Compared to other CALL systems, Dustin combines essential components, functions, and ideas that the others don't. Although each CALL system introduced earlier contributes useful ideas, even the most advanced of them fails to combine some essential elements. For instance, *French in Action*, the video-based language course, provides numerous examples, coaching, tools, and is based on a naturalistic approach. However, *French in Action* is non-interactive. Consequently, its coaching is not individualized and active interactions with native speakers never occur. *Poker Pari* and *Dark Castle* engage students in interactions, but their approach still encourages the study of language itself, which is ineffective for developing communicative competence. They also lack the advantage of multimedia, with the associated extra-linguistic information. *Zarabanda* and *Direccion Paris* implement interactive multimedia, exploration, and supporting tools, with a great amount of extra-linguistic information available to the student. But, like

French in Action, they do not engage students in interactions with native speakers. *No Recuerdo* promises to combine multimedia, tools, simulations, and examples in a interactive fiction. However, it lacks a coach. Students using *No Recuerdo*, besides lacking individualized instruction and feedback, may find it difficult to organize information without a clear definition of the context in which an interaction is taking place.

None of these systems combines all of these elements: interactional view, naturalistic approach, multimedia, coaching, examples, simulation and tools. Given the need for interactions, aural and visual information, individual attention, models, practice, and scaffolding, these six elements are essential to support experiential learning. Table 10 compares the features of each of the systems mentioned earlier.

Table 10

Features of Dustin and other CALL systems

(na - information not available).

Feature	French in Action	Poker Pari	Dark Castle	Zara banda	Direccio n Paris	No Recuerdo	Dustin
Interactive		√	√	√	√	√	√
Interactionist View	√			√	√	√	√
Naturalistic Approach					√	√	√
Multimedia	√			√	√	√	√
Inductive Acquisition of grammar				√	√	√	√
Examples Exploration				√	√	√	√
Tools					√	na	√
Object Manipulation					√	na	√
Simulation						√	√
Natural Language						√	√
Guided Exploration							√
Conceptual Feedback							√
Cohesive model of instruction.							√
Coaching							√

What's RPSS good for?

Dustin was designed to address a very specific problem: help foreign trainees of Andersen Consulting coming to St. Charles for the first time. The model of instruction that it incorporates, however, can be used in other domains. The interface may have to be changed to fit new domains, but the overall architecture, including simulations, models, tutor, and tools stays the same. Below, I suggest some applications for Role Playing in Social Simulation (RPSS) systems. With superficial changes, RPSS can be used to promote experiential learning through apprenticeship in a variety of domains.

Teaching other languages

Dustin can teach other languages. Nothing in Dustin makes it applicable to teach only English. The small prototypes that we built, using the same architecture, to teach Japanese (going to a Japanese restaurant), Spanish (going to a snack bar), and French (meeting people) posed no adaptation problems. Besides, since we were considering users from a number of countries, these prototypes were flexible enough to accommodate a number of source languages (e.g., Spanish, Italian, French). Given its design, we are having no problems developing a French version for Americans, and it would be perfectly possible, for example, to implement a simulation of a business negotiation in Japan to help Brazilians. Besides changing the content of the simulation, we only have to change button labels and translations to accommodate different source languages.

Other Apprenticeship

RPSS can teach auto mechanics. Imagine an auto mechanic undergoing apprenticeship with a master mechanic. A customer comes with a problem (e.g., "the engine is stalling when in idle"), and the master throws the student in a simulation: "fix it." Since computers allow object manipulation, it is possible to replicate the parts of an engine. Instead of interacting in English, the learner interacts by manipulating parts of the engine, disassembling, testing, and inspecting components. The tutor, as in Dustin, intervenes by either asking information-loaded questions, "what makes an engine choke?" or giving hints, "something seems to be missing in the carburetor."

This simply expands the TPR (Total Physical Response) style of instruction. In Total Physical Response (Asher, 1977), students learn languages by responding physically to input (refer to Chapter 2), standing up, sitting down, moving things, etc. Here, students learn by interacting not only with simulated people but also with simulated systems. Computer-based environments are particularly suited for this type of simulations, and using Dustin in this example would require simple changes. Instead of typing "I want to register," for instance, moving objects would generate messages like "carburetor removed." The RPSS architecture may prove useful for training in which students learn by interacting with physical systems (e.g., computers, engines, circuits).

Communication Skills

RPSS develops communication skills. Learning a language is learning WHAT to say and WHEN to say it to achieve a GOAL in a specific CULTURE. Language is a tool for communication and cannot be separated from the situations in which it is used. Learning a language is learning that a hostess in a Japanese restaurant will welcome you with the word "Irashaimasse," while bowing to you, and that the proper response on your part is to nod ever so gently to acknowledge her courtesy. Language learning entails learning about goals and the communication skills necessary to achieve them in a specific culture. Notice, however, that learning how to be a bank teller is very similar. Learning how to be a bank teller is learning a new way of interacting with people, it is learning a new language, it is learning a new culture and the goals associated with the role. This means that the same RPSS architecture can teach people to perform roles in any situation that requires individuals to learn WHAT to say and WHEN to say it to achieve a GOAL in a specific CULTURE. If we wanted to train bank tellers, we would simulate bank customers trying to close accounts, transfer money, etc. Instead of the tools that help students deal with language problems, e.g., recorder, we would include tools that assist bank tellers, e.g., calculator. The overall architecture remains the same.

Summarizing, RPSS systems teach people how to perform well-defined roles (e.g., bank tellers, customer representatives, receptionists, auto mechanics, immigration's officers). Dustin-like social simulations improve performance by allowing learners to play roles in authentic simulations. For example, RPSS systems can train people in every role in the St. Charles environment (e.g., receptionists, cafeteria attendants, etc.) As long as the roles involve predicable interactions, RPSS provides an unusually engaging environment. Knowledge is situated and tasks are always directed at the learner's level of competence.

What next? Ookie? No, Cookie.

One of the aspects of Dustin that I felt people enjoyed and that has great potential to be explored is object manipulation (e.g., dragging money and passport). In the Lithuanian version of Dustin, for example, before we were able to type in a response, we could understand and respond to input by manipulating objects or pushing buttons. The Total Physical Response (TPR) method of instruction (Asher, 1977) encourages the development of comprehension before production, an approach that has shown to be very effective, and interacting through object manipulation may prove very useful for this style of teaching. Besides, relative to the amount of effort involved, this type of interaction usually conveys more information than typing in natural language.

Combined with Dustin's infinite patience, this TPR approach may enable us to implement effective language learning environments for beginners. We don't go through the same learning process our mothers take us through, conversing in motherese, because nobody has the patience to do that for anyone else but his or her own children. Well, Dustin does. Using object manipulation, or even using limited speech recognition (more like noise recognition), we can have a mother say to us, "Do you want the milk or the cookie?" To which we may respond, "oooooggggiiieee," or point to the cookie. She then responds, very supportively, "ok, here's your CCCoookkkiiieee." This type of interaction is essential for learners to develop good sound discrimination, which determines later degree of competence. Interactions through manipulations, with the help of an infinitely patient mother, may enable us to explore Krashen's notion of language acquisition through exposure to comprehensible input while capturing some of the great things about mothers, the best language teaching systems around.

Conclusion

I have discussed the components, functions, and ideas that went into the design of Dustin. As I described previously, Dustin borrowed from second language acquisition research, educational research, psychology, cognitive research, and intelligent tutoring systems (see Figure 18). The difficult part of integrating all these influences lies in translating them into concrete features. It is never clear, for example, how a design principle such as, "Provide extra-linguistic means of communication," translates into interface elements. Or, even with something as simple as "Provide constructive feedback," it is not clear how and when we should do so. Should we interrupt? Should we wait until the student interacts with the tutor?

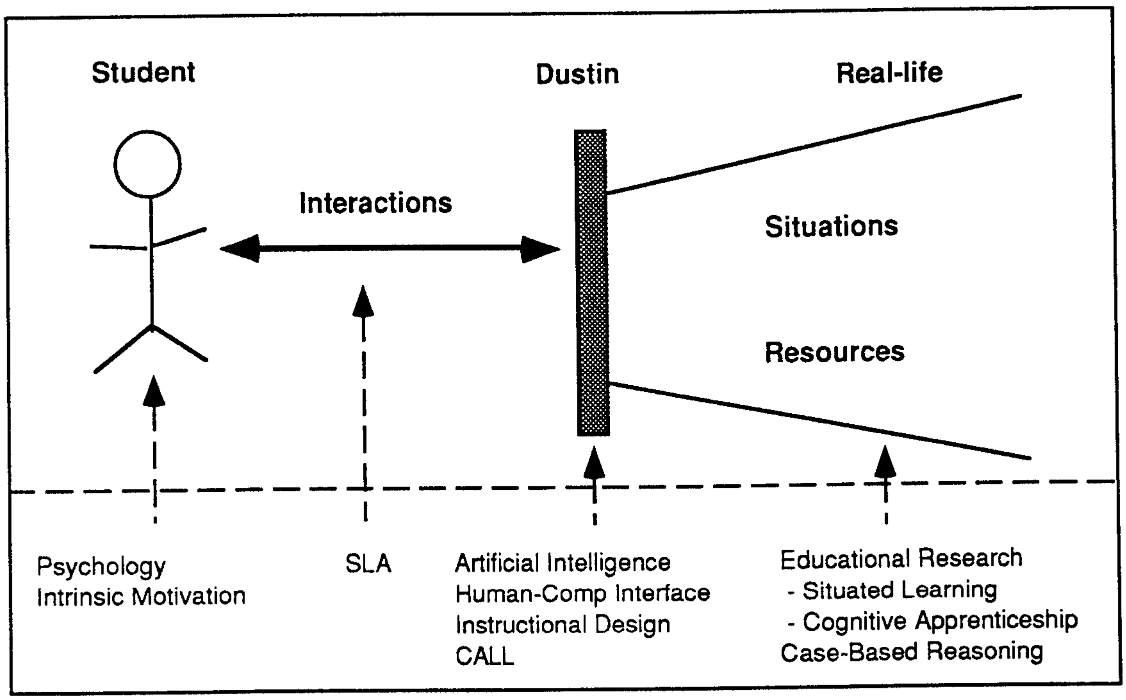


Figure 18. Elements involved in language acquisition.

Solving these problems, as far as I know, is a generate-and-test process. We cannot avoid the, sometimes painful, process of iterating through a number of versions, removing and putting back features, until they begin to converge into a graspable version of the metaphor we had in mind. During the many Dustin-iterations, we were always surprised by how unintuitive our 'intuitively obvious' solutions were, a problem that is pervasive in software design -- a problem that can only be solved, most of the time, through an obsession with listening to users.

From a developer's perspective, among the problems that I faced, the most ubiquitous one was that of having to fight the tendency, my own and of other members of the Dustin team, to be driven by technology rather than by the problem at hand. It took a conscious effort to abide by the principles we had in mind and avoid including features that would not have contributed to language learning. For example, despite our knowledge about the ineffectiveness of structural approaches, we were constantly tempted to implement games to engage students in vocabulary building and grammar exercises. The technology was so conducive to this type of solutions that they attracted us like gravitational fields, tempting us to focus on language structure. Despite our views, we would come up with word association and multiple-choice exercises that were frontally opposed to our design philosophy (i.e., experiential learning through apprenticeship, with a heavy focus on interactive skills). It was easy to see why so many systems end up perpetuating these practices. *Poker Pari*, *Dark Castle*, and even *Zarabanda* display symptoms of this tendency to be driven by technology rather than by the problem they address.

Obviously, this is not to say that implementing games is a bad idea. On the contrary, games that encourage articulation and reflection can be very useful -- I wish I had implemented some myself. However, we must be careful with the focus of these games, resisting the easy path. It is perfectly possible to develop games that explore situated knowledge (e.g., guess what's going to happen next), and games that can help students grasp important cultural and social information (e.g., Is there anything peculiar about the way Japanese people handle money?). We just need to direct this potential in the right direction, not allowing technology to push us around -- which it did for a while in the problem I describe next.

Aside from the problem of putting it all together in the RPSS architecture, the other major problem we faced was implementing Dustin. Organizing information in Dustin posed difficulties that taught us a number of lessons about knowledge representation (i.e., existing formalisms sent us in the wrong direction for a while), scaling up, contextualizing data, and reusing knowledge structures. The complexity of this problem warrants more detailed discussion, and its solution, a cognitive tool called MOPed, deserves a thorough description. That's the topic of the next chapter.

Chapter 7

MOPed - An Authoring Tool

A problem... a BIG problem

Dustin responds to a large number of buttons, organizes twenty scenarios, uses a natural language interface, and maintains a complex network of tasks including remediations and alternate paths. It organizes twenty scenarios in three ways (i.e., by storyline, goals, locations), and, in each scenario, organizes simulated dialogs, vignettes, and tutor interventions. Each simulated dialog encodes goals, speech patterns, agent responses, tutor interventions, button handlers, contextualized help messages, and default behaviors. Processing these items involves inheriting behavior, handling defaults, sequencing actions, backtracking, and handling user-initiated events. Controlling processes and organizing, understanding, and manipulating knowledge structures in a system like Dustin is a task whose complexity increases rapidly with size. Without adequate tools, implementing only a couple of Dustin-like scenarios may still be attainable, but scaling up to twenty scenarios, as I had to do, is a nightmare.

In early implementations, Dustin encoded information in structures like the one shown in Figure 19 -- don't try to understand it, I only show it here to exemplify obtuse representations. This structure, which organized part of the reception scene, contains tutor messages, some help messages, and the steps in the simulated dialog (i.e., the steps in a plan). It also contains links that interlace it with other structures. With forty tasks to arrange and link, it was extremely hard to visualize where a task fit in the larger picture.

```
(in-package "USER")

(SOF SCE-REC-CHECK-IN
 :NAME SCE-REC-CHECK-IN
 :DESC "Check in"
 :REQUIRED-P T
 :WATCH-TEXT "Watch John Harrison check in"
 :WATCH-SCRIPT SCE-REC-CHECK-IN
 :NEXT-TASK (TASK OPT SCE-REC-CHECK-IN)
 :DO-TEXT "Go to the reception desk and check in"
 :DO-AGENT A-KELLEY
 :SUCC-TEXT "Good! Now, go to your room and meet Scott."
 :SUCC-TASK (TASK DO SCE-ROO-MEET)
 :FAIL-TEXT "Let's watch something simpler"
 :FAIL-TASK (TASK ?COMPUTE ?NEXT)
 :OPT-TEXT "Can you check in at the reception desk?"
 :OPT-YES-TASK (TASK DO SCE-REC-CHECK-IN)
 :OPT-NO-TASK (TASK ?COMPUTE ?NEXT)
 :GOAL-TEXT "Try to check in"
 :GOAL-OK-TEXT "You've already checked in. Leave"
 :GOAL-TEST (DONE-P D-REC-CHECKIN)
 :GOAL-SAY ""
 :EXAMPLES NIL
 :PLAN (SCE-REC-GREET SCE-REC-PURPOSE SCE-REC-NAME SCE-REC-PACKAGE SCE-REC-THANK)
 :DIALOGS "d-rec-checkin")
```

Figure 19. Reception scenario described in incomprehensible language.

This cryptic syntax filled all levels of representation in Dustin. To run dialogs, Dustin parsed input into an elaborate internal representation then used it to trigger rules like the ones shown in Figure 20. Updating large sets of rules in a rule-based system, as many knowledge engineers can attest to, can be very confusing. In the case of Dustin's dialogs, even going from three simulated dialogs to just twice as many was a frustrating effort. Dustin encoded dialogs in if-then rules that said: "If he asks you out and you have heard that before, then get upset and say that you are very busy," or "If you ask him how long he'll be in the country and he says more than one year, then negate interlocutor's goal" or "If you ask him how long he'll be in the country and he says less than a year, then confirm interlocutor's goal." The problem with this type of representation is that it gets very confusing very fast, and, sometimes, adding twenty new rules has unpredictable effects. I was confronting a major problem: We didn't understand our own data. We couldn't understand, we couldn't organize, and we couldn't maintain the knowledge structures necessary to implement Dustin. We needed a better way of representing this data.

```
;;; Rules for CASHIERS
;;; ------------------
;;; (clear-agent-rules a-cashier)

(rule "Hi, object please ---> ATRANS & Anything else?"
      a-cashier `(,c-inf-obj-sta ,c-req-obj)
      :prec t
      :exec ((tell ?other ?self c-que-obj-nee-mor)
            (atrans ?other ?self (second (agent-just-heard ?self)))))

(rule "Objects, please ---> ATRANS & Anything else?"
      a-cashier c-req-obj
      :prec t
      :exec ((tell ?other ?self c-que-obj-nee-mor)
            (atrans ?other ?self (agent-just-heard ?self))))

(rule "Objects ---> ATRANS & Anything else?"
      a-cashier c-obj
      :prec t
      :exec ((tell ?other ?self c-que-originate-nee-mor)
            (atrans ?other ?self (agent-just-heard ?self))))

(rule "How much is x? ---> $"
      a-cashier c-que-obj-pri
      :prec t
      :exec ((tell ?other ?self c-inf-obj-pri)))

(rule "Hi, how much is x? ---> $"
      a-cashier `(,c-inf-obj-sta ,c-que-obj-pri)
      :prec t
      :exec ((tell ?other ?self c-inf-obj-pri)))
```

Figure 20. Rule-base for dialogs (Absurd!)

A Solution (MOPed)

The solution evolved from combining the (1) need for a uniform representation with a simple (2) visual artifact and a (3) model of memory organization borrowed from artificial intelligence research.

A Uniform Representation

First, although different processes in Dustin used different formalisms (e.g., rules for dialogs, frame-like structures for tasks, and plans for sequences of utterances), all levels of knowledge representation shared characteristics. Common patterns appeared everywhere. For instance, students make mistakes when interacting with agents, and these mistakes are similar across levels. At the word level, students misspell words. They misplace, omit, transpose, and add unnecessary characters. Similarly, at the sentence level, they misconstruct sentences by misplacing, omitting, transposing, and adding unnecessary words. Moreover, the same error pattern appears at the dialog and even the scenario levels.

These common patterns, however, were not restricted to errors. Two other recurring patterns involved inheritance of behavior and reuse of knowledge structures. Inheritance of behavior occurred in handling both dialogs and help messages. In a dialog, if a rule wasn't available for, say, receptionists, Dustin would inherit rules from the set of rules for humans -- following an ISA hierarchy. Similarly, when trying to help the student, Dustin would inherit help from higher levels of abstraction if help was not available at the current level -- following another hierarchy. Although their representation schemes differed, both cases shared the same mechanism. The reuse of structures, on the other hand, occurred everywhere. For example, the same speech pattern (e.g., ways of saying thanks: "thanks," "thank you," "I appreciate that") could appear in a number of dialogs. When the student says thanks after checking in, for example, the receptionist says "You're welcome." However, if he says it before even greeting her, the tutor takes over, saying "you should greet her first." These are two completely different outcomes based on the same speech pattern. In such cases, we shouldn't have to recreate the same structure to handle each one, and Dustin frequently required reusing existing structures. These similarities across levels of representation seemed to suggest that a single formalism could represent all processes in Dustin.

A Visual Artifact

Second, we stumbled upon a very simple, yet very powerful, artifact. Two members of the team working on Dustin went to St. Charles to collect data. They visited 25 locations and identified 45 tasks that students commonly face in St. Charles. When they returned from St. Charles, I asked them to explain to me how those 45 tasks occur in an employee's typical day. Their explanation sounded like this:

They can go to the cleaners early in the morning before breakfast, or they'll have to wait until the second break in the afternoon. They go to the cafeteria for breakfast, lunch, and dinner, but they can also go to the Four Corners Cafe, but the cafe is closed from 2PM until 6PM...

After listening for ten minutes, I couldn't understand what they were saying anymore. At that point, we resorted to a simple artifact. I asked them to take a big sheet of paper, a stack of post-its (i.e., sticky pieces of paper), plus drawing material, and create a visual representation of what they were trying to say. They came up with a much more comprehensible graph that looked like the one in Figure 21.

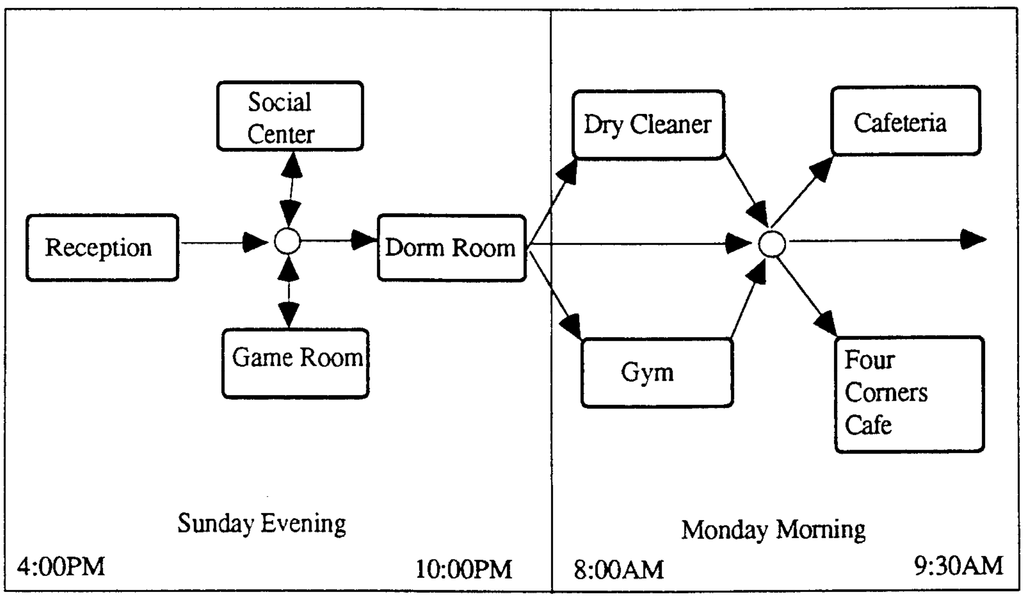


Figure 21. St. Charles - A big sheet of paper with post-its.

Since using paper, post-its, and pen made it so easy to understand tasks in St. Charles, it seemed that the same idea could help us understand the rules in the dialogs, the interactions between the tasks, the tutor interventions, and so on. It seemed obvious that if we could use paper, post-its, and a pen to create Dustin, we'd be home free, almost.

MOP - Memory Organization Packet

Third, a knowledge structure called MOP, Memory Organization Packet, suggested by Schank as a basic unit in a model of human memory, provided the building block for a powerful representation scheme and manipulation tool (Schank, 1982). Building on this basic unit, the MOP, we developed a tool that made it possible not only to implement and use the visual aid described above at all levels of representation in Dustin but also to address the problems of inheritance and reuse of knowledge structures. Furthermore, it provided an excellent mechanism to contextualize information, encode predictions, and organize hierarchies.

To understand how MOPs served as a building block and to understand a few important ideas behind MOPs, we have to go back to the notion of scripts introduced by Schank and Abelson (1977). A script is a structure that describes an appropriate sequence of events in a particular context. For example, when we go to a restaurant we follow a script: we sit, read the menu, select, order, eat, pay, say thanks, and leave. Knowing the restaurant script helps us understand events such as the waitress' giving us a bill and saying "I'll take it when you're ready." In addition, it helps us know that following that we should pay, say thanks and leave.

The next logical question following the work on scripts was: How are scripts stored in memory? Schank proposed an answer to this question in a model called Dynamic Memory (Schank, 82). The Dynamic Memory model introduces structures (i.e., MOPs, TOPs, meta-MOPs) and processes that account for understanding, reminding, cross-contextual reminding, indexing of information, and learning. Here, I will concentrate on MOPs.

Now, to understand the main problem that MOPs address, let's go back to the saying-thanks example mentioned earlier. There are hundreds of situations in which we need to express gratitude. In the restaurant script, for example, we say thanks after we pay the bill, when we are getting ready to leave. When we do so, where is the knowledge about ways of saying thanks coming from? Where is it stored? Assuming that we know a dozen or so ways of expressing gratitude (e.g., thanks, thank you, thanks a lot, thank you very much, I appreciate that, muchas gracias, etc.), would these be stored with the restaurant script? Is it the case that with each script we have stored in memory (e.g., dentist, registering for classes, going to the movies) we store a separate instance of the saying-thanks information? Most likely not. We can expect that what we know about saying thanks may apply in numerous other circumstances, and memory would be too redundant if it stored recurring information separately for each situation. Therefore, it must be the case, proposes Schank, that these large scripts get broken down into smaller sharable units (Schank, 1982). So the restaurant script is broken up into distinct pieces (i.e., the ordering part, the eating part, the paying part, and the saying-thanks part). And here is the most important corollary of this. "There must be some memory structures available whose job is to connect other memory structures together." This is the key to the MOP-based memory.

I'll show exactly how I implement MOPs shortly, but, for now, let me elaborate on the nature of MOPs. The reasoning behind its existence goes more or less like this: (a) memories are shared, (b) scripts use shared memories, (c) there must be a memory structure that connects other memories. Schank proposes that "memory structures must exist that tie other structures together in the proper order." This means that the structure that stores information about going to a restaurant doesn't really store the detailed information itself. Instead, it simply points to other memory structures that hold the detailed information. In the dynamic memory terminology, "information about how memory structures are ordinarily linked in frequently occurring combinations is held in a *memory organization packet.*" (Schank, 82). The only function of the MOP structure, therefore, is to organize other memory structures. As I will show next, MOPs help us reuse knowledge structures and contextualize them under different higher-level structures.

In Dustin, MOPs are like big sheets of paper on which we stick post-its and draw connections. As implemented here, MOP is a graph, but, as I'll show later, a special kind of graph. MOPs contain special types of nodes, or post-its, handle defaults, allow inheritance, use spatial coordinates to encode priorities, order processes, allow multiple entry and exit points, and aid comprehension with visual representations. MOPs organize knowledge and add visual information that makes it comprehensible to developers. The tool that updates these memory units, MOP editor (MOPed), enabled us to implement

Dustin by helping visualize, and manipulate the information involved in Dustin's twenty scenarios. Next, I'll describe this tool, MOPed.

MOPed - A Demo

To make it easier to understand MOPs and the editor, I'll describe how MOPs implement parts of the Immigration scene in Dustin. This will provide the necessary context for the explanation that follows. Figure 22 shows the MOP, the big sheet of paper with post-its, that organizes the events at the O'Hare International Airport.

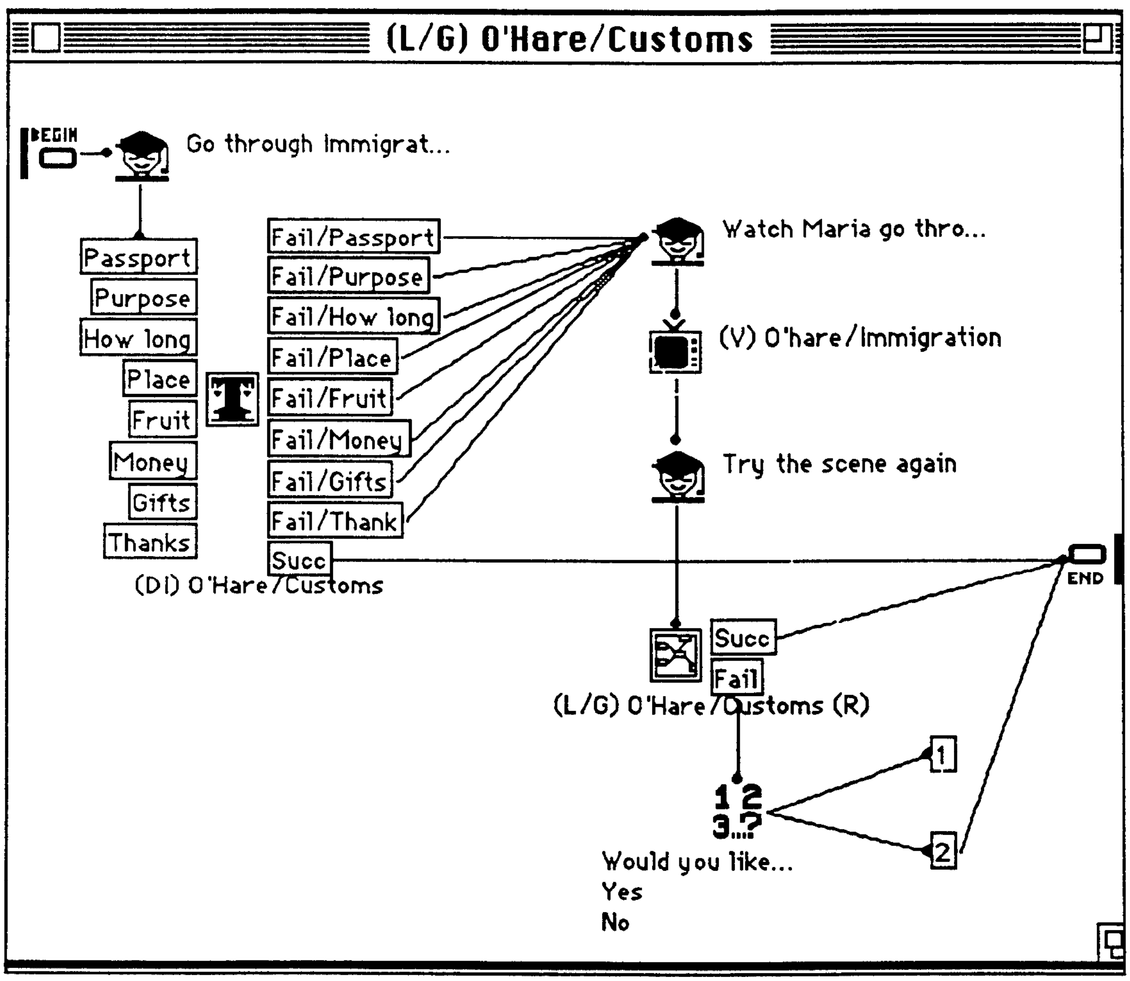


Figure 22. Going through immigrations at O'Hare International Airport

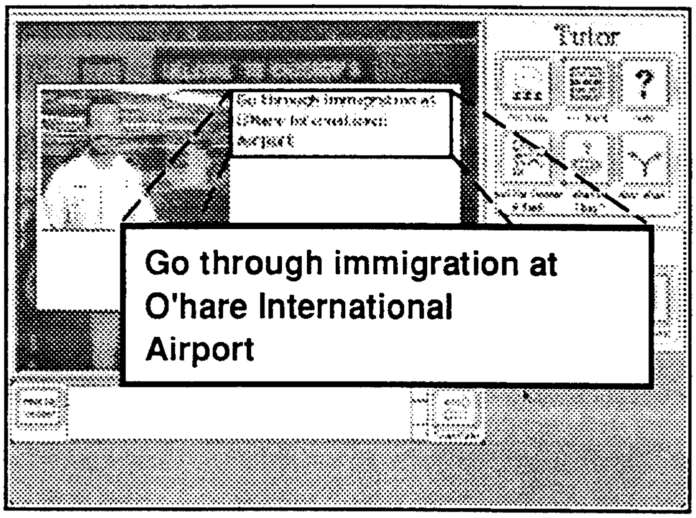
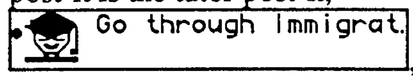


Figure 23. The tutor box appears on the screen.

Immediately after the BEGIN post-it is the tutor post-it,



, the second symbol in Figure 22. When executed, this post-it makes the tutor box appear on the screen, as shown in Figure 23 below.

When the student presses OK, in response to the tutor message, the dialog with the immigration agent starts. Figure 24 shows the MOP representing the dialog.

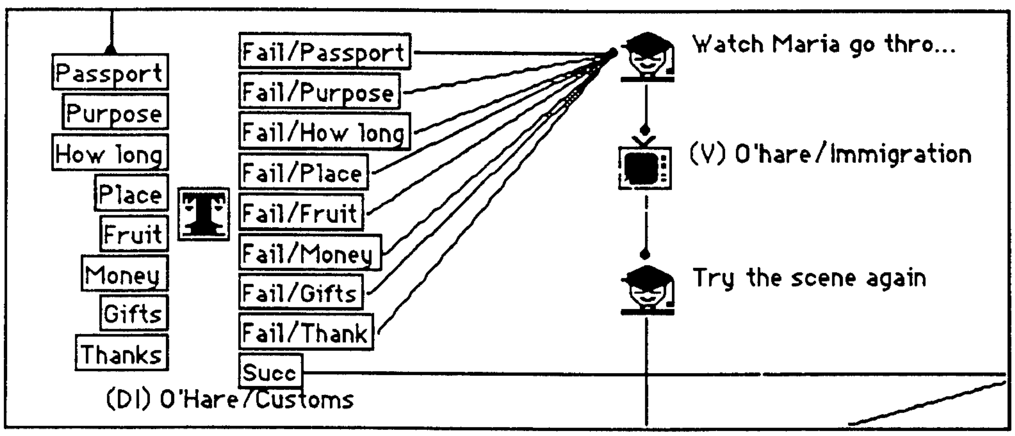


Figure 24. Failing at first try, the tutor takes over.

This dialog, which I'll explain later, has a number of entry and exit points. If the student doesn't know what to do, or fails to perform the task, the dialog aborts, and the tutor takes over.

If the student presses the "Too Hard" button, the tutor says: "Watch Maria go through immigrations," and Dustin shows a video of Maria at the immigration (see Figure 25).

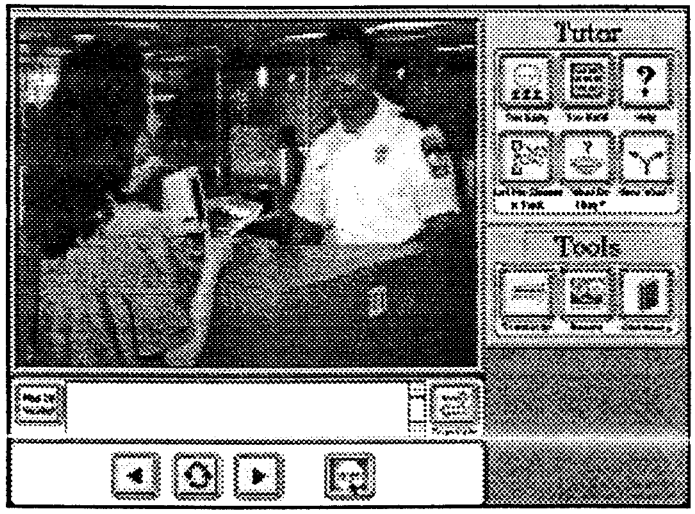


Figure 25. Watching Maria go through immigrations.

MOPed, the MOP editor, offers two advantages: (1) ease of change and (2) immediate activation. For example, suppose I didn't like this sequence of events and wanted the tutor to show the example first and then throw the student into the simulation.

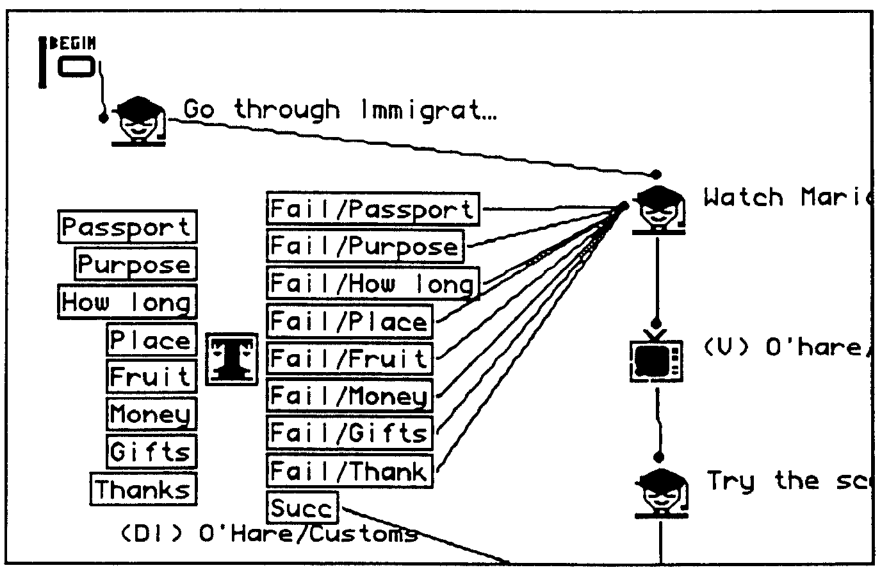


Figure 26. Modified MOP - Skips the dialog and starts with a video.

I implement this change by simply disconnecting the tutor post-it from the dialog post-it and then connecting it to the second tutor (using Macintosh-like point and drag actions). Figure 26 shows the updated MOP. Now, if I execute this MOP, Dustin shows two tutor messages, and then plays the video of Maria at O'Hare.

Let's look at the dialog MOP. Double-clicking on the node, the one with a number of entry and exit points, opens up a new window with that MOP. This happens because the dialog itself is another MOP, another big sheet of paper. Embedded MOPs are visually identifiable by a frame surrounding their icons (e.g., dialog MOP, generic MOP, scenario MOP). Figure 27 shows the first three steps in the immigration MOP with the subdialogs (1) passport, (2) purpose, and (3) how long. Note how entry points lead directly to one of the subdialogs, and how exit points connect to the failure of a specific subdialog.

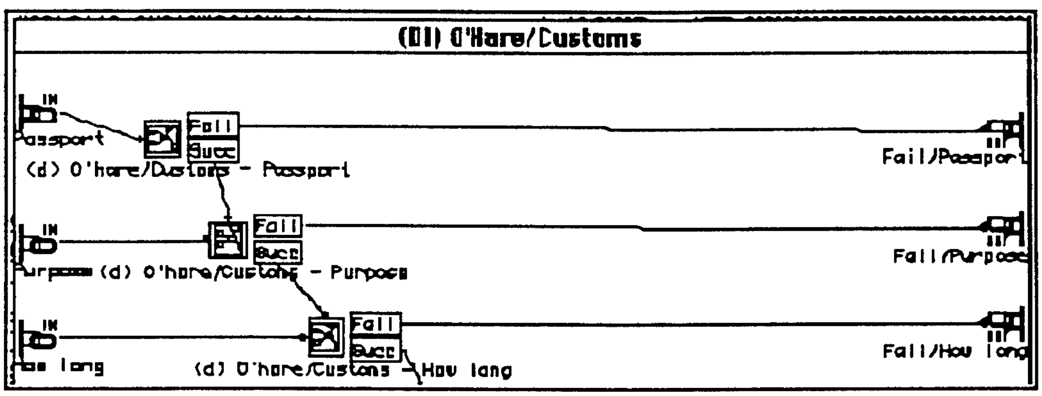


Figure 27. The upper portion of the "(DI) O'Hare/Customs" MOP.

Let's look at the third subdialog, "(d) O'Hare/Customs - How long," in which the agent asks the student how long he'll be in the United States. Remember, double-clicking on the icon opens up a window with the MOP. Figure 28 shows the "How long" MOP.

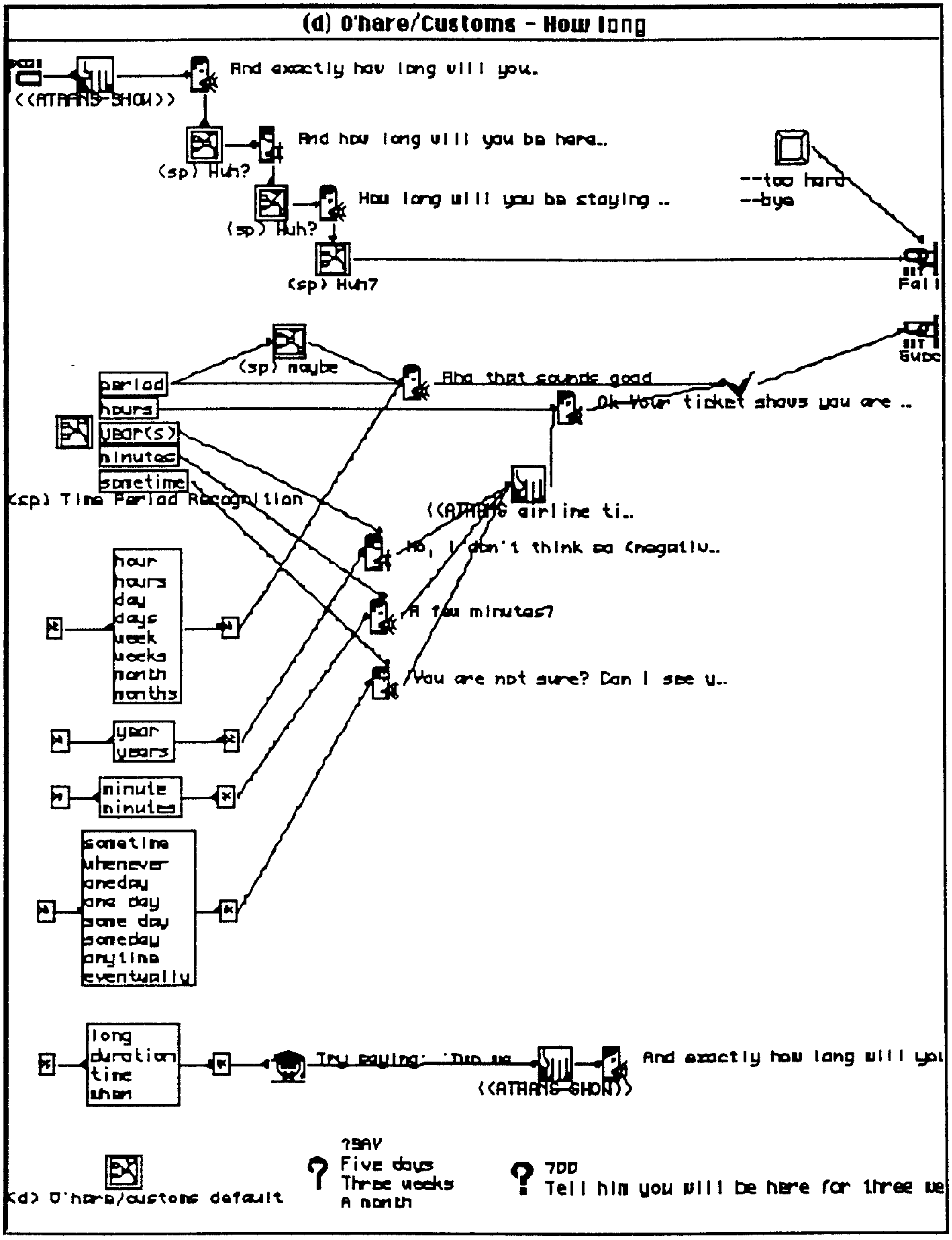
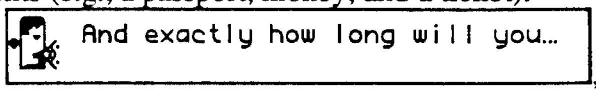


Figure 28. The "How long will you be in the US?" subdialog.

This mop, Figure 28, incorporates a number of features. The first post-it, the hand icon , makes the transfer window appear on the screen, the window with the passport (see Chapter 5). This is how Dustin activates windows, agents, and tutor messages. A post-it like this one contains a message for the object that it represents graphically. For example, the hand icon contains a message, "(atrans-show)," for the TRANSFER object which it represents. Upon receiving the message, the TRANSFER object opens up a window with its defaults contents (e.g., a passport, money, and a ticket).

The next post-it, 

, makes the agent appear on the screen and say "And exactly how long will you be in the US?" The way this works is as follows. The message this post-it holds is the name of a video clip. The AGENT object receives this message, activates the video screen, seeks the video clip, and plays that clip on the center screen.

HUH?

Now, suppose the student pushes the HUH? button, or says something like: "What did you say?" Figure 29 shows the patterns in the "(sp) Huh?" MOP. Figure 30 shows the sequence of responses to a series of similar inputs, inputs defined by the patterns shown in Figure 31. For instance, at this point, "I don't understand you," makes the agent say "And how long will you be here?"

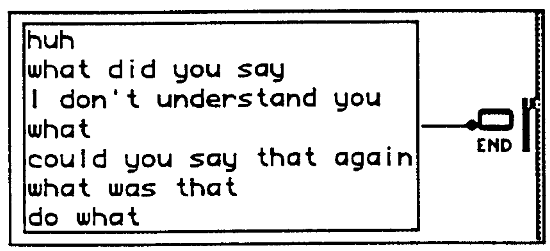


Figure 29. Patterns in the HUH? MOP.

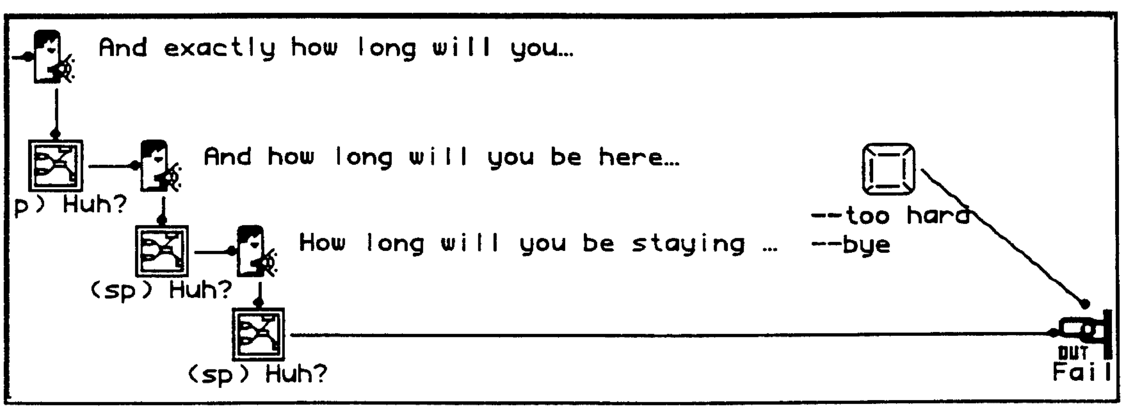


Figure 30. Responding to a series of HUH?s

Input Patterns

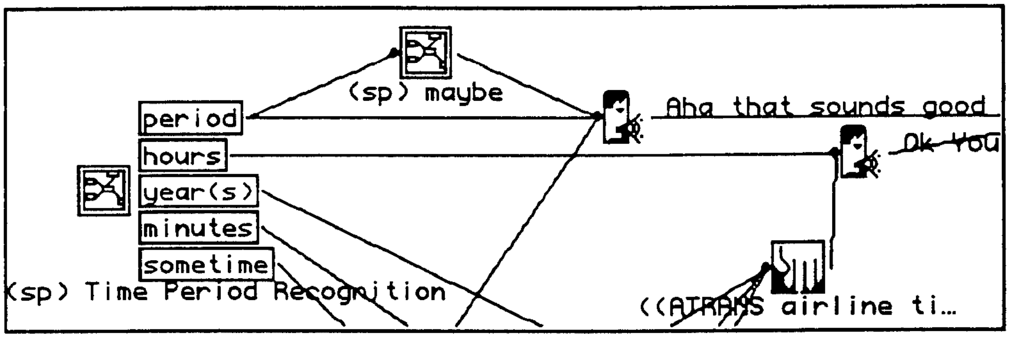


Figure 31. Time period MOP

If instead of saying "Huh?" the student enters some time period, recognizable by the patterns in the MOP "(sp) Time Period Recognition," the agent says "Aha, that sounds good." The post-it pointing to this MOP appears in Figure 31, connected to the agent's response. The MOP itself, with the patterns acceptable in this exchange, appears in Figure 32.

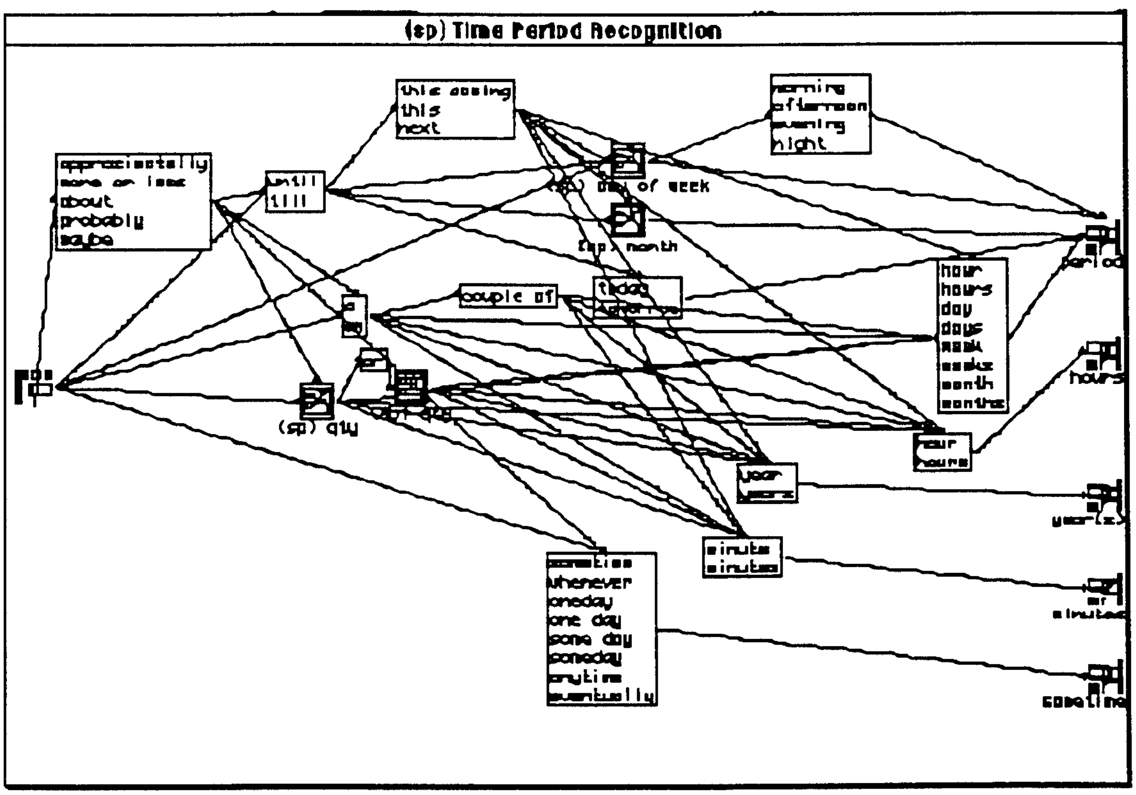


Figure 32. The speech patterns that are acceptable.

Keywords

Immediately below the time period MOP, there are boxes preceded and followed by "*." (see Figure 33). They hold keywords. If the previous structures can't handle the input, Dustin tries to find these keywords. The pattern shown here, for example, catches any sentence containing one or more of the words in the box. So, saying "I think I'll stay a week or two," would make the agent say: "Aha, that sounds good." Note that saying "I do zero to sixty in a day," would too. The developer should take the blame, not the tool.

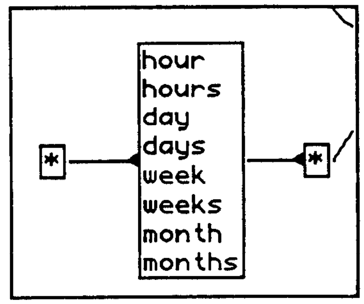


Figure 33. Keywords

Tutor Interventions

In some cases, the tutor intervenes. For instance, if the student says something about time or duration that is too complex for the system's predictions (e.g., "My boss didn't tell me how long"), the tutor takes over and suggests a simpler way of saying it, "Try saying: two weeks." Figure 34 shows how MOPed encodes this reaction.

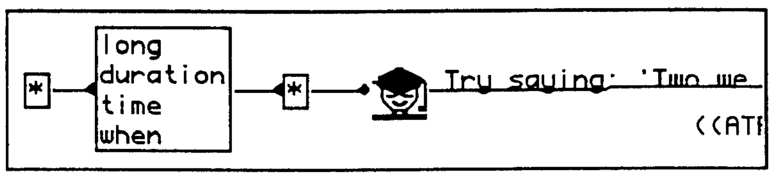


Figure 34. Encoding tutor interventions.

Handling Default

If nothing described so far parses the input in this subdialog, Dustin uses a special message parser, "DEFAULT," that swallows any input (see Figure 35). This handler sits at the end of the subdialog. The contents of the default MOP appears below. So, saying "Your mother is ugly," would make the agent say: "Pardon me?"

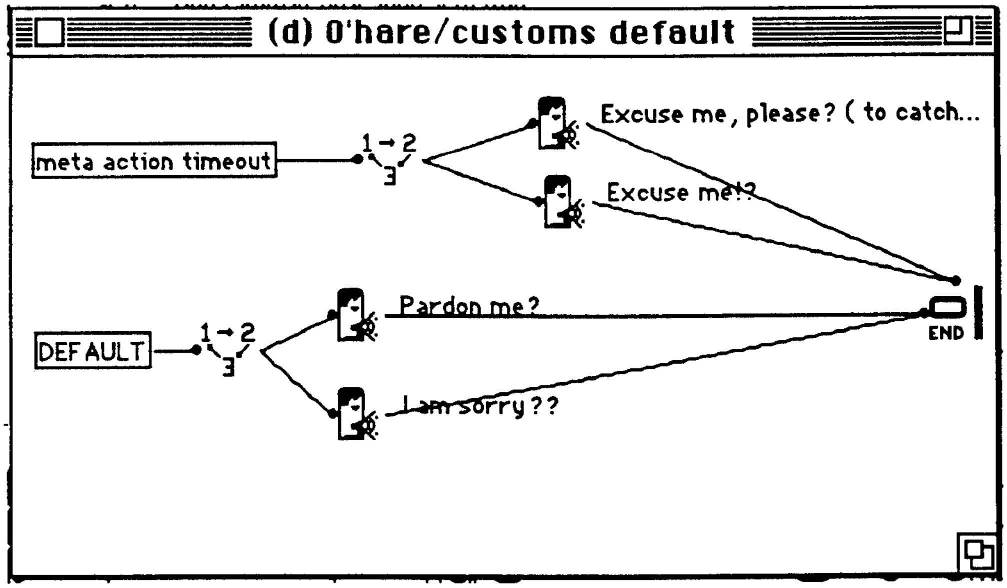


Figure 35. The default handling MOP

Help Messages

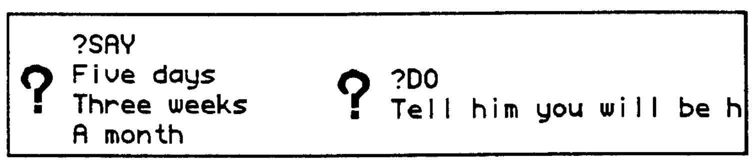


Figure 36. Help messages in a MOP.

At the bottom of Figure 28, there are some help messages. Help messages are context-dependent and MOPs help to contextualize them. The entries in the "How long" MOP hold information for two buttons: (1) "What do I say?" and (2) "What do I do?" (see Figure 36).

Button Handler

Finally, MOPs also handle buttons. Button handlers that appear in the "How long?" MOP take care of the "Too Hard" and "Bye" buttons (see Figure 37).

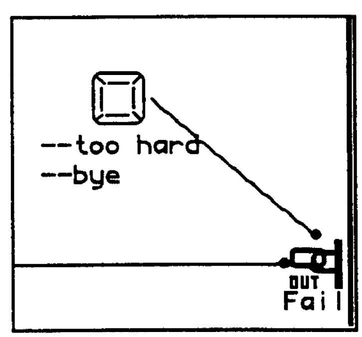


Figure 37. Button Handler.

This is the basic scheme used to encode knowledge in Dustin. The immigration module, actually called "O'Hare Customs," described here appears in a higher level MOP (see Figure 38) that contains all the scenarios for Sunday, showing, yet again, how MOPs encode data at all levels of Dustin.

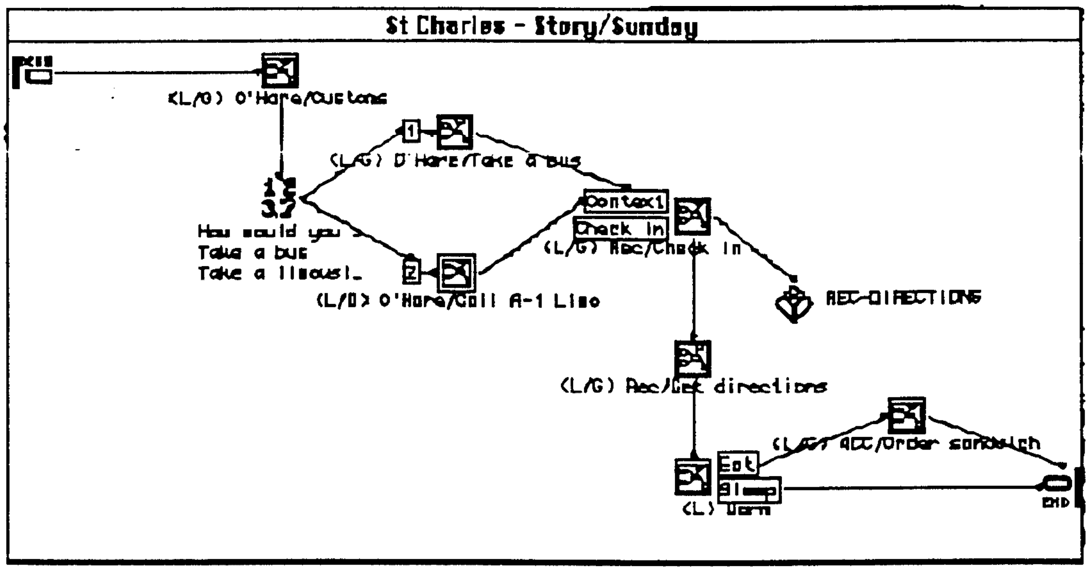


Figure 38. The higher level MOP - Sunday in St. Charles.

Where does MOPed fit in?

Dustin has three components: a data base of (1) memory structures, MOP-based memory, a user (2) interface, and an (3) engine that runs the interface based on the data in memory. The interface includes windows, buttons, and tools such as the dictionary and transcript. These interface units are objects that send and receive messages organized by the MOPs. Some of these objects process their messages and then forward them to other objects (e.g., AGENT object), thus having side-effects; others generate messages (e.g., INPUT-BOX, where the student types his responses). The AGENT object, for example, takes a video clip as a message, displays the corresponding video on the screen, and then sends the text of the clip to the TRANSCRIPT object. The TRANSCRIPT object simply adds the text received to its list of sentences. Sending messages the other way, the INPUT-

BOX, takes the student's input, checks it for spelling, and then sends it as a message to the MOP engine. The engine uses the MOP-based memory to process the message. Similarly, buttons also send messages to the MOP. Unlike the objects just described, some objects exchange messages without the mediation of the engine. A double-click on a word in any window, for example, sends a message to the DICTIONARY, which looks up that word and displays it on the screen.

The role of MOPed, the MOP editor, is to enable users to visualize, create, and manipulate a MOP-based memory. Figure 39 shows how MOPed relates to the components described above. When creating learning environments, there are tools that enable us to manipulate these components (i.e., memory, interface, and engine). In Dustin, the Lisp environment provides the tools to create the engine and the interface. MOPed provides the tools to create and maintain the MOP-based memory.

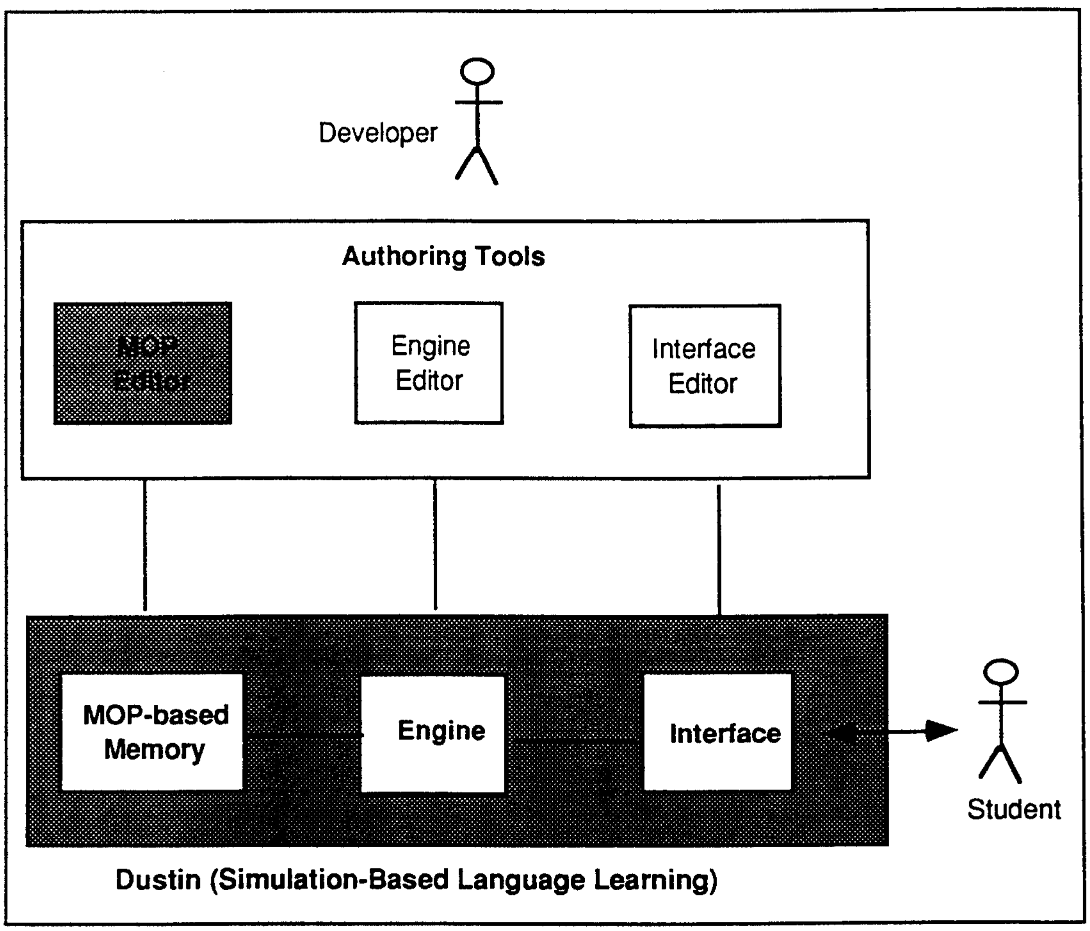


Figure 39. Authoring tools for RPSS systems

What's MOPed?

Besides being a way of organizing and maintaining knowledge, MOPed is an artifact that helps developers understand complex networks of information. It accomplishes that by providing visual aids to improve readability and by grouping information to help contextualize data. In an environment based on objects that process messages, MOPed organizes the messages that objects exchange. It is a message organizer that handles defaults, inheritance, backtracking, and that supports data-sharing. While other tools define objects and how they handle messages (e.g., Smalltalk, CLOS), MOPed focuses on the messages they send to each other. By providing a visual representation of how messages interconnect, MOPed reifies interactions, making explicit the behavior of objects across time.

A MOP, the basic unit in the MOPed scheme, is like a sheet of paper. It holds symbols and even other sheets, or MOPs, combining and organizing them according to certain conventions. One of these conventions is that MOPs have markers, post-its, showing where they begin and where they end (i.e., BEGIN , END). When processing a MOP, the engine starts at the BEGIN post-it and either follows arcs or searches from left to right and from top to bottom until it reaches an END post-it. MOPs can also have entry, IN, and exit post-its, OUT. The difference between these and BEGIN/END is that they affect the way MOPs appear inside other MOPs. For instance, a MOP with two exit points, OUT, one labeled "fail" and another labeled "succ," looks like this, Fail Succ, inside another MOP. This shows how the same MOP acts differently inside other MOPs. Its behavior depends on what connects to its entry and exit points. BEGIN, END, IN, OUT, and the MOP post-it, , which includes another MOP inside a MOP and whose appearance varies with the number of entry and exit points are the most idiosyncratic post-its in MOPed.

All other post-its simply contain messages for the objects that they symbolize: - Video, simulated agent, flag setter, flag tester, randomizer, and tutor.

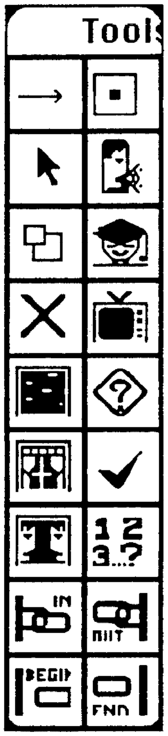


Figure. 40. Palette

When creating a MOP, we simply choose post-its from the palette shown in Figure 40 and place them anywhere inside the MOP (see Figure 41). We can also copy and paste parts of other MOPs, using existing templates to create new ones. To determine processing order, we either create arcs between post-its or place them up or down in the MOP, keeping in mind that processing proceeds from left to right and from top to bottom.

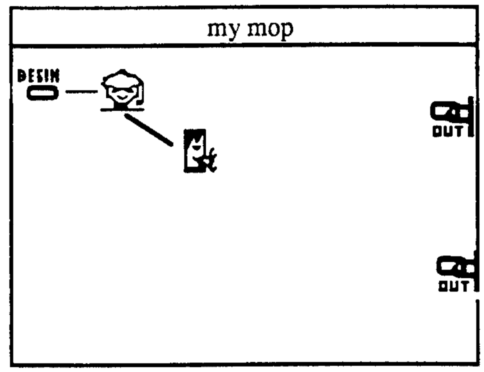


Figure 41. Creating a MOP

As just described, one useful feature of MOPed is that it supports embedded MOPs, which helps group and contextualize information. Another important feature is that we can reuse MOPs without recreating them. A post-it that stands for a MOP simply points to a real MOP; its only content is the name of the real thing. For example, suppose we create a MOP that encodes what the immigration's agent says when he doesn't understand an input. See Figure 42.

In the course of a conversation, whether the agent is asking for the passport, the extent of her stay, or telling her what to do next, if the student says something incomprehensible, like "Capoeria e' de matar," the agent will react the same way, e.g., by saying "Pardon me?" So, whenever he doesn't understand an input, regardless of the context, his reaction is the same. Therefore, we shouldn't have to recreate the structure that encodes this behavior, Figure 42, for each subdialog. We don't. The default handling MOP, "(d) O'Hare/Customs Default," appears in four different subdialogs -- Figure 28 shows one of these subdialogs. Of course, this also holds for speech patterns, or other dialogs which reappear in other MOPs. Reiterating, MOPs are like big sheets with information, and whenever we use a MOP inside another, we simply leave a post-it saying "here we use the Customs/Default MOP." At run time, the engine accesses the real thing, does the job, and releases it for further use. This allows Dustin to use the same structure without duplicating it.

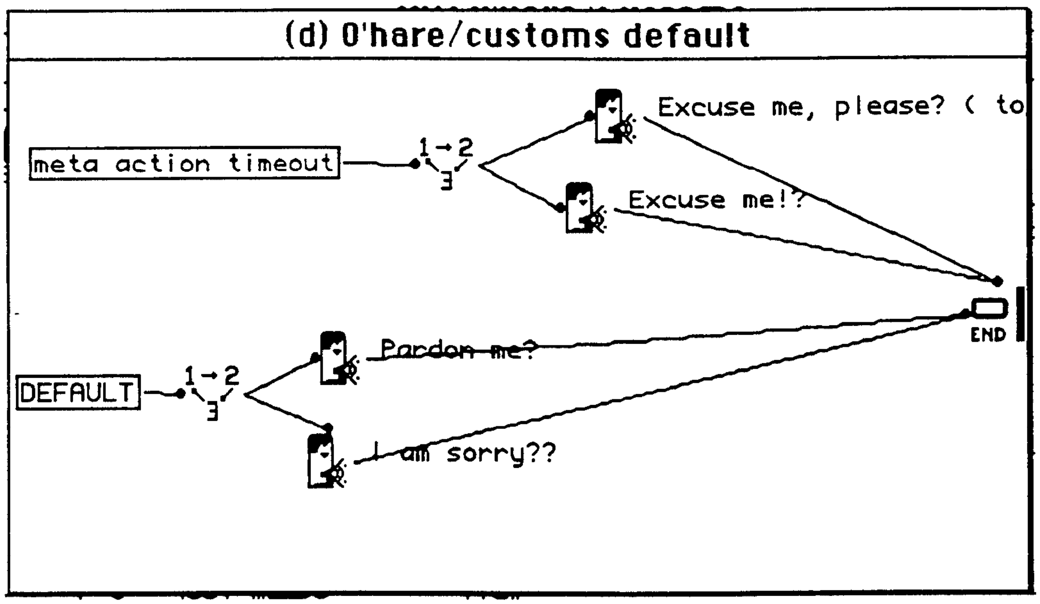


Figure 42. A MOP that is reused in four other MOPs

This scheme organizes knowledge at all levels of Dustin. From scenarios down to speech patterns, including tutor messages, simulations, button handlers, help messages, and examples, everything is organized by MOPs. Button handling and help messages are contextualized through the embedding mechanism, which also supports another important feature. If the current MOP doesn't have a handler for a button, the system tries to inherit the handler from the MOP above it. More precisely, it goes up the run-time hierarchy in which that MOP appears until it finds a handler for that button press.

What's MOPed good for?

When creating a sizable system involving intricate interactions between agents, contextualized processing of buttons, help, and input, with information organized in complex sequences and hierarchies, we need tools to help understand the data. Scaling up without them is impossible.

As a visualization tool, MOPed helps show how elements of the system interact. Even the simple if-then rules used in dialogs, which associate input patterns to output, are much easier to understand when expressed as a graph with meaningful icons and grouped into embedded units. Icons make it easy to determine who is doing what at any point in the system. They help show how and when a help message becomes active and when a button handler is used. By looking at how data is spatially organized inside a MOP, I can quickly determine the order in which the system parses input, what patterns it accepts, what keywords it looks for, and exactly in what order it does that. Furthermore, I can readily grasp how a particular subdialog fits into a larger context, including the consequences of failing to perform that subdialog. At the scenario level, it gives me immediate information on what scenarios are available and how scenes and tasks relate to each other. As a visualization tool, MOPed expands my ability to grasp and understand complex knowledge structures.

As an organization tool, MOPed helps to contextualize information. Consider the following situation. When the student types "Hello, how are you?" the engine tries to find post-its in the currently active MOP that parse that message. If successful, it triggers a response (e.g., the agent says "Fine thanks"). This way of handling messages is analogous to using an if-then rule; if input is x, do y. However, the advantage MOPed offers over common rule-based systems is the contextualization of rules. Humans use predictions to understand what they hear, and MOPs encode these predictions. When parsing an input, only those rules that are relevant in that context are active. They are based on the predictions encoded in the MOPs. This contextualization mechanism not only constrains search but also allows the system to move up the hierarchy to find handlers for input that the current MOP can't handle, a search upwards that is also always contextualized by higher level structures. Since we don't have to consider all the rules in the system, the search space is drastically reduced. Moreover, since every event in Dustin is a message, search space is reduced across the board. Contextualization has helped encode authentic reactions, and its constraining the search space has enabled realistic response times -- simulated agents never take more than a second to respond to the student.

As a manipulation tool, MOPed simplifies the task of creating complex information networks by reusing existing structures and copying parts of existing templates. Once a lesson is developed, it can be copied and used elsewhere. In fact, all scenarios in Dustin derive from a single template. Furthermore, MOPed assists in handling these structures. A specialized editor lets the developer update structures and provides searching tools for finding relevant constructs. As opposed to the changes in earlier versions of Dustin, changes with MOPed become immediately active. Any modification made in any MOP has immediate effect. Combined with the reuse and template capabilities, this dynamic updating saves considerable coding time. Additionally, the use of entry and exit points in MOPs makes it possible to tailor shared structures to particular situations, making it easier to create generic structures.

At its core, MOPed is very simple. It is an editor that allows us to create, link, and store big sheets of paper with post-its. Post-its may point to MOPs themselves or contain messages. Post-its contain only three slots of information: content, type, and position. The type determines how the engine handles it, and that can differ significantly. However, as far as the editor is concerned, there are only six types of post-its: (1) LEAF, (2) IN, (3) OUT, (4) BEGIN, (5) END, and (6) META. Meta post-its simply point to other MOPs. On top of these six basic types, the developer can define any number of post-its. This is where MOPed's flexibility lies. The user can define not only his own post-its but also associate methods to handle them and icons to represent them. He can create a cash register post-it, for instance, that will display a simulated register on the screen. For each new post-it, he can assign a symbol, an icon, and simply map it onto one of the basic types (e.g., leaf, in/out, begin, end, meta).

Conclusion

MOPed needs improvements. To be a complete authoring environment, MOPed must integrate tools that help developers build interfaces and object behaviors. As a knowledge manager, MOPed can improve the way it handles data. The current implementation loads all data in memory, making changes during a session immediately activated. However, at the end of a session, changes have to be saved so that they can be reloaded in the next session. Storing data on disk makes this step unnecessary. As a knowledge manipulation tool, the editor needs to be extended to help define classes of objects and their behaviors, and maybe include some predefined methods. Finally, as a processing mechanism, the

MOPed engine needs standard processes, automatically incorporating the common processes we identify as we develop more simulations.

Unlike other authoring tools, MOPed focuses on knowledge organization. Authorware, for example, provides tools for interface design; MOPed is not concerned with interface. While Authorware has predefined post-its with predefined behaviors, and allows users to order them in predefined patterns, MOPed allows the user to define his own post-its, define his own methods, and organize interactions between them according to powerful structuring mechanisms. Although powerful in other ways, Authorware is weak as a message organizer. It does not allow reuse of data structures, does not handle inheritance, and, compared to MOPed, has inflexible ways of handling events. It provides good visualization and manipulation capabilities, but lacks the representational power to encode complex social simulations. For instance, storing speech patterns with defaults and inheritance in Authorware is nearly impossible. While Authorware's advantage lies in the predefined post-its it provides, MOPed's advantage lies in the power it offers to organize post-its. Regarding other formalisms (e.g., ATN, directed graphs, finite-state-automata), it could be said that MOP-based memory is reducible to some of these formalisms. It is certainly a graph, but, as I mentioned earlier, a special kind of graph. It implements the big-sheet-with-post-its metaphor, providing a mechanism to contextualize information, reuse knowledge structures, reduce search space, and encode temporal and causal relations. It packages data in abstraction hierarchies, and encodes complex interactions between memory units. MOPed reifies interactions between objects, helping visualize interactions. But the most important thing about MOPed is that it is more than a knowledge representation scheme. MOPed is a knowledge-manipulation environment, a cognitive artifact whose strength lies in its power to articulate knowledge that is otherwise difficult to understand, and in the way it allows the user to manipulate that knowledge.

Chapter 8

Conclusion

What's important?

One important contribution of Dustin is that it addresses those problems afflicting classroom language instruction. Students need (1) interactions with native speakers, (2) exposure to the target culture, and (3) individual attention, and Dustin provides that -- incorporating all of these in its simulation-based learning architecture.

The other contribution is that Dustin brings a new degree of realism to computer-based instruction, introducing a new class of simulations. The centerpiece of Dustin's architecture is a combination of visual and aural information with an interface design that creates very realistic imitations of real life interactions. These simulated interactions convey linguistic, extra-linguistic, and contextual information, engage students, and bring into play social factors that are important in learning. By introducing this new generation of simulations, Dustin takes us a step closer to bringing real-life experiential learning to computer-based environments.

In addition, while Dustin contributes a design, MOPed helps build it. MOPed provides the blueprint and the building blocks for creating Dustin-like, simulation-based learning environments.

Language Learning

As I discussed previously, Dustin lets students practice, watch examples, receive guidance, and use tools in an environment that promotes language learning through experience. Students engage in meaningful tasks in the target environment, and develop communicative competence by interacting with others. Chapters 4, 5, and 6 describe the exact nature of this environment and discuss its applicability to domains other than language learning.

Dustin gives students access to tools and experiences coupled with a degree of control and safety that are not available in real life. At the press of a button, the student can eavesdrop on other people's conversation, ask for a transcript of any utterance, get translations for sentences, get a tool to record and play back his utterances, or jump into another conversation. By clicking on a word, she can bring up a dictionary that pronounces, explains, and translates the word for her. Also at her fingertips, is a cooperative tutor, ready to give hints and guide her whenever she wants and intervene with constructive comments when she does something wrong. All these resources and experiences come at no threat and are completely under the student's control. She can explore them according to her own needs, determining the amount of help that she receives and the kinds of experiences that she has, and address her particular interests at her own pace. Overall, Dustin is an environment rich with information, experiences, and tools to promote language acquisition.

Realistic Simulations

The central element of this rich environment is the simulation of interactions. To promote learning, interactions must be authentic, and their verisimilitude determines their effectiveness. Simulations must look realistic, and feel real enough so that students suspend disbelief and engage in experiences that lead to learning.

To make simulations seem realistic, Dustin combines multimedia technology with a number of interface solutions that incorporate essential elements of discourse (Clark & Schaefer, 1989). Discourse involves (1) verbal exchanges, (2) extra-linguistic means of communication, and a (3) common ground comprised of (3a) objects, (3b) concepts, and (3c) goals, which constitute integral parts of discourse. Dustin incorporates these elements by using (1) a natural language interface (i.e., in lieu of verbal communication), (2) conceptual feedback buttons through which users convey extra-linguistic information (e.g., Huh?, BYE), and (3a) a common set of objects that the user can manipulate (e.g., money, passport). In addition, the (3b,c) conceptual ground, including the goals of the interlocutors, is always carefully defined by the tutor.

Moreover, information captured by audio and video contributes to Dustin's realism. Simulated agents look like real people, talk like real people, behave like real people, and even react emotionally to annoying behavior. They are also always in the physical context in which they operate in real life (e.g., immigration's booth). The end result is a simulated interaction that feels real; Dustin creates the illusion that we are interacting with a real person.

Interacting with others is an important part of language learning, not only because we learn by using language in meaningful interactions, but also because interacting with others brings into play social factors that strongly influence learning. When someone gets angry at us, he conveys important information that guides us in fine tuning our own behavior. When using Dustin, at first, users laugh when simulated agents react emotionally, but as the novelty wears off, they begin to act as if they were involved in real life discourse. They understand, for example, that the immigration's agent is rightfully annoyed if they refuse to cooperate, and reactions from simulated agents begin to work as feedback that is essential for learning.

As a learning environment, Dustin introduces a number of innovations. Interactive simulations using natural language and object manipulation, coaching through individual instruction and feedback, and supporting tools combine to create an apprenticeship environment in which students learn experientially. Dustin merges a high degree of interactivity with visual and aural information in a new class of language learning environments. Most importantly, Dustin captures the realism essential to motivate and promote experiential learning, introducing a new look & feel to computer-assisted instruction.

Templates & Tools

In order to be useful to others, learning architectures must be accompanied by a blueprint, building blocks, and tools with which to build them. Dustin's architecture includes complex simulations and an intricate network of structures that is difficult to maintain without adequate tools. MOPed, the authoring tool, provides the mechanisms, the templates, and the knowledge-manipulation tools to help implement Dustin-like systems. The templates already implemented in Dustin serve as building blocks to create the next versions of simulation-based learning environments, and the existing dialogs help developers think about how to capture behaviors in future implementations. Dustin shows what we can do with realistic simulations, and MOPed shows exactly how to do it.

MOPed & AI

One day, when facing the need to scale up Dustin, we realized that small increases in size caused disproportionate increases in complexity. Adding more scenarios required intricate links to existing scenarios, and adding new rules in the rule-base generated complex interferences with existing rules. The interdependencies between new and old data structures made maintaining Dustin very confusing. The complexity of Dustin was already hard to grasp and this disproportionate growth in complexity exasperated the problem. The knowledge representation underlying most AI systems, usually consisting of intricately connected networks of structures, becomes very hard to maintain and understand as they increase in size, and Dustin's representation was no exception.

It became obvious that scaling up under those conditions was going to be unbearably hard. Three scenarios was as much as we could handle without tying a knot in our brains. My ability to grasp Dustin's complexity was not getting any better and Dustin's complexity was getting much worse. If we were to implement a full blown version of Dustin, we needed two things: (1) find a way of restraining this disproportionate growth in complexity, and (2) extend our ability to understand complex systems.

The result was MOPed, a knowledge-manipulation environment that restrains growth in complexity by (1a) reusing and (1b) contextualizing information, and that extends our ability to grasp complex networks of information through (2a) visual aids and (2b) knowledge-manipulation tools.

First, reusing data eliminates redundancy, therefore, reducing size. In addition, when the same information is stored in different places, it is very hard to keep all of them updated and compatible with each other. Reuse of data in MOPed is an important way of containing growth and facilitate maintenance.

Second, MOPed allows us to group information in manageable chunks. This enables us to add new information in modules, controlling complexity and making scaling up more manageable. Combined, MOPed's contextualization and reuse mechanisms provide a convenient support for modularization.

Third, MOPed extends our ability to understand complex networks of information by giving us visual aids, objectifying information, and by providing a knowledge manipulation environment that allows us to navigate conveniently in a large network of information.

In summary, MOPed uses two simple mechanisms to curtail growth in complexity: reuse and contextualization. Reuse guarantees that the amount of data doesn't grow disproportionately relative to size, and contextualization guarantees that the interdependencies added by new structures are contained within manageable contexts. In addition, it expands our ability to understand and maintain complex networks by providing a knowledge-manipulation environment.

Powerful Systems

In the field of AI, one thing that has been very problematic is that most AI systems work on just a few examples (Schank, 1991; McDermott, 1981). The cause is exactly the one I described above; they are complex and their complexity increases geometrically relative to size. They are difficult to maintain, and this difficulty generally means that above a certain size, usually two or three examples, we can no longer understand what's going on inside these systems. The interdependencies among the underlying knowledge structures become intractable. One unfortunate consequence of this problem is that AI systems are often incomprehensible to others, sometimes even to the developer, and researchers can't build on each other's work (McDermott, 1981).

However, AI scientists should be accountable for their computer-based experiments in the same way that psychologists and physicists are accountable for theirs. These researchers have to document their experiments in such a way that results can be replicated, allowing others to test and build on their knowledge. Similarly, if we are to build on AI's mistakes and findings, we need a clear definition of what computer-based experiments do and how they do it. The burden lies on the developer who should always help others understand and manipulate what they invent.

In AI, McDermott points out, researchers get away with building theories on experiments that they didn't really run -- "Only a preliminary version of the program was actually implemented." (McDermott, 1981). They pontificate without even running the experiment, and in those rare cases in which they indeed implement the experiment, they usually build such cryptic systems that we end up having to take their word for it.

Truth maintenance systems, networks of frames, and temporal reasoning involve complex knowledge networks that become very hard to understand and maintain as they increase in size. Our inability to scale them up says something about the limits of our comprehension, showing exactly where humans need help. It shows very clearly the threshold beyond which we humans can't grasp the complexity of intricate networks of information.

For example, if expert-systems researchers had thought of maintainability from day one, they would have discovered the need to help users contextualize, reuse, visualize, and prioritize rules. The resulting effort would have made them either give up on the idea of using production systems or come up with some clever knowledge organization schemes. Either way, they would have saved AI a lot of bad press.

What happens instead is that AI researchers suffer clarity-phobia, fearing that to expose the guts of their programs is to show that they do not do AI at all. Instead of encouraging simplicity, this fear reinforces the creation of cryptic representations that perpetuate the obscurity of AI programs. But to make progress, researchers must understand each other's work. Consequently, if the systems we build are so complex that we wouldn't understand them without tools, then building tools to help others, and ourselves, understand and manipulate them becomes an important part of developing powerful systems.

MOPed makes it very clear what Dustin does and how it does it. After a ten-minute demo of the tool, showing the guts of Dustin, it becomes very easy for anyone to see exactly how Dustin works -- the patterns of speech that it understands, those that it doesn't, when it cheats, how it cheats, how much processing it does with a button press, how things are hard-coded, and how they aren't. Instead of building a complex, incomprehensible system for which I could claim AI feats, we developed a tool that completely exposes Dustin, empowering users to grasp and maintain the complex information embedded in simulation-based learning environments.

Extending Intelligence

MOPed is certainly not the solution to all problems. However, it stands for an important function in AI: to extend intelligence. We have built a complex, simulation-based learning environment, Dustin, and unavoidably, Dustin involves complex networks of information, networks so complex that we can't understand them with the naked eye. MOPed explores computer's potential for creating cognitive extensions, and empowers us to understand more than we normally can.

References

Asher, J. (1977). Learning another language through actions: The complete teacher's guide book. (2nd ed.). Los Gatos, CA: Sky Oaks Productions.

Beebe, L. M. (1988). Issues in second language acquisition. New York: Newbury House Publishers.

Bloomfield, L. (1933). Language. New York: Holt.

Braun, T. E. D., Mulford, G. W. (1987). Computer-based instruction as a supplement to a modern French curriculum. In New Developments in Computer-Assisted Language Learning. (pp. 137-154). New York: Nichols Publishing Company.

Brown, J. S., Collins, A., Duguid, P. (1989). Situated Cognition and the culture of learning. Educational Researcher, vol 18-1, pp. 32-42.

Bruner, J. (1962). On knowing: Essays for the left hand. Mass. : Harvard University Press.

Chomsky, N. (1959). A review of B. F. Skinner's Verbal behavior. Language, 35(1), 26-58.

Chomsky, N. (1966). Linguistic theory. In J. P. B. Allen & P. Van Buren (Eds.), Chomsky: Selected Readings, (pp. 152-9). London: Oxford University Press.

Clark, H. H., Schaefer, E. F. (1989). Contributing to discourse. Cognitive Science, 13, 259-294.

Coleman, C. A. (1929). The teaching of modern foreign languages in the United States. New York: Mcmillan.

Collins, A. (1988). Cognitive Apprenticeship and Instructional Technology. (Report No. 6899). Cambridge, MA: BBN Laboratories Incorporated.

Collins, A. (1990). Generalizing from situated knowledge to robust understanding. Unpublished manuscript.

Collins, A., Brown, J. S. (1988). The computer as a tool for learning through reflection. In H. Mandl & A. Lesgold (Eds.), Learning issues for intelligent tutoring systems (pp. 1-18). New York: Springer-Verlag.

Collins, A., Brown, J. S., Newman, S. E. (1989). Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics. In Resnick, L. B. (Ed.), Knowing, learning, and instruction: Essays in honor of Robert Glaser (pp. 453-494). Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Curran, A. (1972). Counseling-learning: A whole-person model for education. New York: Grune and Sttratton.

Curran, C. A. (1976). Counseling-learning in second languages. Apple River, Ill: Apple River Press.

Doughty, C. (1991). Theoretical Motivations for IVD Software Research and Development. CALICO Monograph Series, 2, 1-15.

Dulay, H., Burt, M., Krashen, S. (1982). Language Two. New York: Oxford University Press.

Ellis, R. (1990). Instructed Second Language Acquisition. Cambridge: Basil Blackwell, Inc.

Finocchiaro, M., Brumfit, C. (1983). The Functional-Notional Approach. New York: Oxford University Press.

Fries, C. C. (1945). Teaching and learning English as a foreign language. Ann Arbor: Unviersity of Michigan Press.

Fries, C. C. (1948). As we see it. Language Learning, 1, 12-16.

Gattegno, C. (1972). Teaching foreign languages in schools: The Silent Way. (2nd ed.). New York: Educational Solutions.

Gattegno, C. (1976). The common sense of teaching foreign languages. New York: Educational Solutions.

Genesee, F. (1988). Neurolinguistic Perspective. In Beebe, L. M. (Ed.). (1988). Issues in second language acquisition: Multiple Perspectives. New York: Newbury House Publishers.

Harley, B. (Ed.) (1990). The development of second language proficiency. Cambridge: Cambridge University Press.

Klier, B. (1987). A microcomputer game in French culture and civilization. In New Developments in Computer-Assisted Language Learning. (pp. 69-77). New York: Nichols Publishing Company.

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

Krashen, S. D. (1985). The input hypothesis: Issues and implications. London: Longman.

Krashen, S. D., Terrell, T. D. (1983). The Natural approach: Language acquisition in the classroom. Oxford: Pergamon.

Lampe, D. R. (1988). Athena MUSE: Hypermedia in Action. The MIT Report, pp. 4-12.

Lepper, M. R., Chabay, R. W. (1985, April). Intrinsic motivation and instruction: Conflicting views on the role of motivational processes in computer-based education. Invited address to the Biennial meeting of the Society for Research in Child Development, Toronto.

Lepper, M. R., Greene, D. (Eds.). (1978). The hidden costs of reward. Hillsdale, N. J.: Erlbaum.

Maddison, P., Maddison, A. (1987). The advantages of using microcomputers in language teaching. In New Developments in Computer-Assisted Language Learning. (pp. 20-31). New York: Nichols Publishing Company.

Malone, T. W. (1981). Toward a theory of intrinsically motivating instruction. Cognitive Science, 4, 333-369.

McDermott, D. (1981). Artificial Intelligence meets natural stupidity. In J. Haugeland (Ed.). Mind Design. Cambridge: The MIT Press.

McLaughlin, B. (1978). The Monitor Model: Some methodological considerations. Language Learning, 28(2), 309-332.

McLaughlin, B. (1987). Theorie of second-language learning. London: E. Arnold.

MIT (1990). Project Athena Visual Computing Group. (Available from Ben Davis, MIT).

Morgenstern, D. (1986). Simulation, Interactive Fiction and Language Learning: Aspects of the MIT Project. Bulletin of the CAAL, pp. 23-83.

Murman, E. M. (1989, June). The Athena Project: Goals, Philosophy, Status, and Experiences. Nordic Conference on Computer Aided Learning in University and Higher Education. Trondheim, Norway.

Murray, J. H. (1987). Humanists in an institute of technology: How foreign languages are reshaping workstation computing at MIT. Academic Computing, pp. 34-38.

Murray, J. H. (1990). The Athena Language Learning Project. (Available from Janet H. Murray, MIT, Department of Humanities, Athena Language Learning Project).

Palincsar, A. S., Brown, A. L. (1984). Reciprocal teaching of comprehension-fostering and monitoring activities. Cognition and Instruction, 1, 117-175.

Papert, S. (1976). Mindstorms: Children, Computers, and Powerful Ideas. New York, Basic Books, Inc.

Papert, S. (1987). Microworlds: Transforming Education. In T. W. Lawler, Y. Masoud (Eds.). (1987). Artificial Intelligence and Education (Vol I). New Jersey: Ablex Publishing.

Richards, J. C. (1990). The Language Learning Matrix. Cambridge: Cambridge University Press.

Richards, J. C., Rodgers T. S. (1986). Approaches and Methods in Language Teaching: A description and analysis. Cambridge: Cambridge University Press.

Rivers, W. M. (1983). Communicating Naturally in a Second Language. Cambridge: Cambridge University Press.

Rivers, W. M. (1987a). Interaction as the key to teaching language for communication. In W. M. Rivers (Ed.), Interactive Language Teaching. Cambridge: Cambridge University Press.

Rivers, W. M. (1987b). Preface In W. M. Rivers (Ed.), Interactive Language Teaching. Cambridge: Cambridge University Press.

Saunders, P. (1987). 'The Dark Castle' - An adventure in French. In New Developments in Computer-Assisted Language Learning. (pp. 69-77). New York: Nichols Publishing Company.

Scardamalia, M., Bereiter, C. (1985). Fostering the development of self-regulation in children's knowledge processing. In S. F. Chipman, J. W. Segal, & R. Glaser (Eds.), Thinking and learning skills: Research and open questions (pp. 563-577). Hillsdale, NJ: Lawrence Erlbaum Associates.

Scardamalia, M., Bereiter, C., McLean, R. S., Swallow, J., Woodruff, E. (1989). Computer-supported intentional learning environments. Journal of educational computing research., 5, 51-68.

Schank, R. C. (1982). Dynamic Memory: A theory of reminding and learning in computers and people. Cambridge: Cambridge University Press.

Schank, R. C. (1991). Case-Based Teaching: Four experiences in Educational Software Design. (Report No. 7). Evanston, IL: Institute for the Learning Sciences.

Schank, R. C. (1991). Where's the AI? AI Magazine, Winter.

Schank, R. C., Abelson, R. (1977) Scripts, Plans, Goals and Understanding: An inquiry into human knowledge structures. Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Schoenfeld, A. H. (1983). Problem solving in the mathematics curriculum: A report, recommendations and an annotated bibliography. The Mathematical Association of America, MAA Notes, No. 1.

Shoenfeld, A. H. (1985). Mathematical Problem Solving. New York: Academic Press.

Slaton, A. (1991). How to get started in interactive videodisc: A user's perspective. CALICO Monograph Series, 2, 25-29.

Terrell, T. D. (1977). A natural approach to second language acquisition and learning. Modern Language Journal, 61, 325-336.

Terrell, T. D. (1981). The Natural Approach in bilingual education. Ms. California Office of Bilingual Education.

Terrell, T. D. (1982). The natural approach to language teaching: an update. Modern Language Journal, 66, 121-132.

Underwood, J. (1991). Interactive Videodisc as Hypermedia. CALICO Monograph Series, 2, 63-71.

Weizenbaum, J. (1966). ELIZA - A computer-program for the study of natural language communication between man and machine. Communications of the ACM, 9, 36-45.

Wilkins, D. A. (1973). The linguistic and situational content of the common core in a unit/credit system. In Systems Development in Adult Language Learning, Strasbourg, Council of Europe.

Wilkins, D. A. (1976). Notional Syllabuses. Oxford: Oxford University Press.

Yale University (1987). Instructor's Guide. French in Action: A beginning course in language and culture - The Capretz method - Instructor's guide, 1.