

SOCIO-TECHNICAL ENVIRONMENTS  
SUPPORTING DISTRIBUTED COGNITION FOR  
PERSONS WITH COGNITIVE DISABILITIES

by

STEFAN PARRY CARMEN

B.Sc., University of Colorado, 1992

M.S., University of Colorado, 2002

A thesis submitted to the  
Faculty of the Graduate School of the  
University of Colorado in partial fulfillment  
of the requirements for the degree of  
Doctor of Philosophy  
Department of Computer Science

2006

This thesis entitled:  
Socio-Technical Environments Supporting Distributed Cognition for Persons with  
Cognitive Disabilities

written by Stefan Parry Carmien  
has been approved for the Department of Computer Science

Dissertation Committee:

Gerhard Fischer (chair) \_\_\_\_\_

Clayton Lewis \_\_\_\_\_

Cathy Bodine

James Martin

Lesyia Palen

James Sullivan

Date: \_\_\_\_\_

The final copy of this thesis has been examined by the signatories, and we  
find that both the content and the form meet acceptable presentation standards  
of scholarly work in the above mentioned discipline.

HRC protocol # 0402.01

Carmien, Stefan Parry (Ph.D., Computer Science)

Socio-Technical Environments Supporting Distributed Cognition for Persons with Cognitive Disabilities

Thesis directed by Prof. Gerhard Fischer

In the United States, 4.64 million persons have cognitive disabilities, and of these a significant fraction is potentially able to live more independently save for a deficiency of mnemonic and executive capability. In many cases, these persons are supported by concerned caregivers who want them to live in a less dependent fashion. Persons with cognitive disabilities as well as caregivers could all benefit from a socio-technical environment designed to support their legitimate and reasonable aspirations. My research platform, MAPS (**M**emory **A**iding **P**rompting **S**ystem), aims to provide a simple effective prompting system for individuals with cognitive disabilities with an interface for designing prompting scripts by caregivers. MAPS provides a socio-technical environment that acknowledges the needs and abilities of members of the communities of caregivers and persons with cognitive disabilities. By using and extending human-computer interaction (HCI) frameworks and theory—such as distributed cognition, metadesign, and symmetry of ignorance—in a principled design environment, this research demonstrates, analyzes, and documents how to create systems that potentially could avoid the all-too-common fate of assistive technology, that of abandonment. MAPS comprises two technical components: the MAPS script-design-environment, a personal computer (PC)-based system that allows a caregiver to edit, store, and reuse scripts of multimedia prompts for task support; and the MAPS-prompter, a PDA-based mobile prompting environment for persons with cognitive disabilities. The process of adopting MAPS was observed and analyzed by using ethnographic methods to study dyads of persons with cognitive disabilities and caregivers doing real tasks in home, shopping, and employment environments. Based on these observations and analyses, this research delineated new ways to use traditional HCI perspectives and produced a set of heuristics to aid in the design and use of prompting systems and the more general design of assistive technology.

## Dedication

In the Buddhist tradition, we aspire to give the benefit of any significant work, whether a long retreat or any day-to-day task, to all beings in the world, for the relief of their confusion and suffering. So in that spirit I wish any benefit that this work obtains be given to those in need, specifically to the community of young adults with cognitive disabilities and caregivers, and in a larger sense to all suffering beings.

By this merit may all obtain omniscience.

May it defeat the enemy, wrongdoing.

From the stormy waves of birth, old age, sickness, and death,

From the ocean of samsara, may I free all beings.

## Acknowledgments

I would like to express my thanks to the Coleman Foundation for inspiring and funding my work, to Gerhard Fischer, my advisor, for leading me to many of the topics presented here and for being the helmsman of the Center for LifeLong Learning and Design; to my fellow CLever researchers, especially Andy Gorman for his Lifeline system (and many hours of visionary speculation), Anja Kintsch for helping me pay attention and for her invaluable domain expertise, Jim Sullivan for his difficult-to-answer questions, Rogerio DePaula for his theoretical correctness, and to my friend and undergraduate research apprentice Genevieve Hudak. A special thanks goes to the participants in my field trials: parents, caregivers from BVSD, Imagine!, and the participants with cognitive disabilities, both for providing the real ground for research and for sharing their world and changing mine. Hal Eden and Francesca Iovine of L<sup>3</sup>D both helped with infrastructure and paperwork and also provided a warm home for my work in the last six years. The L<sup>3</sup>D center itself and its members, especially Mark Dubin and Ernie Arias, have provided an invaluable forum to present my work over the years, offering genuine and pointed criticism. Elisa Giaccardi clarified and confused me, but always cheered me on. Fellow practitioners from the Aphasia project, RESNA, and ASSETS have been a source of inspiration and have provided a matrix of existing work to build upon.

My gratitude goes to my committee, especially Cathy Bodine and Clayton Lewis, who have asked the difficult questions and demanded the best effort from me; and Leysia Palen, who initiated me into ethnography. My father and stepmother contributed much support during the time it took to peristaltically work my way through graduate school. My family has always been the center of my mandala, generating inspiration and chaos—both necessary ingredients in any genuine search for knowledge. So to Sophia, Saskya, and Mikhail, my deep gratitude for grounding my life during the time I have not been as present as I wished, instead doing this work, and especially to my wife Hjordis, my appreciation for the richness of support and love. Ultimately, my search for personal meaning and the skillful means to bring compassion to action stems from my time with Chögyam Trungpa Rinpoche and from his teachings, to whom I am always deeply grateful.

## TABLE OF CONTENTS

<b>1. Introduction .....</b>	<b>1</b>
1.1. <i>Overview</i> .....	1
1.2. <i>Motivation</i> .....	6
1.3. <i>Framing the Problem</i> .....	8
1.4. <i>Goal of the Research</i> .....	9
1.5. <i>Research Activities</i> .....	11
1.6. <i>Reading Guide</i> .....	13
<b>2. The Problem.....</b>	<b>16</b>
2.1. <i>Some Definitions</i> .....	17
2.2. <i>Independence</i> .....	26
2.3. <i>Abandonment</i> .....	27
<b>3. Theoretical Framework .....</b>	<b>29</b>
3.1. <i>Overview</i> .....	29
3.2. <i>Socio-Technical Environments</i> .....	31
3.3. <i>Distributed Cognition</i> .....	32
3.4. <i>Symmetry of Ignorance</i> .....	36
3.5. <i>Metadesign</i> .....	37
3.6. <i>Bridging from Framework to Design</i> .....	38
<b>4. Related Work and Systems.....</b>	<b>40</b>
4.1. <i>Overview</i> .....	40
4.2. <i>Stationary Multimedia Prompting Systems</i> .....	40
4.3. <i>Multimedia Handheld Prompting</i> .....	42

<b>5. The Memory Aiding Prompting System (MAPS) Socio-Technical Environment .....</b>	<b>47</b>
5.1. <i>Scenarios</i> .....	47
5.2. <i>MAPS Design</i> .....	60
5.3. <i>The MAPS Socio-Technical Environment</i> .....	67
5.4. <i>Technical Approach</i> .....	69
<b>6. Assessment and Field Studies .....</b>	<b>81</b>
6.1. <i>Methodology</i> .....	82
6.2. <i>Design of the Handheld</i> .....	88
6.3. <i>Design of the Script-Design-Environment</i> .....	90
6.4. <i>Preliminary Field Trial of the Handheld with a Complex Script</i> .....	93
6.5. <i>Image Testing</i> .....	95
6.6. <i>Realistic Field Trials</i> .....	96
6.7. <i>Evaluation of Field Trials</i> .....	118
6.8. <i>Realistic Field Trials and Heuristics</i> .....	129
6.9. <i>Key Design Contributions</i> .....	133
6.10. <i>Concluding Thoughts about Assessment of MAPS</i> .....	136
<b>7. Theoretical Contributions .....</b>	<b>138</b>
7.1. <i>Overview</i> .....	138
7.2. <i>Distributed Cognition</i> .....	138
7.3. <i>Symmetry of Ignorance</i> .....	155
7.4. <i>Metadesign</i> .....	157
7.5. <i>Conclusion of Contributions</i> .....	166
<b>8. Future Work .....</b>	<b>170</b>
8.1. <i>Lifeline</i> .....	171
8.2. <i>Extending MAPS</i> .....	172
8.3. <i>MAPS as a Platform for Active Distributed Support</i> .....	176

8.4.	<i>Work Supporting Prompting: An Extension of the Image Experiment</i> .....	181
<b>9.</b>	<b>Conclusions</b> .....	<b>182</b>
9.1.	<i>Leveling the Playing Field</i> .....	183
<b>10.</b>	<b>Glossary</b> .....	<b>185</b>
<b>11.</b>	<b>Abbreviations</b> .....	<b>190</b>
<b>12.</b>	<b>References</b> .....	<b>192</b>
<b>13.</b>	<b>Appendices</b> .....	<b>205</b>
13.1.	<i>Appendix A – Initial Interview with Experts: Topics</i> .....	206
13.2.	<i>Appendix B – Usability Test</i> .....	207
13.3.	<i>Appendix C – Usability Evaluation Sheet</i> .....	210
13.4.	<i>Appendix D – Caregivers’ Expectations</i> .....	211
13.5.	<i>Appendix E – Task Analysis Checklist</i> .....	213
13.6.	<i>Appendix F – Script Template Form</i> .....	214
13.7.	<i>Appendix G – Field Trials Exit Interview</i> .....	216
13.8.	<i>Appendix H – MAPS Database Schema</i> .....	218
13.9.	<i>Appendix I – MAPS-Prompter Log</i> .....	219
13.10.	<i>Appendix J – Basic Use of Prompter Test – Glider</i> .....	221
13.11.	<i>Appendix K – Glider Test Evaluation Form</i> .....	223
13.12.	<i>Appendix L – Script Sample – Linear Script</i> .....	225
13.13.	<i>Appendix M – Script Sample Using Forking and Looping</i> .....	228
13.14.	<i>Appendix N – Script Designed for Collapsing</i> .....	232
13.15.	<i>Appendix O - Ethnographic Analysis Matrix</i> .....	234
13.16.	<i>Appendix P - Caregivers Crib Sheet Example</i> .....	241



## List of Tables

Table 1: <i>Summary of Research Agenda</i> .....	11
Table 2: <i>Comparison of Frameworks</i> .....	38
Table 3 - <i>Initial Prompter Test</i> .....	89
Table 4 - <i>MAPS Script Design Environment Usability Testing</i> .....	91
Table 5 - <i>Chinese Cookies Script Test</i> .....	94
Table 6 – <i>Image Type Experiment</i> .....	95
Table 7 - <i>Realistic Field Trials</i> .....	97
Table 8: <i>Realistic Field Testing</i> .....	98
Table 9 - <i>Summary of MAPS Testing</i> .....	117
Table 10 - <i>Overview of Scripts, Internal and External</i> .....	141
Table 11 - <i>Overview of Tools for Living and Tools for Learning</i> .....	143
Table 12 - <i>Symmetries of Ignorance</i> .....	156
Table 13 - <i>Overview of Metadesign Attributes</i> .....	158

## List of Figures

Figure 1 -The MAPS Prompter .....	2
Figure 2 - The MAPS Script Design Environment .....	3
Figure 3 – CLever: Attributes That Together Assist Meaningful Independence .....	4
Figure 4 - Csikszentmihalyi's Optimal Flow Diagram.....	5
Figure 5 - Traditional Training Prompting Script.....	8
Figure 6 - Research Process.....	12
Figure 7 - Range of Target Population for MAPS .....	22
Figure 8 - Abilities in Sea of Deficits .....	24
Figure 9 - Deficits in Sea of Abilities .....	25
Figure 10 - AbleLink's Pocket Compass and Task Builder.....	42
Figure 11 - PEAT Memory Aid .....	45
Figure 12 - Isaac .....	46
Figure 13 - MAPS Video Help.....	52
Figure 14 - MAPS Script-Design-Environment Preview Pane.....	53
Figure 15 - Switch Database.....	55
Figure 16 - Load Template Option.....	56
Figure 17 - Prompter in a Palmtop Computer Holder.....	60
Figure 18 - MAPS and Lifeline Project as an Information Ecology.....	61
Figure 19 – MAPS Script-Design-Environment Functional Detail .....	63
Figure 20 - PowerPoint User Interface .....	64
Figure 21- MAPS-Prompter Functional Detail .....	65

Figure 22 - MAPS Scripter Splash Screen .....	66
Figure 23 – MAPS-Prompter Easter Egg Dialog Boxes.....	71
Figure 24 – MAPS-Prompter in Multiscript Mode .....	72
Figure 25 – MAPS Script-Design-Environment Detail .....	73
Figure 26 - File Options: Load and Delete .....	74
Figure 27 - Script-Design-Environment in Append (left) and Insert Modes (right).....	75
Figure 28 - MAPS in Visualize Whole Script at Once Mode .....	76
Figure 29 - MAPS Script-Design-Environment User Options.....	77
Figure 30 – Wav Filename Sorting and Insertion Tool .....	77
Figure 31 - Tool Tip Help.....	79
Figure 32 - Two Prompting Steps from the Glider Script .....	88
Figure 33 - MAPS Script-Design-Environment Iterative Design.....	92
Figure 34 - Distributed Cognition Artifacts in the Environment of a Task.....	104
Figure 35 - Secondary Artifact for Script Assembling in the Script-Design-Environment .....	105
Figure 36 - Clothes Store Color Order.....	114
Figure 37 - ChatPC II Carrying Case.....	115
Figure 38 - Microsoft Sound Recorder .....	122
Figure 39 - Internal and External Scripts.....	140
Figure 40 - Bicycle with Training Wheels — A Tool for learning.....	145
Figure 41 - Elder Tricycle.....	147
Figure 42 - A TfLea Used as a TfLiv?.....	151
Figure 43 - MAPS at Script-Design Time .....	159

Figure 44 - MAPS Prompter at Script-Use Time.....	160
Figure 45 – Metadesign – Design Over Time and Role .....	161
Figure 46 - Metadesign, Plans, and Situated Action.....	162
Figure 47 - Prompts on Cards.....	168
Figure 48 - Lifeline Caregivers Interface .....	170
Figure 49 - MAPS Script-Design-Environment Interface for MAPS/Lifeline Prototype .....	171
Figure 50 - MAPS Error Trapping and Conditional Prompting Annotation Interface .....	178
Figure 51 - Lifeline Caregiver Interface with Error Message .....	180

# 1. Introduction

## 1.1. Overview

As “universal machines,” computers offer tremendous promise; they have made us more powerful in previously inconceivable fashions, but in many ways this revolution has served only to complicate our lives. One area in which computationally enhanced environments have held much promise but delivered only a fraction of this potential is that of *assistive technology* (AT) [US Technology-related Assistance for Individuals with Disabilities Act, 1988], the use of technology to “even the playing field” for people with various kinds of deficiencies, ranging from mobility to cognitive. As computer scientists, we know much about intelligence augmentation and how humans use and interact with computers (known as human-computer interaction, or HCI). Might this focus and accumulated knowledge be put to the task of aiding those who might be uniquely suited to benefit from our domain of expertise—people with *cognitive disabilities* [American Psychiatric Association. Task Force on DSM-IV., 2000]<sup>1</sup>?

Individuals with cognitive disabilities are often unable to live independently due to their inability to perform *activities of daily living* (ADL) [Katz S et al., 1963], such as cooking, housework, or shopping. By being provided with *socio-technical environments* [Mumford, 1987] to extend their independence, these individuals can lead richer, fuller lives. Traditionally, training has provided ADL support by utilizing prompting and task segmentation techniques. Historically, prompting has been part of this instructional technology: being prompted through tasks in a rehearsal mode and then using the memorized instructions in daily life. Having learned to do a specific task, individuals then can go into the world with new skills. These instructions constitute a *script* [Schank & Abelson, 1977] that, when followed in the exact environment wherein it was designed, allows a person to complete a task.

---

<sup>1</sup> A recent trend uses the term “intellectual disabilities” as a subset of “cognitive disabilities”; this is to differentiate disabilities that result in lowered cognitive function (i.e., retardation) from disabilities that cause a lessened ability to read (i.e., dyslexia) or communicate (i.e., aphasia). In this work, *cognitive disabilities* refers to the former.

However, some individuals do not have the capacity to memorize and properly recall the steps necessary for some tasks. In addition, the context of the task and the task itself may change, rendering the training useless. Recent advances in computer technology, including powerful personal digital assistant (PDA) devices, have provided an opportunity to create prompting systems that could remedy this problem.

Providing a theoretical support for my research, and affording a basis for evaluating and extending my work, are HCI theoretical studies in *distributed cognition* [Hollan et al., 2001]; *symmetry of ignorance* [Rittel, 1984]; and *metadesign* [Giaccardi & Fischer, 2005]. In addition to these theoretical underpinnings, this dissertation explores several extensions. Two embellishments of distributed cognition, *internal and external scripts* (see section 7.2) and *tools for learning/tools for living* (see section 7.2.2), form useful conceptual frameworks for a design rationale. Application of the principle of symmetry of ignorance in the service of designing assistive technology for this unique population yielded the notion of *dual user interfaces* (see section 7.4.3). Applying the aforementioned theoretical perspectives and the dual interface principle to this problem produces a framework that supports metadesign as *design over time* (see section 7.4.1) and *one end user with two roles* (person with cognitive disabilities and caregiver), as well as metadesign for *three persons* (person with cognitive disabilities, caregiver, and tool designer) (see section 7.4.2). This framework also provides a way to understand the larger issues involved in creating socio-technical systems for assistive technology and other domains.



Figure 1 -The MAPS Prompter

A substantial portion of all assistive technology is abandoned after initial purchase and use [Reimer-Reiss, 2000]. A large component of the cause for such *abandonment* is difficulty in configuring and adapting (reconfiguring) software [King, 1999; King, 2001; Phillips & Zhao, 1993; Reimer-Reiss, 2000; Russel, 1997]. The prompting system MAPS (Memory Aiding Prompting System), in answer to these difficulties, incorporates a palmtop-based multimedia *prompter* [Davies et al., 2002] (see Figure 1) and a PC-based intuitive interface for script creation and configuration (see Figure 2). The MAPS caregivers interface, running on a PC and utilizing a web-based repository of existing scripts and script templates, was designed with input from AT experts and *caregivers* [Snell, 1987] to provide a learning curve as shallow as possible for initial use.

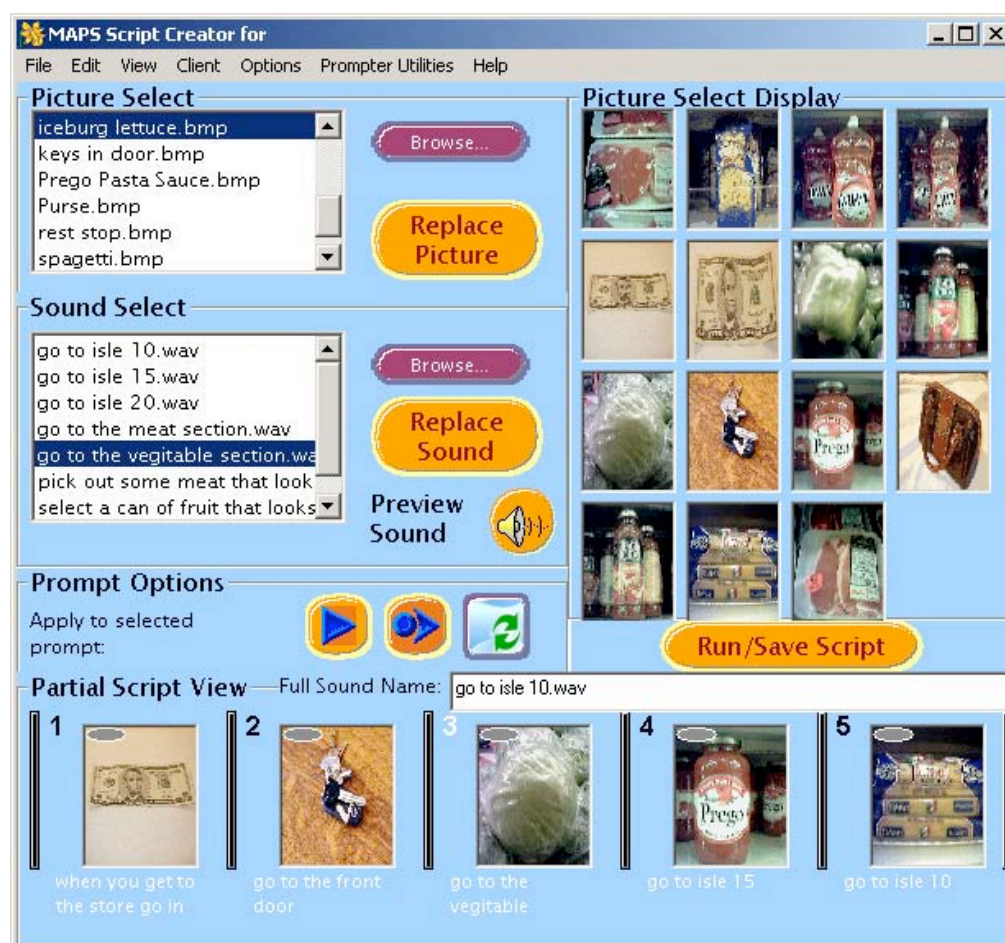
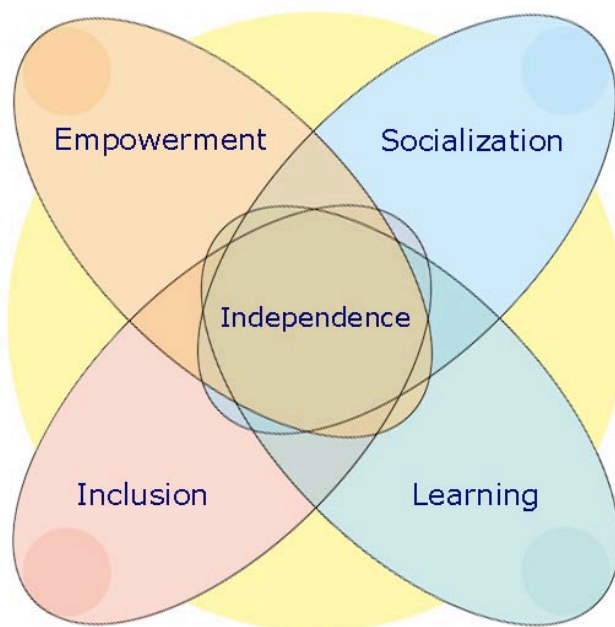


Figure 2 - The MAPS Script Design Environment

MAPS is one of a set of applications and frameworks that have been produced by the Cognitive Lever (CLever) project [CLever, 2005], a research group within the Center for LifeLong Learning and Design (L3D) at the University of Colorado, Boulder [L3D, 2005]. The mission of the CLever project is to develop computationally enhanced environments designed to assist not only people with a wide range of cognitive disabilities, but also their support community.

Figure 3 represents the different approaches CLever takes to support independence for persons with cognitive disabilities. The notion of independence here does not mean unattached or totally unconnected; it is shaded by the very human values of inclusion and socialization. The Lever in CLever is a metaphor for an approach to designing technology with an aim to augment rather than replace intelligence [Fischer & Nakakoji, 1992; Terveen, 1995]. Traditional approaches to the use of artificial intelligence (AI) techniques attempt to replace missing functions, such as active spell checkers replacing the word you were typing with the word it thinks you meant to type..

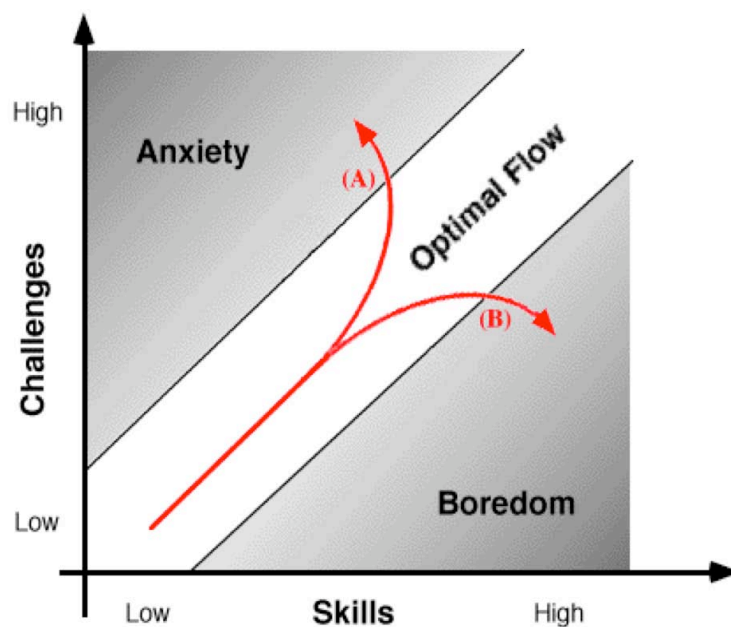


**Figure 3 – CLever: Attributes That Together Assist Meaningful Independence**

The approach taken here, however, is to use what abilities there are and to leverage them by using a distributed cognition approach (see section 3.3) to support the person with cognitive disabilities in



his/her desire for independence. The task is accomplished by the person with cognitive disabilities plus the correct AT tool. If the tool is not supportive enough, the experience of the user swings upward toward anxiety, and the task becomes too difficult to do with ease; if the tool is too much of a replacement for skills that the person *could* gain, the balance swings toward boredom (from Csikszentmihalyi's flow framework [Csikszentmihalyi, 1996], see Figure 4)—neither extreme bodes well for adoption. The issue is to determine the best place to put the fulcrum and what form the representations of the distributed cognition artifact to take.



**Figure 4 - Csikszentmihalyi's Optimal Flow Diagram**

CLever research activities range from community building [dePaula, 2004], to technology recommender systems [Kintsch, 2000], to assistive technology systems focused on users with cognitive disabilities and their caregivers. CLever is also part of the Rehabilitation Engineering Research Center for the Advancement of Cognitive Technologies (RERC-ACT) funded by The Coleman Institute [Coleman, 2004]. MAPS developed in the caldron of the L<sup>3</sup>D lab that produces these systems and benefited from the panoramic perspective it nurtured. Early in the CLever project, many different stakeholders were involved; particularly beneficial was the hiring of Anja Kintsch, an AT specialist with the Boulder Valley School District (BVSD) [BVSD, 2004]. Anja proved enormously useful in keeping CLever researchers from pursuing nonproductive inquiries; in addition, she is a strong

expounder of “best practices.” The Aphasia [Boyd-Graber et al., 2006] project has a similar AT position on their team, for the same reasons. Any research project attempting to do real AT systems development should have such a position as a requirement.

The purpose of this research is to investigate, in the domain of assistive technology, for persons with cognitive disabilities:

- Distributed cognition [Hollan et al., 2001; Salomon, 1993]
- Metadesign [Fischer, 2000; Fischer & Giaccardi, 2006]
- Symmetry of ignorance [Fischer et al., 2002; Rittel, 1984]

These frameworks, along with expertise in designing and evaluating user interfaces for, and cognitive issues with, persons with cognitive disabilities as well as expertise in the domain of assistive technology, provide the basis for the MAPS project. Because the problem is so intimately bound to the very different stakeholder roles and embedded in a social milieu, any design solution to this problem must take a socio-technical approach. Socio-technical environments consist of more than the coincidental concurrence of people and technical components [Mumford, 1987] or their connection by HCI. In socio-technical environments, the technical components are embedded into this web of communications in a specific way to accomplish an action or ongoing enterprise. Socio-technical design is an approach to design that recognizes the strong and co-evolutionary interactions between people and technology.

## 1.2. Motivation

There are 4.64 million<sup>2</sup> persons with significant cognitive disabilities in the United States [Braddock, 2006], many of whom might be able to live more independently with properly designed

---

<sup>2</sup> Out of the total of 21.7 million people with cognitive disabilities, this represents those with mental retardation or intellectual disabilities (4.64 million) only.

help [The American Association on Mental Retardation, 1992]. They are often unable to live on their own due to deficiencies in memory, attention, and executive functionalities [LoPresti et al., 2004]. These deficits may lead to an inability to consistently perform typical domestic tasks such as cooking, taking medications, performing personal hygiene, and taking public transportation. By providing socio-technical tools to extend their independence, persons with cognitive disabilities can have richer, fuller lives.

Prompting is a traditional technique that rehabilitation and transition professionals have used to support increasing independence for persons with cognitive disabilities [Carmien, 2004b]. A traditional prompting instructional tool is shown in Figure 5 [LRE for LIFE Project, 2001]. The leftmost column numbers the steps in the task, which in this case is *sweeping the floor*. The next column lists the separate subtasks that, if memorized and performed in the right sequence, constitute successful floor sweeping (note that they are ordered from bottom to top); the rest of the document is devoted to notational space for a job coach to indicate the repetitions of learning and technique that are traditionally used to memorize the task by the person with cognitive disabilities. In this sort of prompting, scripts are designed to facilitate memorization by the clients, who then could have the script as one of the tasks that they could do on their own. In the case of MAPS, the scripts remain on the handheld prompter, and the person with cognitive disabilities *with* the prompter are able to do the task unassisted.

A substantial portion of all assistive technology is abandoned after initial purchase and use—as high as 70 percent in some cases [Phillips & Zhao, 1993]. The tragedy here is that the very population that could most benefit from technology is paying for exceptionally expensive devices (many thousands of dollars due to the complexity and limited number manufactured) that end up in the back of closets after a short time. In summary, the motivation for this research is to understand the process of designing and providing a prompting tool that enables independent living by persons with cognitive disabilities and will be adopted by this population.

Name: \_\_\_\_\_

Goal: Sweep floor

Data: / = correct response  
 X = incorrect response  
 P = prompted response  
 Circle Total Number Steps Correct

Natural Cue(s):		Natural Consequences:					
step	Step/Behavior						
14	Replace items on the floor	14 14 14 14 14	14 14 14 14 14	14 14 14 14 14	14 14 14 14 14	14 14 14 14 14	14 14 14 14 14
13	Puts chairs on the floor	13 13 13 13 13	13 13 13 13 13	13 13 13 13 13	13 13 13 13 13	13 13 13 13 13	13 13 13 13 13
12	Return broom/dustpan to storage	12 12 12 12 12	12 12 12 12 12	12 12 12 12 12	12 12 12 12 12	12 12 12 12 12	12 12 12 12 12
11	Disposes of dirt	11 11 11 11 11	11 11 11 11 11	11 11 11 11 11	11 11 11 11 11	11 11 11 11 11	11 11 11 11 11
10	Sweeps dirt into dustpan	10 10 10 10 10	10 10 10 10 10	10 10 10 10 10	10 10 10 10 10	10 10 10 10 10	10 10 10 10 10
9	Sweeps entire surface	9 9 9 9 9	9 9 9 9 9	9 9 9 9 9	9 9 9 9 9	9 9 9 9 9	9 9 9 9 9
8	Sweeps to opposite corner	8 8 8 8 8	8 8 8 8 8	8 8 8 8 8	8 8 8 8 8	8 8 8 8 8	8 8 8 8 8
7	Starts in corner	7 7 7 7 7	7 7 7 7 7	7 7 7 7 7	7 7 7 7 7	7 7 7 7 7	7 7 7 7 7
6	Holds broom with both hands	6 6 6 6 6	6 6 6 6 6	6 6 6 6 6	6 6 6 6 6	6 6 6 6 6	6 6 6 6 6
5	Place dustpan aside	5 5 5 5 5	5 5 5 5 5	5 5 5 5 5	5 5 5 5 5	5 5 5 5 5	5 5 5 5 5
4	Chairs on table, if any	4 4 4 4 4	4 4 4 4 4	4 4 4 4 4	4 4 4 4 4	4 4 4 4 4	4 4 4 4 4
3	Move items from floor	3 3 3 3 3	3 3 3 3 3	3 3 3 3 3	3 3 3 3 3	3 3 3 3 3	3 3 3 3 3
2	Go to designated room	2 2 2 2 2	2 2 2 2 2	2 2 2 2 2	2 2 2 2 2	2 2 2 2 2	2 2 2 2 2
1	Locate broom/dustpan	1 1 1 1 1	1 1 1 1 1	1 1 1 1 1	1 1 1 1 1	1 1 1 1 1	1 1 1 1 1
		0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
	Codes:						
	Location						
	Date						

Figure 5 - Traditional Training Prompting Script.

### 1.3. Framing the Problem

Persons with cognitive disabilities often cannot live independently. They cannot do so because their disabilities prevent them from performing core ADLs without human support (i.e., on their own). This research aims to provide a socio-technical environment to replace the need for a caregiver’s presence in accomplishing core ADLs, so people with cognitive disabilities can live more independently. What does this replacement for direct human support entail? Such a system needs to provide support specific to the person with cognitive disabilities, and responsive to the situation with respect to accomplishing the goal. By involving caregivers in tailoring external support for the user with cognitive disabilities one could provide an effective replacement to a caregiver's presence while performing the core ADLs that a person with cognitive disabilities requires for independent living. The resultant system would then be responsive to the environment, user, and goal.

## 1.4. Goal of the Research

The goal of this research is to explore a principled approach to designing and evaluating a *high-functioning*<sup>3</sup> socio-technical environment for persons with cognitive disabilities and their support community (e.g., caregivers) that surmounts the configuration/abandonment problem and safely and effectively supports task completion in the real world. To do so, several prototype parts of the system involving the MAPS prompter and script design environment have been developed. The intent of this complex system is to provide a research platform to explore and assess its use in the real world as a support for independent living [Farber, 1962].<sup>4</sup>

This research addresses the particular interface requirements for users with cognitive impairments, views the configuration and other caregiver tasks as different and equally important requirements for a second user interface, and applies techniques such as task-oriented design [Lewis & Rieman, 1993]. Thus the MAPS system can be considered to have a *dual user interface* [Carmien & Kintsch, 2006], one for the caregiver and a second for the person with cognitive disabilities. Both interfaces share the same fundamental understructure, but present radically different affordances to the two sets of users. The configuration environment employs web-based script and template repositories so data can be designed and shared by assistive technology experts and caregivers of different abilities and experiences. The design of the script design environment—a tool that enables caregivers to create, store, and edit scripts—presents problems particular to metadesign, a process for creating new media and environments that allows users to act as designers [Fischer & Scharff, 2000].

In most applications of metadesign, the designer and end user are one and the same person. In this application domain, the MAPS system is a tool that allows users (caregivers) to create systems that are used by other users (persons with cognitive disabilities). This presents unique research challenges that

---

<sup>3</sup> High-functioning in this context means a computationally based tool with complex structure that is very flexible in configuration and use.

<sup>4</sup> Additionally this research seeks to embody the pushing back of the frontiers of science, reflecting the well-known aphorism: “Knowledge is Good.”

have not been deeply explored previously. The design process for the MAPS script design environment primarily utilized a participatory design approach [Ehn, 1989; Schuler & Namioka, 1993] and explored concepts and principles from domain-oriented design environments [Fischer, 1994; Fischer, 1999a], user modeling [Allen, 1997; Fischer, 2001; Kobsa, 2001], and software reuse [Ye & Fischer, 2002] to make this design task more tractable and the resultant application usable with minimal training for caregivers.

Having designed the components of the proposed socio-technical environment, the next steps were to perform a series of studies of the introduction of MAPS to dyads of caregiver and persons with cognitive disabilities. These dyads were recruited from three typical populations: home, transition training provided by the local school system special education teams, and group homes. Ethnographic techniques were employed to understand and analyze current home life and the introduction of the MAPS system.

This research was driven by three related topics of interest:

- A fundamental understanding of how people with moderate to severe cognitive disabilities perceive and use information in prompting systems for tasks on mobile handheld devices
- User-centered development of a caregiver design environment that supports mobile device customization, personalization, and configuration; and
- A study of the process of adoption of the MAPS socio-technical environment by dyads of a person with cognitive disabilities and a caregiver.

The target populations for MAPS are individuals with cognitive disabilities in the mild (IQ 50-55 to 70) and upper range of moderate (IQ 35-40 to 50-55) [American Psychiatric Association. Task Force on DSM-IV., 2000], as well as the caregivers who would compose MAPS scripts. In this document, the caregiver will be referred to as the **caregiver** and the person with cognitive disabilities will be referred to as the **client**.

The research problems associated with the design and implementation of an effective and usable script are multi-leveled and range from basic research in cognitive science, to system-level issues in

interface protocols, HCI, and user modeling, to ethnographic studies of technology use with a population that may not be able to express themselves cogently (see Table 1). Even though “best practice” traditions in the field of assistive technology [Epstein et al., 2001] exist, little research has been done in the fields of image recognition for users with cognitive impediments and appropriate verbal structure of prompts. A recent study in these areas [Carmien & Wohldman, 2006] has provided preliminary results that support the hypothesis that different image attributes produce significantly different recognition times.

**Table 1: Summary of Research Agenda**

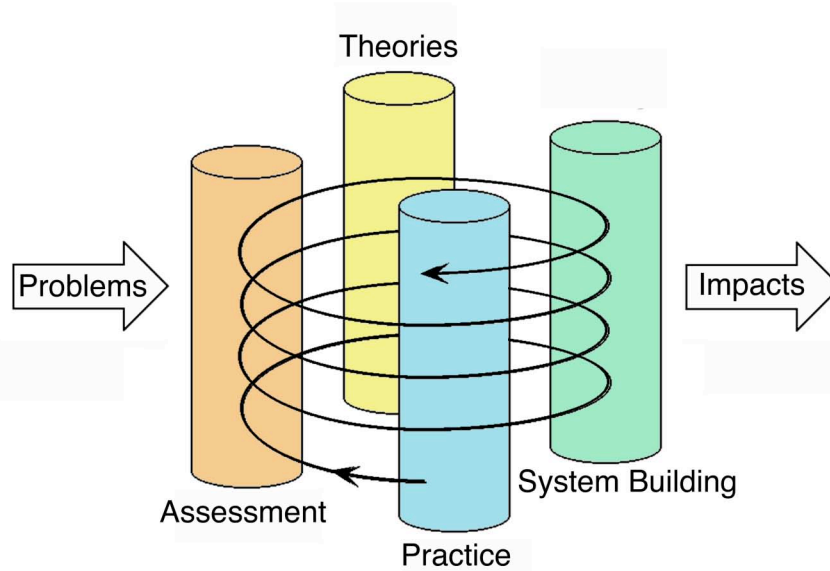
Key challenges	Research questions	Approaches
Creating a useful and usable design environment for non-technical users (caregivers) to create computationally complex multimedia scripts.	How can one design a multimedia script-editing environment such that relative computer illiterates can compose scripts?	<ul style="list-style-type: none"> <li>• <i>Metadesign</i></li> <li>• <i>Participatory design</i></li> <li>• <i>Task-centric design</i></li> <li>• <i>Ethnographic analysis</i></li> </ul>
Choosing the appropriate task, segmenting it into existing internal scripts, and performing the mechanics to capture the prompt elements.	How can caregivers create usable task-support scripts: choosing the task, appropriately segmenting the task into steps, taking pictures and recording prompts, and installing the script on the prompter?	<ul style="list-style-type: none"> <li>• <i>Task segmenting video</i></li> <li>• <i>Task segmentation tutorial</i></li> <li>• <i>Script templates</i></li> </ul>
Studying use of scripts on prompter in real tasks	How can the steps in tasks be represented such that they provide the right support for specific individuals and environments to accomplish ADLs?	<ul style="list-style-type: none"> <li>• <i>Scenario development</i></li> <li>• <i>Image recognition studies</i></li> <li>• <i>Distributed cognition</i></li> <li>• <i>Symmetry of ignorance</i></li> </ul>

## 1.5. Research Activities

To understand and develop useful theoretical frameworks for the two problems of metadesign and distributed cognition/situated action in this unique domain required a four-step process (see Figure 6):

- Study the domain, both the sets of users (caregivers and persons with cognitive disabilities) and existing technological approaches. [Assessment]
- Gain an understanding of HCI theoretical frameworks and methodology for similar problem domains. [Theories]

- Design a socio-technical environment that addresses the problems discovered in the study of the domain and exploits the strengths of extending the HCI approaches into this domain. [System Building]
- Evaluate the process of introduction and adoption of the resultant system in realistic usage by person with cognitive disabilities/caregiver dyads. Analyze the accumulated data and explicate the new insights as additions to the body of HCI frameworks. [Practice]



**Figure 6 - Research Process**

This research approach can be seen at the macro level as described above and also at a micro level, which informed the iterative appraisal to the construction of the MAPS system out of subcomponents. An example of this is the design process resulting in the MAPS prompter: existing prompting applications and systems were studied, analysis of the process of person with cognitive disabilities being prompted through tasks were informed by theoretical approaches, a system was designed and used in the lab at first, then in the world, and this use was assessed.

The MAPS system comprises a PDA prompter and a PC-based script design environment tool. In this dissertation, **MAPS** with no modifier refers to the socio-technical environment as a whole, **MAPS-**



**design-environment** refers to the caregiver script-design-environment (also known as the script editor), and **MAPS-prompter** refers to the mobile prompting environment used by person with cognitive disabilities end user.

The full scope of developing the research platform, beyond implementing and evaluating the MAPS-prompter and MAPS-design-environment, includes evaluating the system in use. This last part of the research activities involves evaluation of the suite of applications in controlled, laboratory experiments and in the real world. The evaluation of the system in the real world, with real users and meaningful tasks, is driven by the dichotomy between rigor and relevance in research [Schön, 1983]. On the one hand, rigor reflects a need for experiment reproducibility, but gains this by rigidly controlling all aspects of system testing, which means that the chaos of the real world is excluded. The results may be scientifically valid but not relevant in any meaningful way to the real use of the system. On the other hand, relevance, which is the argument of ethnography and case studies, has meaning because the use is studied in context. However, the results might not be generalizable, and numbers and reproducibly are limited. This study starts with the relevant side of the spectrum and thus provides the ground that may facilitate further rigorous research.

## **1.6. Reading Guide**

In this dissertation, chapter one defines the problem space, offering an introduction to assistive technology, persons with cognitive disabilities, and prompting. It broadly describes the proposed solution and the research environment for the development of MAPS. It then details the scope of the research questions raised. Finally, this chapter lays out the approaches used to answer the research questions.

Chapter two starts by discussing the cultural matrix of the dilemma in which developing assistive technology is embedded. Following this are further definitions of assistive technology and the key stakeholders—persons with cognitive disabilities and their caregivers. Finally, it discusses the dynamics and details of high-functioning AT, as well as the particular dilemma of high levels of abandonment.

Chapter three defines the HCI topics that together form the framework for this research. Following an overview of the chapter, the enclosing approach for study and socio-technical environments are discussed, and then the three main topics that comprise the framework—distributed cognition, metadesign, and symmetry of ignorance—are introduced and discussed with respect to how a theory, a principal, and a design methodology interact to support the MAPS system design and evaluation.

Chapter four discusses related work and describes several systems that are similar to MAPS. First there is a review of the literature about computationally enhanced prompting systems. Following this is a discussion of existing prompting systems and a detailed analysis of the AbleLink Pocket Compass and Task Builder systems.

Chapter five looks at the design of the MAPS system, starting with a discussion of the use of scenarios as a foundation for design. This chapter presents several example scenarios to both illuminate the design process and illustrate the operation of the system as it was implemented. Next, the design rationale and parameters for designing the two principal parts of MAPS, the prompter and the script design-environment, as well as a discussion of hardware are presented. The design philosophy of “low hanging fruit” is introduced, as well as the problems with implementing program forking and looping both with regard to implementation in the scripter and prompter and issues with training caregivers to program without actually being programmers. The technical details of the script-design-environment and scripter as well as the underlying architecture and the supporting database are discussed next, with particular emphasis on the details of the video help system.

Chapter six describes the assessments of MAPS in the design process, in the lab, and in the field. A section on methodology details usability testing and the use of ethnographic techniques, participant observation, and interviews. The process of iterative design and assessment in the coding of the prompter and script design-environment is discussed, with particular emphasis on the script design-environment with respect to users with marginal computer skills. Following this is a short section that presents an experiment examining the relationship between images on the prompter and accurate

selection of objects. The rest of the chapter is devoted to the field trials of the MAPS system, detailing the participants and presenting aggregate data from the qualitative studies of task support in the real world. A short set of heuristics for the design of cognitive orthotics and assistive technology in general is presented. Finally, the chapter proposes some conclusions resulting from the field.

The next chapter, seven, presents theoretical contributions resulting from the special adaptations into which the framework presented in chapter three evolved as a result of this study with MAPS and the special population for which it was designed. Each of the constituent parts of the framework are considered in turn. Distributed cognition theory contributed the perspective of internal and external scripts as well as tools for living and tools for learning. The principle of symmetry of ignorance and stakeholder analysis contributed a unique approach to system design and the concept of dual user interfaces. Metadesign was modified with the concepts of design over time and multiple users and personas. Finally, a short discussion of how these insights may be contributed back into the body of HCI thought is presented.

Chapter eight presents possible future work inspired by the lessons learned in the design and assessment phases of the research. A section on expanding the functionality of MAPS describes ways to provide computational support to process what caregivers are doing by hand. Following this are possible directions that the prototype MAPS/Lifeline [Gorman, 2005] system could take to truly support *active distributed support* [Carmien et al., 2005b]: context awareness, script annotation, and active caregiver involvement that would constitute error trapping and correction. Finally, some suggested extensions to the “image experiment” are suggested.

The final chapter lists the conclusions of the research and reiterates the themes presented in earlier chapters. Two generic questions of the validity of this approach are discussed, and the dissertation ends with an overview of the research and its importance to the field of assistive technology. Appended to the dissertation are a glossary, list of abbreviations, a reference section, and a set of supporting documents.

## 2. The Problem

*The moral test of a government is how it treats those who are at the dawn of life, the children; those who are in the twilight of life, the aged; and those who are in the shadow of life, the sick and the needy, and the handicapped.*

—Hubert Humphrey

*A nation's greatness is measured by how it treats its weakest members.*

—Mohandas K. Gandhi

We live in a world that has become increasingly saturated in information, sometimes in knowledge, but rarely in wisdom. The motivation for this work is based on a vision of humanness that is about quality of life—and poetry. Historically, persons with cognitive disabilities have been shunned by society: retarded Aunt Karen, who is hidden from others in her town, glimpsed only when she is looking out the window; Thomas, who was born with Down's syndrome and who was placed into a facility for the mentally feeble to spend the rest of his life, warehoused; Sarah, who was mainstreamed through high school, but after graduation spent much of her life in a group home, never fulfilling her potential as an employee and citizen. The stories are ubiquitous and heartbreaking. Here, in the dawn of the twenty-first century, our technical mastery permits us to create systems that could empower the weakest, in addition to extending the reach and vision of the *most* able.

The goal is not just independence for persons with cognitive disabilities, but inclusion and engagement in society; and it is not just for their benefit, it is also for our benefit, the able-bodied and able-minded. When the full range of humanity is allowed a place in the daily parade of life, everyone benefits. Those without obvious impairment see that in the whole span of human abilities: the essential humanness is not lost—all have their own poem, their own song. This vision of the span of embodiments of being human inclines all toward a certain tenderness and poignancy that enables us to be, in turn, more human. The immediate goal of this work is to provide support for persons with cognitive disabilities to participate in society as far as aided-ability might allow.

This chapter introduces the details of an approach to this problem, starting with some definitions, and discusses abandonment of assistive technologies. Providing computational task support for persons with cognitive disabilities and caregivers requires: (1) providing a usable and successful tool that helps both members of the dyad to be more satisfyingly independent; (2) gaining insight into the nature of HCI problems from established perspectives; and (3), as a result of (1) and (2), being able to reformulate these HCI abstractions to gain insight for them as such and in the context of AT design.

## **2.1. Some Definitions**

### **2.1.1. Prompting**

Prompting is a technique used by adults and older children with cognitive disabilities [Aist, 1973; Reed, 1989] for both learning [Krantz et al., 1993] and performing a task. Prompting consists of breaking down a task into constituent parts and creating individual prompts, consisting of pairs of images and verbal instructions, for each step. A prompting script is a sequential set of prompts that, when followed, accomplish a task. Special education and rehabilitation studies focus on comparing techniques and creating a principled understanding of prompting approaches with a perspective of maximizing internal recall and unaided performance of the steps to complete a task by persons with cognitive disabilities [Reed, 1989]

Gerontologists have classified the skills needed to live independently as activities of daily living (ADL) [Lawton MP & EM., 1969] and instrumented activities of daily living (IADL). The difference between the two is that ADLs are focused on basic biological functions, such as dressing, eating, and bathing, and IADL functions concern skills that are more social in nature, such as traveling by bus or car, using a telephone, or shopping. The assistive technology and vocational rehabilitation literature often uses ADL to cover both sets of skills. Independent living transition professionals teach IADLs by prompting the person with cognitive disabilities through a task by verbally instructing the client through each step, either with or without instructional cards in a rehearsal mode; then the client will use the memorized instructions at use time. This study uses the term ADL to represent both ADL and IADL activities. Prompting studies provide a background for the study and design of computationally

based prompting systems. Several papers have been published on the topic of computerized prompting and individuals with cognitive impairments [Lancioni et al., 1999; Lancioni et al., 2000; Lynch, 1995].

The domain of special education and rehabilitation services focuses on studies comparing techniques and creating a principled understanding of prompting techniques with a perspective of maximizing internal recall and unaided performance of the steps to complete a task [Aist, 1973; Reed, 1989]. The choice of the right task from which to create a script is based on the abilities and needs of the person with cognitive disabilities; these may range from hand washing for persons with severe cognitive disabilities, to employment tasks for those with higher levels of memory and executive function. In any case, the task to be taught or aided with prompts needs to be small enough to memorize or implement on a mobile computationally enhanced prompter. Tasks are typically a few minutes to several hours in duration, and they concern a single topic. Appropriate tasks for prompting do not span the various items that might make up a larger more complex task (e.g. mopping this floor versus the set of tasks that constitute a janitorial job description). Key to the production of efficacious task scripts is the appropriate segmentation of the chosen task into atomic subtasks of appropriate granularity [Saskatchewan Learning - Special Education Unit, 2003; Snell, 1987]. The prompts must be geared to the cognitive level of the user (i.e., users must be able to hold the whole subtask in their minds), and capable of being accomplished in one single act. These sequences of actions are *internal scripts* [Carmien et al., 2006] (see section 7.2), and are cued by an external label, a set of which constitutes an *external script* [Carmien et al., 2006]. For some users, they may be as complex as “go to the post office and get stamps,” whereas for others, “get out two slices from the open bag of bread” may be an optimal segment size.

With the arrival of computationally based task support, the memorization and decision-making elements of the task could be offloaded to the device and the system that supported it. All the elements of the prompting activity were the same as in the traditional use: there was a notion of a task, and the task broken down into steps which could be prompted using images and verbal instructions. Missing from this implementation of prompting is the use of physical modeling and guidance that is often present in traditional use. Moreover computer-based prompting can add context awareness and other

forms of script annotation (see section 8.2.5), notions that are not applicable to traditional prompting. Prompting studies provide a background for the study and design of computationally based prompting [Lancioni et al., 1999; Lancioni et al., 2000; Lynch, 1995]. Distributed cognition [Fischer, 2003; Fischer, 2006; Hollan et al., 2001] (see section 3.3) is the notion that in most cognitive acts there are elements in the environment that support and extend the individual's knowledge and ability to act in the world. The distributed cognition framework describes how the addition of cognitive support in the world often changes the nature of the task [Norman, 1993]; in this case, the script playing on the handheld device transforms the skills needed for task accomplishment from memory and executive functionality to those involved with using the prompter and following its instructions.

### **2.1.2. Assistive Technology**

Broadly speaking, any artifact that is used to compensate for deficits in ability falls into the realm of assistive technology. A legal definition of assistive technology includes:

Any item, piece of equipment, or product system, whether acquired commercially off the shelf, modified, or customized, that is used to increase, maintain, or improve functional capabilities of individuals with disabilities. AT service is directly assisting an individual with a disability in the selection, acquisition, or use of an assistive technology device.[1988]

Assistive technologies may include mechanical, electronic, and microprocessor-based equipment; this includes microcomputers, electronic communication devices, and other sophisticated devices. However, assistive technologies also include non-mechanical and non-electronic aids; for example, a ramp to replace steps would fit in this category [LaPlante et al., 1997]. Whole systems can also fall into the category of assistive technology; examples include specialized instructional materials such as large print for persons with visual impairments, as well as specialized services and strategies. Assistive technology also bridges *augmentative and alternative communication systems* [Beukelman & Mirenda, 1998], defined as systems that aid individuals who are not independent verbal communicators. These systems can include speech, gestures, sign language, symbols, synthesized speech, dedicated communication aids, or microcomputers.

This research is concerned with providing aid to persons with cognitive disabilities. Support for this particular population often implies providing support also for the persons who provide help in their day-to-day life, caregivers, especially for more complex devices. Because there are two people, at minimum, who are to be supported, the assistive technology design process is concerned not just with a person but with a socio-technical environment [Mumford, 1987]. The design must effectually fit not only the end user, but also the caregiver and the unique relationship between these two people.

Several domains of professional resources support design and evaluation of assistive technology. The Rehabilitation Engineering and Assistive Technology Society of North America (RESNA) is an interdisciplinary association of people with a common interest in technology and disability. The Cognitive Disabilities and Assistive Technology special interest group (SIG 20) of RESNA is a group of academics and professionals interested in creating and evaluating AT for persons with cognitive disabilities. The Association for Computing Machinery (ACM) Special Interest Group on Computers and the Physically Handicapped (SIGCAPH) promotes the professional interests of computing professionals focused on the research and development of computing and information technology to help people with disabilities. The Americans with Disabilities Act of 1990 [ADA, 1990] mandates support for equal opportunity for persons with disabilities in employment, state and local government services, public accommodations, commercial facilities, and transportation which creates an environment encouraging design and dissemination of effective AT systems.

### **2.1.3. The End User – The Person with Cognitive Disabilities**

The Cognitive Disabilities and Assistive Technology special interest group of RESNA defines cognitive disability as follows:

By “cognitive disability,” we mean any disability that affects mental processes. A list of examples includes, but is certainly not limited to: mental retardation, attention deficit disorder, dyslexias, aphasias, language delay, dementias and other psychiatric and behavioral disorders. [LoPresti, 2003]



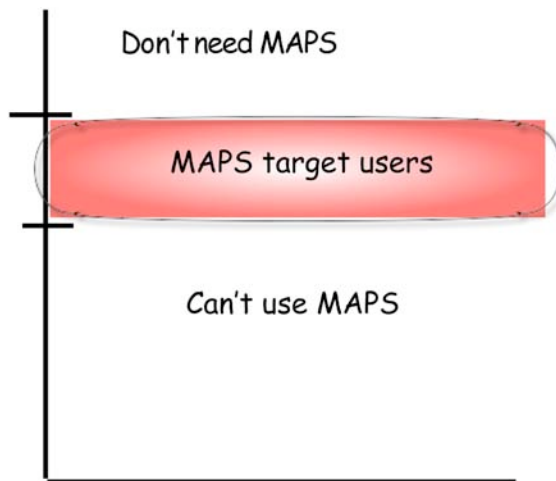
An individual with cognitive disabilities is defined by the Diagnostic and Statistical Manual of Mental Disorders IV (DSM-IV) [American Psychiatric Association. Task Force on DSM-IV., 2000] as a person who is “significantly limited in at least two of the following areas: self-care, communication, home living social/interpersonal skills, self-direction, use of community resources, functional academic skills, work, leisure, health and safety.” Four different degrees of cognitive disability are defined: mild, moderate, severe, and profound. The target populations for MAPS are cognitively disabled individuals in the mild (IQ 50-55 to 70) and upper range of moderate (IQ 35-40 to 50-55) [The American Association on Mental Retardation, 1992]. The American Association on Mental Retardation (AAMR) currently defines mental retardation as having an IQ of below  $70 \pm 5$  and the disability originating before age 18 [The American Association on Mental Retardation, 2002]. In this work the DSM-IV definition is used

Concretely specifying the target user group with gross metrics is difficult, however, because:

- IQ and other numeric measurements of cognitive ability are not fine grained enough to measure whether a given person could accomplish ADLs with or without task support [Luckasson & Reeve, 2001]. Note that IQ may provide a bottom limit (perhaps 50) below which ADLs, even with task support, could be accomplished.
- This population often has multiple disabilities, and cognitive deficiencies are frequently combined with sensory and motoric problems, which are often made more complex by psychological issues.
- Specifying detailed deficiencies does not convey the fact that persons with cognitive disabilities may have normal abilities in other areas, which may be sufficient to accomplish with assistance what they otherwise could not.

Because of the above, and following the lead of many AT professionals, a better approach to qualifying the population best suited for using MAPS is to look in terms of specific disabilities and abilities [McCue et al., 1994]. To benefit from this kind of task support, the cognitive ability of the end user (person with cognitive disabilities) must fall into the band between being just able to use the handheld prompter and follow its directions and still needing to have prompting support to accomplish ADLs. In broad terms, the MAPS system can provide the support for that set of people who are almost able to live independently except for the mnemonic and executive functionality that MAPS can provide

(see Figure 7). More concretely, the user must be able to follow instructions (the verbal and visual *prompt* [Epstein et al., 2001] ), be able to operate the MAPS-prompter successfully, and be motivated to live independently with the aids referenced above. They need not be able to read nor communicate fluently by talking.



**Figure 7 - Range of Target Population for MAPS**

The lack of executive and mnemonic functions constitute the primary deficiencies that drive the need for a system such as MAPS. Executive function is defined as follows:

The term executive function describes a set of cognitive abilities that control and regulate other abilities and behaviors. Executive functions are necessary for goal-directed behavior. They include the ability to initiate and stop actions, to monitor and change behavior as needed, and to plan future behavior when faced with novel tasks and situations. Executive functions allow us to anticipate outcomes and adapt to changing situations. The abilities to form concepts and think abstractly are often considered components of executive function. [Barry, 2006]

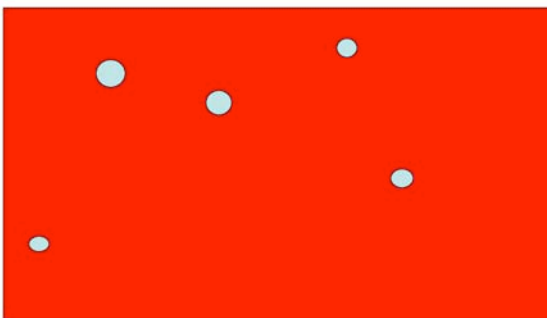
Norman and Shallice have outlined five situational types for which routine activation of behavior would not be sufficient for optimal performance [Norman & Shallice, 2000]:

- Those that involve planning or decision-making.
- Those that involve error correction or troubleshooting.
- Situations for which responses are not well learned or that contain novel sequences of actions.
- Dangerous or technically difficult situations.
- Situations that require overcoming a strong habitual response or resisting temptation.

The executive function is that subsystem of cognitive functioning that supports planning and responding to novel situations [Packer, 2004]. This functional description highlights one portion of the missing abilities of persons with cognitive disabilities who are good candidates for MAPS task support.

The mnemonic deficiency can be of several types [Cole & Dehdashti, 1998a]. A deficit of long-term memory can cause problems with attempting to do a task that has been mastered in the past; all the subtasks are still in the repertory of abilities, but the larger set of cues are not internalized. Deficits in short-term memory can result in “senior moments,” in which the person with cognitive disabilities has “lost track” of the steps in a given task. MAPS is designed to supplant existing functions by externalizing the memory required to perform the task.

Each individual in this population presents a unique set of needs and abilities. This actually could be said of anyone, but in this case, persons with cognitive disabilities often also have sensory, motor, and psychological disabilities. Further, the combination of needs and the specific context that supports (or does not support) them produces deeply unique instances that often need to be addressed by nontrivial customization of technical support. This design problem-space is termed the *universe of one* problem [Fischer, 2001], in that a solution for one person and their community and environment will rarely flawlessly work for another [Cole, 1997]. The steps involved in any given task (whose invoked behavior is called an *internal script* [Carmien et al., 2006]) may be nearly the same across many users and instances, but the image supporting the subtask in any prompt step as well as the verbiage describing the proper action to take must be specific to *this* task, *this* context, and *this* user. Thus, for this population a high degree of customization, approximating unique custom interfaces and functionality, is a minimum requirement [Cole & Dehdashti, 1998b].

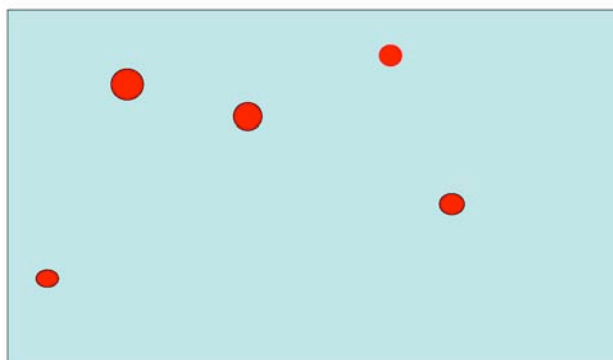


**Figure 8 - Abilities in Sea of Deficits**

Islands of abilities in seas of deficits: Unexpected abilities that can be leveraged

Exacerbating the universe of one condition are the unexpected islands of abilities (see Figure 8 [Cole, 2006]); some persons with cognitive disabilities can have unexpected skills and abilities that must be leveraged to ensure a better possibility of task accomplishment. Similarly, some persons with cognitive disabilities have had unexpected defects of abilities (see Figure 9 [Cole, 2006]); often these defects are related to time of day, level of previous activity, and metabolic state. Accessing and addressing these unexpected variations in skills and needs, particularly with respect to creating task support, requires an intimate knowledge of the client that caregivers are uniquely suited to provide.

There is much concern about this population's ability to use sophisticated hand computers. Fortunately, the current environment is particularly supportive for the introduction of this technology to this population, as young adults with cognitive disabilities have been raised with a plethora of electronic devices – cell phones, MP3 players, and computers. These computational devices are seen as just another utility, like hot water taps or the telephone. Handling personal devices of this kind demonstrates their ability to consistently correctly use and safely carry the MAPS-prompter (see section 6.2).



**Figure 9 - Deficits in Sea of Abilities**

Islands of deficits in seas of abilities: Causes of unexpected activity failures

#### **2.1.4. Caregivers**

A caregiver is defined as a “*person providing some form of assistance to another person who is dependent on that help due either to medical reasons or to a simple inability to conduct their routine ADLs*” [National Educational Association, 2004]. Another perspective defines a caregiver as “*someone who provides assistance, generally in the home environment, to an aging parent, spouse, other relative, or unrelated person, or to an ill or disabled person of any age. A caregiver can be a family member, friend, volunteer, or paid professional*” [Frankford Hospitals, 2004]. This study considers caregivers to be either family members (or otherwise tied by emotional considerations) or helping professionals. In either case, the caregiver and the end user are considered to be a canonical pair [McCallion, 1997] such that the entire system must be designed to fit both parts for their respective roles: the person with cognitive disabilities as the user of the mobile handheld prompter and the caregiver as the script designer. In this sense, they are unique partners in a dual interface project [Piuchno, 2003].

In studying this population, it soon became obvious that the role of caregivers in day-to-day life is key to a successful and satisfying life being held by persons with cognitive disabilities. Besides providing social support, caregivers are key to successful adoption of support devices in that they shoulder the burden of adapting the devices to the changing abilities and needs of users with cognitive

disabilities. Successful design of the configuration and modification aspects of computationally based independence augmentation devices requires that they be very easy to use; our goal is to approximate automatic teller machine (ATM) ease-of-use. In our design, process caregivers were assumed to have computer skills on the level of being able to compose a letter on a word processor.

## 2.2. Independence

The goal of support for this population is *independence* [Carmien et al., 2005a]. What does independence mean in this context? To be more independent, for persons with cognitive disabilities, is to extend the ability to choose and do as much of ADLs as possible. It does not mean isolation from others, so independence also has an inclusive quality as well as a quality of socialization. To become as independent as possible implies empowerment and learning. Many persons with cognitive disabilities currently are employed, but only with constant or frequent support and supervision of a professional job coach—for them, independence means being able to do the job on their own. Many persons with cognitive disabilities have all the skills to prepare complex meals, but are currently cooking only with help or supervision—for them, independence means cooking on their own. Many persons with cognitive disabilities have helped caregivers to shop for the weekly groceries—for them, it means being able to go to the local grocery store and select groceries and pay for them. Employment means being able to participate in group outings with *their own* money; it means cooking a meal for friends and eating together; it means seeing and talking to the familiar checkout woman as a peer, and meaningfully participating in homecare activities.

At the heart of the research design goals is the intent to extend the ability of the user with cognitive disabilities to act “independently” while providing the caregiver role of the dyad a sense of security in the safety of the user. However, from a commonsense perspective, it may seem that MAPS provides only the *illusion* of independence for the user. In fact, users would be able to move outside their homes independently only with the aid of the MAPS system, but they would be dependent on the system for *independent* action. An outside observer might say that because they would be unable to do the task without MAPS, they were *dependent* on MAPS for even the attempt to leave their homes at

all. However, the MAPS end users are not so different from the “normal” population would we be able to travel in a strange city without the ability to read the signage or ask questions in the only language we knew? So from this perspective we are all living in the illusion of our *own* independence, the only difference being that the elements that we use to create for ourselves the sense of self-dependency are below the level of our day-to-day awareness [Trungpa, 1988 (reprint)]. It is from this perspective that we approach the frameworks discussed below.

### 2.3. Abandonment

Device rejection is the fate of a large percentage of purchased assistive technology [King, 1999; King, 2001]. Caregivers report that difficulties in configuring and modifying configurations in assistive technology often leads to abandonment<sup>5</sup> [Kintsch & dePaula, 2002], an especially poignant fate considering that these sorts of systems may cost thousands of dollars. Some experts estimate that as much as 70 percent [Martin, 1999; Reimer-Reiss, 2000] of all such devices and systems are purchased and not used over the long run, particularly those designed as a cognitive *orthotic* [LoPresti et al., 2004]. Causes for abandonment have many dimensions; a study by Phillips and Zhao reported that a “change in needs of the user” showed the strongest association with abandonment [Phillips & Zhao, 1993]. Thus, those devices that cannot accommodate the changing requirements of the users were highly likely to be abandoned. Although this study did not address configuration issues directly, it follows logically (and is confirmed by interviews with several AT experts [Bodine, personal communication, 2003; Kintsch, personal communication, 2002]) that an obstacle to device retention is difficulty in reconfiguring the device. A survey of abandonment causes lists “*changes in consumer functional abilities or activities*” as a critical component of AT abandonment [Galvin & Donnell, 2002]. A study by Galvin and Scherer states that one of the major causes for AT mismatch (and thus abandonment) is the myth that “*a users assistive technology requirements need to be assessed just once*” [Scherer & Galvin, 1996]; ongoing re-assessment and adjustment to changing needs is the appropriate

---

<sup>5</sup> There is another kind of abandonment, which is not using the system or device because the need no longer exists. This “good” abandonment of AT is not in the purview of the current study.

response. A source for research on the other dimensions of AT abandonment, and the development of outcome metrics to evaluate adoption success is the ATOMS project at the University of Milwaukee [Rehabilitation Research Design & Disability (R<sub>2</sub>D<sub>2</sub>) Center, 2006].

Successful AT design for this population must support the interface requirements for users with cognitive impairments as well as view configuration and other caregiver tasks as different and equally important requirements for a second user interface [Cole, 1997]. To address this problem, MAPS embodies the notion of a *dual-user interface* system [Carmien & Kintsch, 2006]. In such a system, the design pays as much attention and care to the interface for the caregiver as it does to the interface for the device used by the client. One proven approach applies techniques such as task-oriented design [Lewis & Rieman, 1993] to mitigate technology abandonment problems. Research [Fischer, 2001] and interviews have demonstrated that complex, multifunctional systems are the most vulnerable to abandonment due to the complexity of the many possible functions. Therefore, the initial goal was a simple system that does one (or few) things very well for a large range of users/caregivers with an interface that is exceptionally easy to use initially. This same system should be able to support complex scripts. The overarching goal is a system that can easily create useful scripts immediately yet without a low ceiling on its functionality, and also has the potential to make arbitrarily complex scripts.



### 3. Theoretical Framework

#### 3.1. Overview

In examining the problems of socio-technical support for persons with cognitive disabilities, an interesting pattern emerges. Certainly, the views of distributed cognition, symmetry of ignorance and metadesign can all shed light on supporting both members of the target group; however, because of their particular defining characteristics, many of the tenets of the theoretical frameworks must be used or looked at in a manner different from that in which they are originally articulated. For instance, distributed cognition theory emphasizes that computational mediating artifacts can provide support to the cognitive process of a user; however, what they must do for our population is supply the *missing* cognitive functionality for the user. Because the cognitive abilities of the user are different than the typical “H” of HCI, for the theories to be useful, they must be modified to fit this specific context. Another way that these approaches must be modified to usefully fit the users is that in almost all cases when referring to the user, reference should really be to a pair of users, the client and the caregiver, that constitute the target for the design. Sometimes the role of the single caregiver is further composed of a team—a primary caregiver and a set of professionals who assist the primary caregiver in providing and extending support for the person with cognitive disabilities..

This chapter discusses three HCI motifs, which span classes from theory (distributed cognition), to principle (symmetry of ignorance) to design methodology (metadesign). These three constitute the theoretical foundations for the design of MAPS.

In the discussion of the use of prompting to achieve more independence (see section 2.1.1) the theoretical model of distributed cognition fits the situation well and provides both analytical and prescriptive insights. The process of analytically exploring the existing practice of prompting and the proposed computational implementation of prompting exposes a need for an additional perspective on classic distributed cognition. Distributed cognition theory posits a linear model of knowledge construction for both the person/artifact and the person/person cognitive act. For this special population the theory needs to accommodate the unusual relation of person with cognitive disabilities

and caregivers. This could be provided by the principle of symmetry of ignorance, which emphasizes that, for both socially constructed and artifact mediated distributed cognition, the parts of the solution lie amongst the two primary stakeholders. The caregiver can provide scripts and the client can provide the situated use of the scripts in action [Suchman, 1987]. Having reviewed the problem space with insights gained from the application of the principle of symmetry of ignorance to the base of distributed cognition theory, the system designer needs to think of how to implement a system that satisfies all the stakeholders. Metadesign, as a design approach, offers the sort of meta-affordances that can help the creation of such a system. MAPS is built on this framework of theory, principle and design methodology.

### **3.1.1. Design by Composition and Design by Modification**

In considering the caregiver's relation to the MAPS script-design-environment and eventually to the script running on the MAPS-prompter, there is a need to incorporate the fact that, even though caregivers are experts on persons who will use the MAPS-prompter, initially, they are most probably not experts on scripts. More specifically, they lack expertise in designing scripts used in a script driven support system. This problem is intimately connected with the abandonment issue because an initial failure of script creation may result in abandonment of the system. To address this concern one can turn to the ideas of design by composition and design by modification. Design by composition and by modification are issues that have been discussed at length in the world of software engineering [Ye, 2001], and the same need - to quickly and effectively produce computational tools - exists in this domain.

Design by modification refers to using existing successful designs and modifying them to suit new purposes. The MAPS script repository exists on an on-line database server containing about 170 successful scripts used by others and templates of scripts obtained from assistive technologists who created them to support the older, traditional script memorization techniques used with our target population. These templates range from "working at Mc Donald's" to "brushing your teeth" to "bowling" to "cooking muffins." By reusing successful scripts, caregivers can circumvent the explicit learning of task segmentation and "absorb" the skill by using the borrowed scripts as models.

Design by composition refers to the technique of building complex tools out of subcomponent parts that may be available as parts of a palette of components. Similarly, the MAPS script-design-environment provided the ability to reuse segments of successful scripts that can be used as-is because they are parts of existing scripts designed and tested as successful for this user doing this task. As an example, consider the class of scripts that implement bus trips for a given user; a majority of them will have the same set of prompts for getting ready to leave the house, and for traveling from the front door to the neighborhood bus stop. These two segments can be made into components (intermediate abstractions), such that the caregiver could select them from a palette of existing segments and begin q script design by using them [Fischer et al., 1998].

### 3.2. Socio-Technical Environments

Driven by a desire to study AT design and adoption, the unit of study needs to be not only the technology, and not only the persons using that technology, but the whole environment consisting of the system, the person's interactions between themselves and the technology and the environment in which the activity takes place. This kind of thinking about socio-technical environments and systems was first done by groups studying industrial work, and documented and created new ways of thinking about production efficiency that went beyond the Tayloristic fixations of time and motion. This perspective drew from ecological thinking and general systems theory to focus on the process rather product. Mumford summarized the development of this approach and collected a set of socio-technical principles [Mumford, 1987]. Several of her principles are pertinent to the MAPS environment, another example of how existing theoretical foundations needed to be modified to apply to the MAPS dyad approach. Among these principles several are particularly applicable to this design space:

- (*principal 2*) The principle of minimal critical specification. “*No more shall be specified than is absolutely essential ...*” What is essential needs to be identified
- (*principal 5*) The principle of boundary location This principle is that boundary location must be chosen with care and the boundaries require management

- (*principal 9*) The principle of incompleteness. This principle states that design is an interactive and continuous process. [Mumford, 1987]

The principles of incompleteness and minimal critical specification are good guidelines for designing a script design tool. These requirements support the intimate interrelations between the technical parts of the environment and the end-user dyad.

Socio-technical designers always try to see complex systems design as a unified process. This means taking account of technical, economic, organizational and social issues at every stage of the design process. [Mumford, 2000b]

The initial assessment of MAPS using ethnographic tools reflects this approach. Only after understanding the socio-technical environment as a whole can more traditional optimization of the different parts of the system be attempted. In understanding the activities in a socio-technical environment, the focus is not necessarily on what is currently going on but to see the goal with respect to how the initial behavior in the environment attempts to reach that goal. The socio-technical approach is particularly appropriate for this population as the need for a tool that is deeply personalized lends itself to studying the environment *and* the technology as the users see and interact with it and with each other.

### **3.3. Distributed Cognition**

Gregory Bateson remarked that memory is half in the head and half in the world [Bateson, 1972; Pea, 1993]. We exist in a world full of examples of this distributed cognition: the shopping list that “remembers” for us, the speedometer on our car, the position of the toggle on our light switch (up for on), the very words that we are reading right now. Distributed cognition is the view that both the internal assets of the person and the cultural structures and artifacts support the intelligence or cognition in a given human action. The knowledge and skill put into building a house are drawn not only from the builder’s internal knowledge and abilities, but also from his tools, tape measure, saws, and hammer, and also his cultural support, language, and customs. Acts and knowledge are not

constructed unilaterally. An interesting question is “*Where is the boundary between my knowledge and the context that supports my knowledge?*” [Salomon, 1993]. Distributed cognition is an approach that views the cognitive act as a result of a system comprising an actor, the artifacts in the actor’s environment, and possibly other people. These artifacts can be as concrete as a notebook and as ethereal as language. Viewing cognition in this fashion can enable analysis and prediction of cognitive behavior that has a basis beyond the solitary human mind.

Distributed cognition attempts to analyze problem-solving behavior with a unit of analysis that spans individuals, artifacts, and others [Hollan et al., 2001]. The artifact can provide external support (i.e., amplification and transformation) for cognitive acts that may be beyond the ability of the unaided mind (e.g., cube roots). The artifact may be providing a true cognitive orthotic role, as in the MAPS prompting system, or may just extend sensory abilities, as in the classic blind man’s stick in Gregory Bateson’s example:

But what about “me”? Suppose I am a blind man, and I use a stick. I go tap, tap, tap.  
Where do I start? Is my mental system bounded at the handle of the stick? Is it bounded by my skin? Does it start halfway up the stick? Does it start at the tip of the stick? [Bateson, 1972]

Distributed cognition is a cognitive science model, in the sense that it basically is concerned with the individual’s internal cognitive processes and the support/extension that artifacts can provide, in contrast with the sociological/ethnologist view that sees the user and artifact as part of a system of relationships [Suchman, 1987]. One view of distributed cognition is that it is attempting to describe how distributed units are coordinated; how information is represented, stored, and transformed; and in turn how the representation of information transforms the task at hand [Pea, 1993] In this sense, the representation and the computational mechanism that manipulates the representation become part of the cognitive process. But this transformation is not a static event. “*In distributed cognition, one expects to find a system that can dynamically configure itself to bring subsystems into coordination to accomplish various functions*” [Hollan et al., 2001]. Therefore, the system is often dynamically

interacting between the agents and objects, each modifying and mutually supporting the effort toward the system goal.

From the distributed cognition perspective, this process of “external cognition” [Carroll, 2003] consists of agents creating and using information in the world, rather than simply within their heads, to do three key things: (1) externalize information to reduce memory load (e.g. reminders); (2) simplify cognitive effort by “computational offloading” onto an external media; and (3) allow the tracing of changes, for example over time and space, through annotation [Perry, 2003]. The external cognitive artifacts or mediating artifacts that support this offloading increase memory capacity; in addition, the representation held in the artifact may “*not simply augment, or amplify existing human capabilities. Rather, they transform the task into a different one*” [Norman, 1993].

To analyze a task or environment from a distributed cognition perspective one needs to answer Roy Pea ‘s three questions [Pea, 1993]:

1. *What is distributed (i.e., different components of the problem-solving process as well as the product)?*
2. *What constraints govern the dynamics of such distributions in different time scales (e.g., microgenesis, ontogenesis, cultural history, and phylogenesis)?*
3. *Through what reconfigurations of distributed intelligence might the performance of an activity system improve over time?*

The process of deconstructing the problem with this framework can be useful in creating a system that distributes knowledge-in-the-world [Norman, 1990] by redistributing expert skills into a system. In this case, (1) *what is distributed* are mnemonic and executive triggers and content and the current skill set of the person with cognitive disabilities, (2) *the constraints on the system* are the timeliness and fit of the prompts to the current context, and (3) *the improvement of the performance over time* maps to error correction, script rewriting, and scaffolding concerns (see sections 7.2, 8.3, and 8.2.5).

By viewing the cognitive system as a system comprising an actor and mediating artifacts with the perspective of distributed cognition, one can look at goals and plans to attain these goals as being effected by a system comprising actors, singly or in groups (e.g., classes of actors), mediating artifacts,

and their interactions. There is no particular bias in this perspective toward human actors; all elements are evaluated on the same plane [Nardi, 1996a]. Distributed cognition looks for cognitive processes wherever they may occur, on the basis of the functional relationships of elements that participate together in the process. In distributed cognition, one expects to find a system that can dynamically configure itself to bring subsystems into coordination to accomplish various functions [Hollan et al., 2001].

### **3.3.1. Distributed Cognition: Artifacts and Other Representations**

Looking back to the larger issues of distributed cognition, it is necessary to decide the level of granularity that the analysis requires [Perry, 2003]. In the largest sense, the unit of analysis is the functional system—in this case, the task, client, and caregiver, as well as the artifacts and transformations that entail the task support system. From the designer's perspective, among the first things to analyze are any artifacts that are currently being used, before the introduction of the MAPS system (see section 6.1.2 on ethnography).

Having studied existing work practices, the next task is to design a system reliant on the principles of distributed cognition. Such a system would support the creation of artifacts that support the task as well as accommodate exactly how those artifacts would transform the task and support changes in the representational media as the task support system becomes more refined and fitted. Artifacts can span in complexity from scripts on the prompter to marks or pieces of tape on the controls of devices; all of these contribute to the cognitive act shared between the person with cognitive disabilities and the caregiver, and the task support system. The artifact chosen must match the needs and abilities of the user; for example, a person with limited sight would need to lean on the verbal prompt more to attain the same goal, and a person who has hearing problems may require more attention paid to the creation of visual prompts. Even within one modality of artifact support, the crafting of the details of the artifact must be well fitted. Consider the case of the sixteen-year-old with cognitive disabilities who has typical power issues with her mother. The prompts in her case should *not* be recorded by mom, and perhaps not even in a woman's voice.

### 3.4. Symmetry of Ignorance

The concept of symmetry of ignorance emerged from discussions among designers of systems as disparate as urban planning and industrial organization design in the mid-1980s [Rittel, 1984]. Symmetry of ignorance is a way of describing situations in which several participants or roles in an endeavor each individually has parts of the knowledge needed to accomplish the task, but none has enough to accomplish the tasks independently [Fischer, 2000]. An end user may know exactly what an application needs to do, but be unable to program, whereas a programmer may know how to develop robust applications, but in isolation creates unusable software.

In parallel to this notion, and complementing it, was the study of wicked problems [Rittel & Webber, 1984] [Simon, 1984]. Ill-defined design and planning problems can be labeled “wicked” (i.e., difficult) to contrast against the relatively “tame” problems of mathematics, chess, or puzzle solving. Wicked problems have incomplete, contradictory, and changing requirements, and solutions to them are often difficult to recognize as such due to the complex interdependencies. Typically, wicked problems have these characteristics:

- The problem is not understood until after formulation of a solution.
- Stakeholders have radically different worldviews and different frames for understanding the problem.
- Constraints and resources to solve the problem change over time.
- The problem is never solved.
- Solutions to wicked problems are typically better, worse, or good enough (satisficing).

This wicked problem space requires both primary symmetrical holders of knowledge, the client and caregiver, to contribute to the solution – the caregivers contributing the finished scripts and the client contributing the existing internal scripts (see section 7.2). Wicked problems are not statically solved; rather the ongoing solution often is described as a process. An example of this is the plotting of a bus route through a residential neighborhood, where the trade-offs include local passengers, property owners, traffic managers, and urban planners. The route as planned may be good for only several years or less.



Many “real-world” problems can only be resolved with an approach that utilizes the symmetry of ignorance or asymmetry of knowledge [Jonathan Ostwald et al., 2003] of the stakeholders to design a satisficing solution. A satisficing solution [Simon, 1982] refers to a solution that is “good enough” for all the stakeholders to achieve some minimum of desired functionality. This is a useful notion, because any attempt to reach an optimal solution becomes very difficult due to limits on resources and time.

### 3.5. Metadesign

The design of the MAPS system, especially the MAPS script-design-environment, is centered about metadesign [Giaccardi & Fischer, 2005]. One definition of metadesign is design for designers. Metadesign then is the set of techniques and an approach to making artifacts that in turn can be used to design and make further artifacts. Analogously, on a physical level the metadesign process is like the tool and die maker who uses the lathe, the turret drill, and metal formers of various sorts to create tools that in turn can produce finished goods. In the social realm, *Robert's Rules of Order* [Robert III et al., 2000] is a metadesign object that supports the fashioning and operation of legislative bodies whose primary purpose is to enact laws. Another perspective is:

Besides designing artifacts, designers also often design the tools they use to do design. This additional level of design is metadesign. System building techniques that support metadesign include “end-user modification” and “end-user programming” [Jonathan Ostwald et al., 2003]

Metadesign is used by various *communities of practice* [Lave, 1991] to support a range of approaches. Artists that use a metadesign approach to support collaborative work use it to denote not just the *boundary object* [Nardi, 1996b] that supports the creation of works of art but also to discuss the specific process that is undertaken in using the metadesign system [Giaccardi, 2004]. Computer scientists use a more restricted definition of metadesign. In this dissertation, metadesign refers to the design of the metadesign object and the design approach (e.g. underbuilding, low entry cost, and high ceiling) that is required to create metadesign objects.

### 3.6. Bridging from Framework to Design

The particular problems of this unusual domain and its HCI implications are classically wicked in that it is impossible to specify the solution separate from *doing* the solution [Rittel & Webber, 1984]. As has been stated earlier, a sad truth about complex high-functioning AT (in contrast to mechanical or single-function aids) is the very high level of abandonment (see section 2.3) of this technology.

MAPS design is grounded in the aforementioned HCI perspectives and adapted their methods and perspectives to this population. Being aided by advice from experts in AT use for persons with cognitive disabilities as well as personal experience gained by preliminary usability testing with users and caregivers resulted in an effective and useful cognitive orthotic. Designers of high-functioning AT need to be cognizant of the unique two-user requirements for these devices. All of the theoretical approaches assume a single user and the task at hand, whereas in this domain, the needs and abilities of both the end user and the caregiver “partner” need to be considered [Kintsch, personal communication, 2002]. Simply relying on theoretical considerations in interface and functionality design, however, does not necessarily give any advantage. The best approach is to become aware of the special needs and abilities of the stakeholders in the AT realm as a group (not just as individuals) and bring the powerful perspectives of the HCI community together on the problems of assisted technology design.

**Table 2: Comparison of Frameworks**

	Unit of analysis	Components	Model
<b>Distributed Cognition</b>	Cognitive system composed of individuals and the artifacts	Actor(s), Artifact	Centering on cognitive act in the context of attaining a goal
<b>Symmetry of Ignorance</b>	The whole of knowledge about a domain	Symmetrically knowledgeable users	Centering on coordination of knowledge holders
<b>Metadesign</b>	Design over time	Metadesign object, users	Centering on co-evolution of metadesign object

In discussing the design process for MAPS (see section 5.2), the basic characteristics of each approach are enumerated as well as the appropriate modifications to the “classic” view of each that

must be done to make them applicable (see chapter 7 ). For each perspective, there is a discussion of the unit of analysis, the elements it uses to deconstruct the problem, and the model that it uses to describe and predict behavior. Table 2 highlights these different perspectives.

## 4. Related Work and Systems

### 4.1. Overview

Beyond the traditional mnemonic use of prompting by rehabilitation professionals [Saskatchewan Learning - Special Education Unit, 2003; The Oregon Research Institute, 1988] and studies by academics [Aist, 1973; Reed, 1989] have been several examples and studies of computationally enhanced prompting systems in the academic [Lancioni et al., 1999; Lancioni et al., 2000; Lynch, 1995] and commercial world [AbleLink, 2003; Davies et al., 2005; Keating, 2006; Levinson, 2003]. Also, several researchers and commercial developers use PDAs to support tasks and ADLs without multimedia prompting [Levinson, 2003]. In addition, broader guides exist for designing and evaluating assistive technology, such as “Assistive Technology, Essential Human Factors”, by King [King, 1999], “Living in the State of Stuck” by Scherer [Scherer, 1996], “Human Computer Interfaces for People with Disabilities” by Newell and Gregor [Newell & Gregor, 1997], and “Augmentative and Alternative Communication” by Beukelman and Mirenda [Beukelman & Mirenda, 1998]. Additional research resources exist in the journals of RESNA [RESNA, 2004] and the proceedings of the SIGCAPH Assets conference [SIGCAPH].

### 4.2. Stationary Multimedia Prompting Systems

During the last decade, some research has been done into the design and efficacy of using computationally enhanced prompting systems. Lancioni and colleagues performed several studies evaluating a stationary prompting system to aid persons with cognitive disabilities to perform tasks [Lancioni et al., 1999; Lancioni et al., 2000]. The authors did not specify IQ ranges but they did report for two studies, using a total of nine subjects, the *Vineland Adaptive Behavior* [Sparrow, 1985] scale scores for daily living skills of between 2 to 6.5 years and 1 to 2.5 years in socialization. Lancioni et al. used a stationary PC and speaker inside a case with the interesting addition of a vibrating box capable of being put on a subject’s belt, to provide haptic input. The experiments compared a task being done

with and without external prompting and utilized a primitive timeout mechanism to repeat prompts in the case of a lack of interaction with the prompter beyond a predetermined time. They obtained good preliminary results comparing prompting with no prompting for their population.

*Visions* [Baesman & Baesman, 2000] is a commercial stationary multimedia scheduling and prompting system that has enabled cognitively disabled individuals to move from group or family homes into independent living environments. *Visions* uses a PC with touch screens and in-ceiling speakers to prompt users through ADLs such as grooming, cooking meals, or morning routine (medication, hygiene, etc.). *Visions* also supports out-of-house activities such as shopping through a card reminder system. The combination of scheduler and presenting menus of possible activities has been remarkably successful in aiding independence for persons with cognitive impairments.

Impeding the widespread adoption of *Visions* were several problems, some socio-technical in nature, some more technical. In one installation of *Visions* in a cluster of apartments, many of the staff felt that installing *Visions* was motivated by an attempt to replace staff [B J Sample, personal communication, 2000]. Introducing assistive technology into a stable family configuration that has accommodated the member with cognitive disabilities is sometimes seen by the caregiver as possibly disrupting the psychological structure of the family [Kintsch, personal communication, 2002]. This perspective, although not core to the present research, must be kept in mind in considering the issues involved with device abandonment.

Two structural, technical problems are encountered with the *Visions* approach: (1) the system is physically tied to the house and (2) it is difficult to reconfigure. Because it is tied to the house, a significant part of the behaviors that constitute independence, such as shopping or taking a bus, are not directly supported. When *Visions* is initially installed, the scripts in the system may be sufficient to support independent living; however as abilities, needs, desires, and the environment change, the system must be reconfigured to continue to support user independence. Inserting or changing scripts is a nontrivial task that requires a technician [B J Sample, personal communication, 2000]. Often the cost of this was not considered in calculating the cost of the whole system over time, and reconfiguration can be put off or neglected entirely, resulting in frustration, misfit, and often abandonment.

### 4.3. Multimedia Handheld Prompting

In commercial offerings, AbleLink [Davies et al., 2002; Davies, 2004] produces, among other devices for persons with cognitive disabilities, a handheld prompting system called Pocket Compass (see Figure 10). This device is based on the research of the AbleLink principals [Davies et al., 2002; Davies & Stock, 1996], and solves the mobility problem by moving the platform from a stationary PC and touchscreen to a PDA. The making of new scripts is supported with the Task Builder graphical task-building application for creating prompting content; the resulting script conforms to an “instructional media standard,” which was designed and has been promoted by AbleLink as the AIMS (AbleLink Instructional Media Standard) standard [Davies et al., 2005]. AIMS-compliant scripts can be played on any AIMS-compliant prompter. Scripts built with Task Builder run on the two handheld prompting offerings of AbleLink, Pocket Coach and Pocket Compass. Pocket Coach provides only verbal prompts, whereas Pocket Compass supports image and verbal prompts (as well as video prompts).

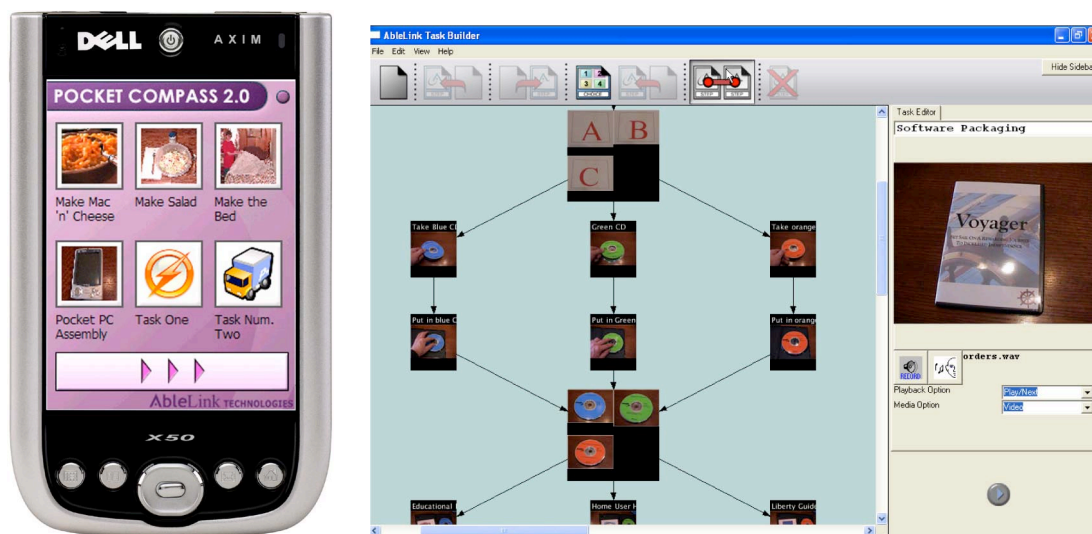


Figure 10 - AbleLink's Pocket Compass and Task Builder

Task Builder allows users, or *content developers* as they refer to MAPS's caregivers, to create scripts. A new script or task (as the Task Builder documentation calls them) is created by first using the file menu/new task option, then selecting a directory in which to place the new script, and naming

it. Having named a new task/script, the user can next add steps by clicking on one of several options displayed on the top of the design space below the menus. At any point in the script a new step or a “decision point” can be added. Decision points are special steps that allow the person with cognitive disabilities (or as the Task Builder manual refers to them, *content users*) to select from up to four possible optional branches, thus supporting forking, which is quite essential to scripts supporting employment. At each step, the caregiver can add an image by clicking on the step and choosing from the MicroSoft operating system (MS OS) (standard) file selection box, which initially presents the “working directory for images” (the last used directory for images) in list form. The user can change this list to a ‘thumbnail’ perspective if desired. Selecting the desired image causes the image to be displayed in the step box representing the task step in the left-hand side (the *task work area*) of the application. Similarly, a verbal prompt can be added from pre-recorded prompts in the file system. If desired, the user can record the verbal prompt on the fly by using the application’s built-in sound recorder, which the manual says can be previewed using the built-in wav file player. Task steps can also have a timer associated with them and the timer can, when triggered, cause a sound file to be played, the next task step to be displayed, or replay the current step of the task. At any step the user can insert a replay option, causing the script to restart at the first step. As the task is built up, step by step, the steps populate a directed graph representation in the left-hand task work area. The application shows five steps at a time; viewing steps of scripts larger than five in length is accomplished by use of a standard slider control.

Task Builder (AIMS) tasks/scripts are stored as a single eXtensible Markup Language (XML) file representing the task structure and individual image and wav files. When an AIMS task is distributed to an application, these files (XML, image, wav) are copied into the file system of the application to run the script. The AIMS Task Builder provides flexible support for forking via the decision point steps, and looping is supported by either the restart step option or a restart button built into one of the decision point choices.

AbleLink’s system is similar to that of MAPS: both support caregivers building scripts for task support and storing them for play on handheld Windows Compact Edition (WIN-CE) platforms. Architecturally, the two differ in that MAPS stores its scripts in a commercial database and provides

the ability to switch between a local database and a database hosted on an Internet server; the AbleLink offering stores scripts using the native OS file system. The primary advantage to the database is that all operations are atomic (i.e., a user can't "lose" or corrupt part of the script); the application either gets the whole script or it returns an error message. Likewise, switching the MAPS script-design-environment to use the Internet based database server is trivial and does not involve any OS-level commands or actions. Using the file system to store the scripts removes one layer of middleware, however.

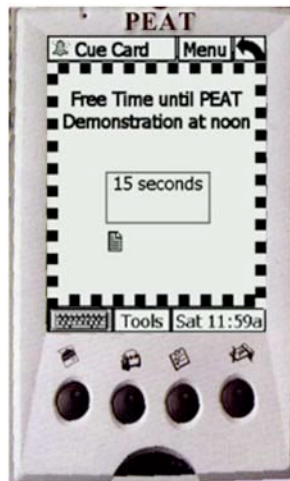
Graphically, both provide graphical user interfaces (GUIs) and menus to create scripts and both use the native OS file system for storing and retrieving images and sounds. The AbleLink system has support for using videos in place of sound wav files. MAPS provides multiple views of the script in the process of its being created (the filmstrip, the three or four wide whole script preview, the single prompt preview) as well as a preview of the finished script. AbleLink provides a user's manual and MAPS provides a page of text-based help shortcuts and video help in the form of 19 short instructional videos. MAPS and Task Builder differ in how they support forking and looping: AbleLink provides a graphical element, and MAPS expects the caregiver to create looping and forking as an intersection between prompt content and multiscript affordance. The scripts that Task Builder supports can have a timer associated with a given prompt, which can cause forking, prompt replay, or other actions – a useful function that MAPS does not have.

The Planning and Execution Assistant and Training (PEAT) system (see Figure 11) is similar to the AbleLink products, but focuses on triggering behaviors by aiding scheduling. PEAT combines *“artificial intelligence with graphical displays, touchscreen controls, personalized verbal and audio cues, and other features designed to compensate for cognitive, perceptual, and motor deficits”* [Levinson, 2003]. Users can also use the system to cue series of behavior, approximating MAPS script task support.

Similar to PEAT, Picture Planner [Keating, 2006] is an icon-based personal planning system that supports persons with cognitive disabilities in scheduling single and repeating events by multimedia prompting. Synthesized voice prompts provide push functionality. Picture planner is PC (Windows or



Macintosh)-based, and, although not mobile, has an interface that is extremely easy to use, producing good results on first tries.

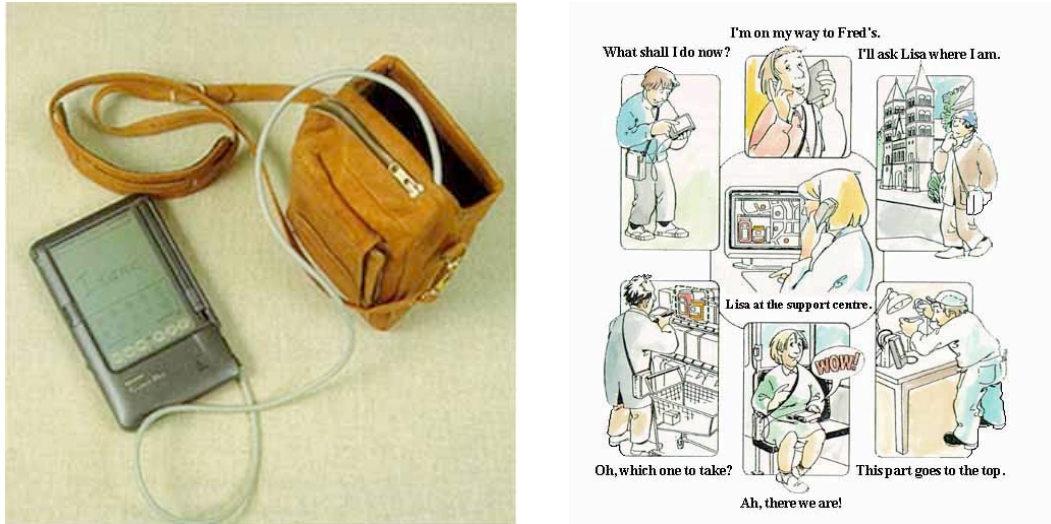


**Figure 11 - PEAT Memory Aid**

#### **4.3.1. The Isaac Project**

The Isaac project [Isaac, 1998], a research initiative from Sweden in the mid-1990s (see Figure 12), was a visionary exploration of possibilities inherent in PDAs for persons with cognitive impairments. It foresaw PDAs having global positioning systems (GPSs), cell phones, and digital cameras providing support in an integrated environment. Isaac posited a central caregiver to support multiple Isaac device users that would provide support in the case of the inevitable breakdown situation.

Several research papers provide insights into their vision of support for independence for people with cognitive disabilities. They addressed issues spanning knowledge representation for this population to use of scenarios [Svensk, 2001] in the design process. One of the key insights that inspired the MAPS design was viewing the user as part of the design team and not a passive object [Jönsson & Svensk, 1995].



**Figure 12 - Isaac**

Unfortunately, the Isaac project ended before any real use of the one prototype was evaluated. What proponents of the Isaac project did leave was a bold vision of empowering persons with cognitive disabilities with mobile computational devices. They also had a vision of seamless integration and appropriate levels of representation. They correctly did not attempt to replace human caregivers with artificial intelligence, not only because it was too difficult, but also because the added dimension of caregiver involvement was more than just solving the problem at hand. Unfortunately, the technical infrastructure was simply not available to create a truly mobile and robust device in 1994; the Isaac prototype shown in the left side of Figure 12 consisted of a “hacked” Apple Newton hand computer connected to a shoulder bag containing a hand-built state-of-the-art (as of 1993) PC with a cell phone, video camera, and GPS receiver attached.

## 5. The Memory Aiding Prompting System (MAPS) Socio-Technical Environment

### 5.1. Scenarios

The scenarios that follow provide a context for the discussion of the HCI approaches presented in the next chapter. As part of the design process, the CLever group has extensively used scenarios to clarify and expound on design possibilities [Svensk, 2001]. Scenarios can expose assumptions as well as allow others without software design experience to participate in the design process, thus exploiting the *symmetry of ignorance*. Some functionalities of the MAPS system described are not yet implemented but are explicitly spelled out in these scenarios; they are presented here to provide a glimpse of the breadth of vision of the CLever project. Scenarios can be used to acquaint a stakeholder—or anyone, for that matter—with a vision of how a socio-technical environment may work, for recruiting possible study subjects, or to engage in academic discussions. These scenarios must be rewritten when the system is well into the fabrication phase to avoid confusion when interacting with the prototype or finished system.

An additional reason for placing these scenarios here is to provide a good context for understanding the process of developing and using the system, each scenario being an idealized version of the process that the four participants of the realistic field trials experienced. This provides an understanding of the nature of the socio-technical environment that MAPS supports, as well as the details that, unless experienced, would be difficult to imagine. The roles of persons with cognitive disabilities and caregivers are typically removed from most people's daily experiences. The scenarios that follow were not the original scenarios used at the very early prototype design phase. They are a combination of design scenarios and articulation of actual trial experiences with the system

(appropriately made anonymous) that give a better feeling of the setting, roles, plans, actions, events, and artifacts involved [Rosson & Carrol, 2002].

### **5.1.1. Scenarios as Design Tools**

Designing assistive technology by scenarios allows one to “try out” possible design solutions to wicked problems without a large commitment of effort [Rosson & Carrol, 2002]. By using scenarios and having special education or assistive technology professionals participate as proxies for the persons with cognitive disabilities [Boyd-Graber et al., 2006], all the stakeholders can participate in the design process. This can solve the problem of doing participatory design with members of a population that cannot express themselves with the necessary level of complexity. Due to the low-cost tryout of possible solutions and evaluation by all the stakeholders, design by scenarios exposes assumptions. Scenarios can force implicit information and skills to the forefront, moving the tacit to the explicit.

Scenario design may be a two-part process: (1) the creation of problem scenarios and (2) their resolution with design scenarios. The problem scenarios make explicit a user’s world that may be highly different from the designer’s world. The creation of problem scenarios can be driven by an ethnographic approach to studying the end-user populations. This process pulls the tacit expert knowledge into the forefront, enabling the designer to accommodate and support the whole task and each stakeholder’s part. It enables (or forces) the description of roles that can then be corrected and relied upon for the design, and it supports claims analysis by stating the benefit of this trade-off.

A useful way to start generating scenarios is to describe in some detail several hypothetical stakeholders and use them as actors in the scenarios being developed. Scenario writing is interleaved with claims analysis in that features of a situation that have important effects for the actors are identified and written down, including both good and bad effects. A good guideline is to write a scenario for each kind of event and from the perspective of each of the roles in the process. With this method, some stakeholders may discover the tacit knowledge about which they have made unconscious assumptions, particularly with respect to already internalized skills or internal scripts (see section 7.2). Make explicit the claims that drive the scenarios, and carefully analyze whether the resultant system will support the claims. Explicitly stating a claim (by “using this script on a prompter,

people with cognitive disabilities with this script could do the laundry by themselves”) invites study of the processes that need to support it, as well as exposing missing supports. For example, in the case of doing the laundry, what was discovered to be missing was a low cognitive-level support for where the dials needed to be set. The pictures on the handheld were not clear enough regarding setting the water temperature and other controls, so pieces of colored tape were introduced, changing the task from “set the water temperature dial to warm” to “match up the white tapes and then match up the yellow tapes” (see the discussion of this scenario and Figure 34 in Chapter 6).

Scenarios allow the use of metaphors at a pre-prototype stage. Metaphorical examples in MAPS include: using the script-design-environment is like assembling a filmstrip, or like making a PowerPoint set of slides. Within the script-design-environment, such examples would be: selecting images and sounds is like finding a file in the Windows operating system (OS). A scenario should convey what its actors are like and what forces are influencing their behavior [Rosson & Carrol, 2002]. One of the metaphors in the early stages of MAPS design was that the prompter was like a “talking book.” At that point in the implementation of MAPS there was a rudimentary script-design-environment and the prompter software ran on a palmtop machine.

Consider this scenario:

For our son’s fifth birthday, we wanted a treasure hunt in which all the small boys went in a group from clue to clue and eventually were led to the “treasure” and could share it as all good pirates did. A typical five year old does not read, and the clues were a bit complex, so we put the clues into the prompter. The gang of kids successfully found the treasure by listening to the prompts and matching up pictures with places in the house and yard. The talking book metaphor worked well for this population.

Scenarios allow a designer to explicitly use metaphors and discover whether the details of the user interaction are easy to understand and congruent with the correct behaviors in the user experience. One powerful thing about choosing the correct metaphor is that users can be guided in their interaction, but the “real” process remains hidden. One metaphor is that the script-design-environment is like a filmstrip, yet the design of the supporting processes that constituted the script-design-environment was

not modeled on this at all. During usability testing, the users were asked to describe “how” the program worked, and the successful understanding of the metaphor may have had little to do with the literal actions of the script-design-environment. Being successful means that if the user makes another action based on the metaphor, that action will have the correct result [Preece et al., 1994].

### **5.1.2. Overview Scenario of MAPS Task Support**

Doris, who recently received a certificate of completion upon finishing high school, has a cognitive disability that has prevented her from living independently. She has the basic skills to move into her own apartment with a roommate and with visiting caregiver help. She knows the basics of how to cook and clean her home; she just gets befuddled about the details and their timing. She could get around these problems by using a handheld computer that provided step-by-step prompting of doing the specific parts of a household task. Doris’s caseworker, Steve, has created a number of scripts to help her over the sticky points. For each task, he wrote out a list of the steps to do the task; then he wrote the specific words he needed to record to guide Doris through the task (e.g., “get the large frying pan out of the cabinet”) and made a list of the pictures he would need (e.g., a shot of the large frying pan on Doris’s kitchen counter, next to her stove). He then recorded the verbal prompts with his computer and took the pictures with his digital camera and uploaded them to his computer. Next, he put together the script using his MAPS script-design-environment, placing the images and recordings into a series of prompts, and saved the sequence as a script. He downloaded the script to Doris’s handheld MAPS-prompter, which she used later that day to successfully cook the dinner—a task that had, in the past, had left Doris angry and frustrated with the difficulty of doing all the steps at the right times. With the support of the custom script on her handheld prompter, Doris was able to complete the task correctly and in good spirits.

### **5.1.3. Script Creation Scenarios**

Steve had learned about MAPS from the special education assistive technologist in the school where Doris was enrolled in the transition program until last year, when she turned twenty-one. Doris had recently moved into a group home where Steve was a caseworker. Steve was one of several caseworkers and job counselors who were on staff rotation in and out of the house, which had three

two-bedroom apartments and a small office for the staff. Steve got the MAPS system and installed the script-design-environment on his PC. The PDA came with the MAPS prompter software preloaded. Steve read the materials included and did the exercises to help him learn how to use the script-design-environment. Along with the MAPS package came a workbook and DVD [The Oregon Research Institute, 1988] that taught him the principles of task segmentation and script creation. Even though he had a fair amount of experience doing this when he taught his clients new tasks, the formal and principled approach of the workbook and DVD helped him to think about scripts from the MAPS perspective.<sup>6</sup> Steve built up the training script from the included prompt pictures and wav files of verbal prompts. After he had finished the MAPS training, Steve and Doris got out the prompter preloaded with the demo script, and he showed Doris how to work the controls, going forward through the script, replaying the verbal prompts, and going backward. Steve decided to look at the more complex multitask demo script after Doris successfully used the prompter for several scripts.

#### 5.1.3.1. Script creation and editing – Linear

Steve decided that a good first script for Doris would be cleaning up the laundry room, as all the residents shared cleaning and cooking chores. Doris never could quite get all the details done right—either she would do all the parts badly or do most of the steps but skip some of them. Steve started by mentally doing the task and writing down all the parts in a list. He then used the list and went into the laundry room and, using the list, performed the task. He discovered that he had missed several key steps, such as where to get the pail for mopping the floor, which he added back to the list. He then typed the list into the template text file that came with the MAPS package (see appendix 13.6) and made notes on it as to what would be the best picture and the right verbal prompt for each script. He called his friend Bjork, who had originally told him about the MAPS system, and showed her his script. Bjork pointed out several places where Steve had made assumptions about doing the task where following only the script might result in a breakdown, and Steve amended his list of steps to include the missing ones. Bjork also pointed out two subtasks that Doris would be likely to learn fairly quickly

---

<sup>6</sup> Whereas traditional prompting sees scripts as a technique to teach the whole task, MAPS sees them from a distributed cognition perspective, as external supports that transform the task.

and suggested that Steve put “collapse point” steps (see section 6.6.2) into the script so that it would be easy to collapse later when Doris did not need the detail of the initial script. Bjork also suggested that Steve personalize the verbal prompts by using Doris’s name in several steps and giving her a couple off “attagirl” prompts in the middle and at the end, adding a congratulatory finish prompt.

With the completed and tested list in hand, Steve set his digital camera at the lowest resolution and took all the necessary pictures, being careful—as the MAPS guide suggested—to shoot all of them in portrait mode (with the long way vertical), as it would appear on the MAPS-prompter. When he finished, he downloaded them to his PC, onto a folder labeled “Doris laundry room cleanup pictures” and changed the file names from the sequential numbering that the camera used to more meaningful file names such as “get the mop.” Having done that, Steve got out his microphone, plugged it into his PC, and by using the “sound recorder” application that was one of accessories provided with Windows XP, Steve recorded each of the verbal prompts from his list. As each was recorded, he saved them as “telephone quality” files (to make them smaller, but quite audible through the MAPS-prompter tiny speaker) and gave them as a file name the text that he had read into the microphone. He put all these prompt files into a folder on the PC labeled “Doris laundry room cleanup prompts.”

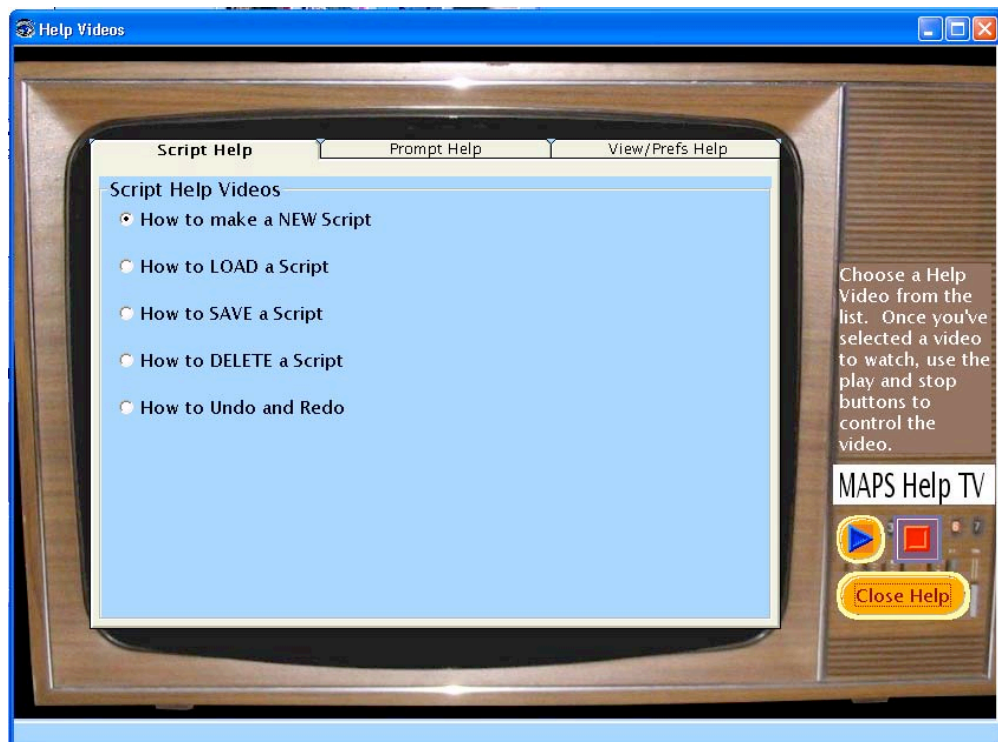


Figure 13 - MAPS Video Help



Steve now had all the parts he needed to assemble his first script. He started up the MAPS script-design-environment and set the picture and verbal prompt displays to the two directories to which he had saved the pictures and wav files. He had a bit of a problem at first in starting the process, but he went to the video help application, selected the “script help” tab, and checked the “how to make a NEW script” option, and then he pushed the play button (see Figure 13). After watching the short movie, he went back to the main application and followed the steps that the video told him to, and soon he had a new empty script ready to put together. With his script template form at hand, he selected “new script” from the file menu, and one-by-one he put the images and verbal prompts into a sequence of prompt steps, filling up the 25 steps of the laundry room cleaning script.

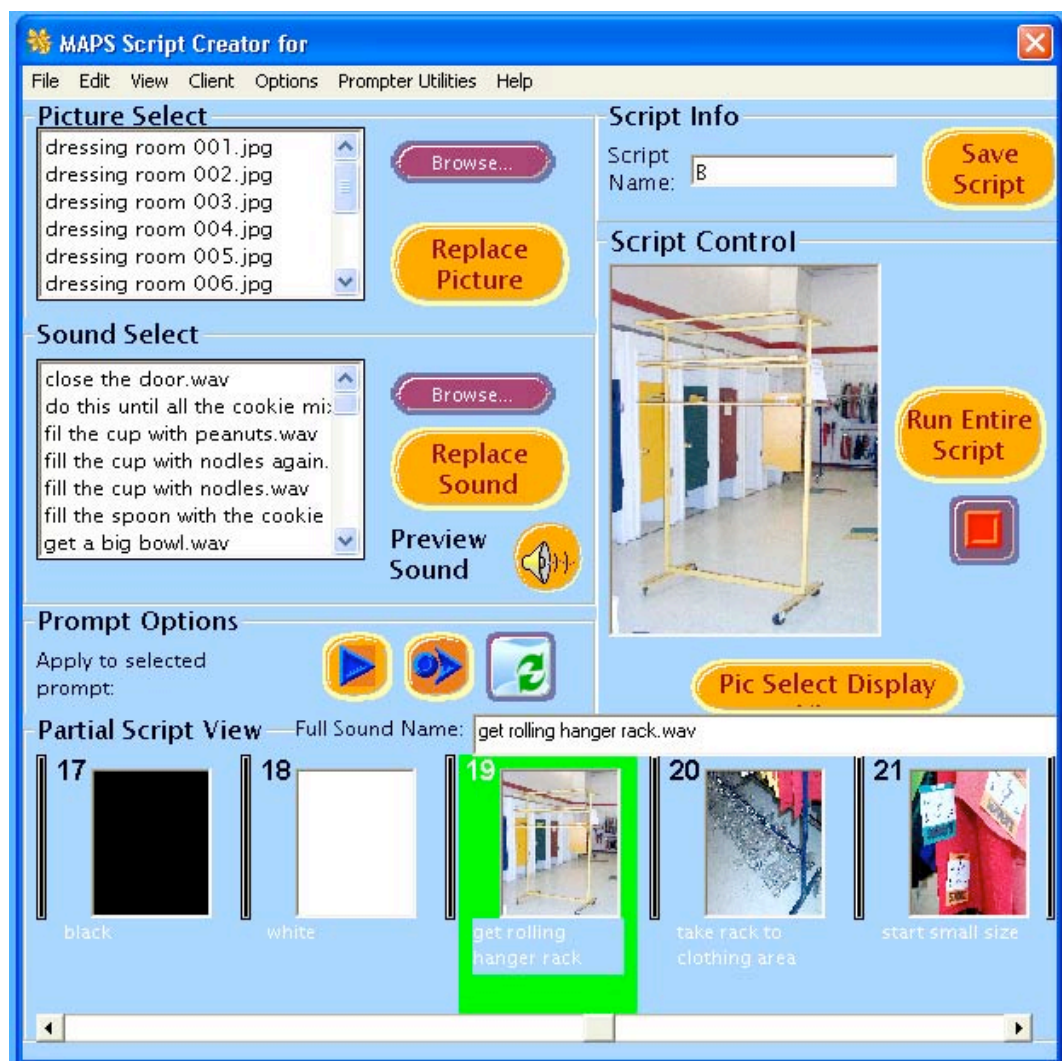
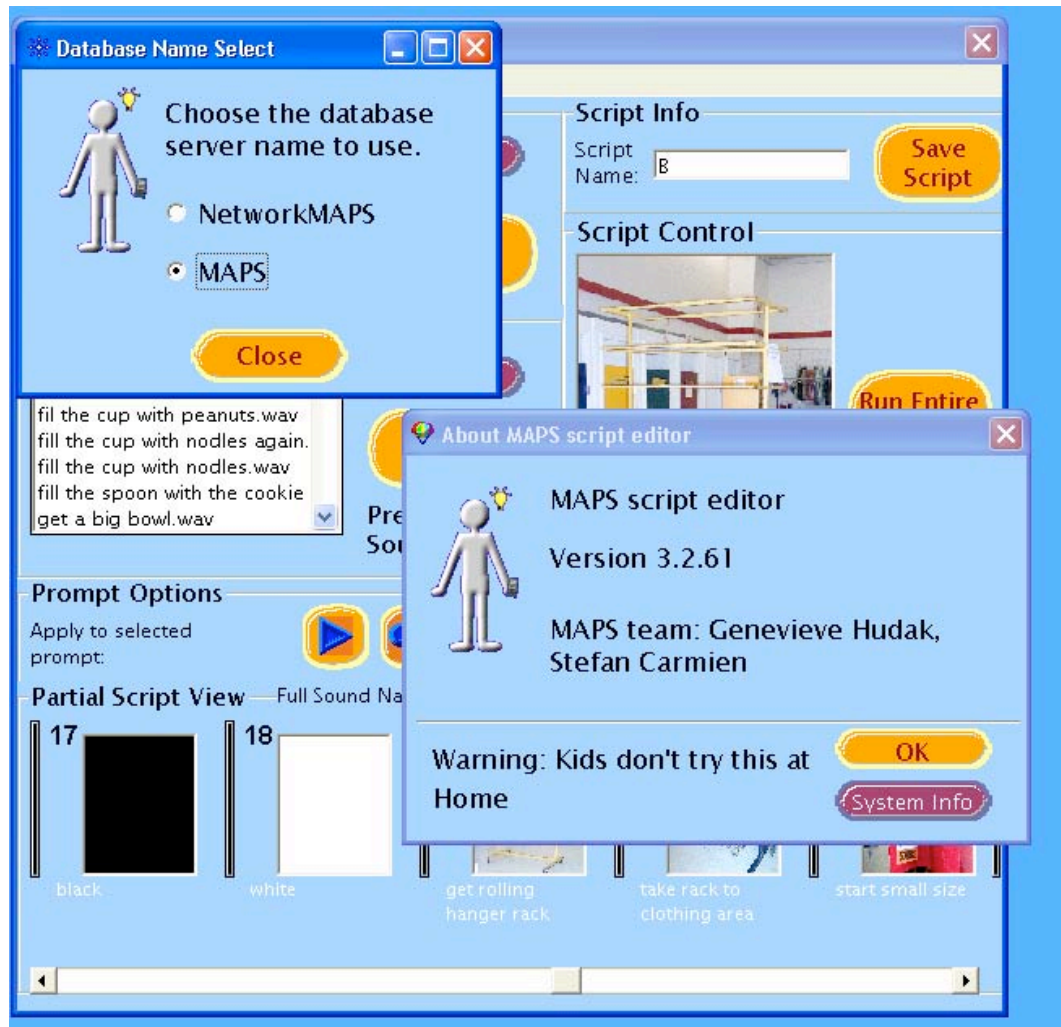


Figure 14 - MAPS Script-Design-Environment Preview Pane

Steve then previewed the script in the script-design-environments preview pane (as shown for another task for Doris in Figure 14), and since he was satisfied with the results, used the options menu and selected the “prepare the script to load to the prompter” option. He then put the prompter into the synch cradle and selected the “Easter egg” (see Figure 23) and then selected the “add a script” option, and the MAPS-scripter application synched with the database and downloaded the script. Steve then previewed the script on the prompter, decided that one of the images was not clear and one of his verbal prompts was too soft in volume. Steve took a new picture and re-recorded the prompt and then put them into the correct folders. He then opened up the MAPS script-design-environment and was stumped at first about how to edit the existing script, but a quick visit to “video help” got him going again. He loaded the script, replaced the picture and voice recording, and saved the corrected script. He then re-synched the script with the prompter and, satisfied with the script, was ready to help Doris.

#### 5.1.3.2. Script creation and editing – Design by modification

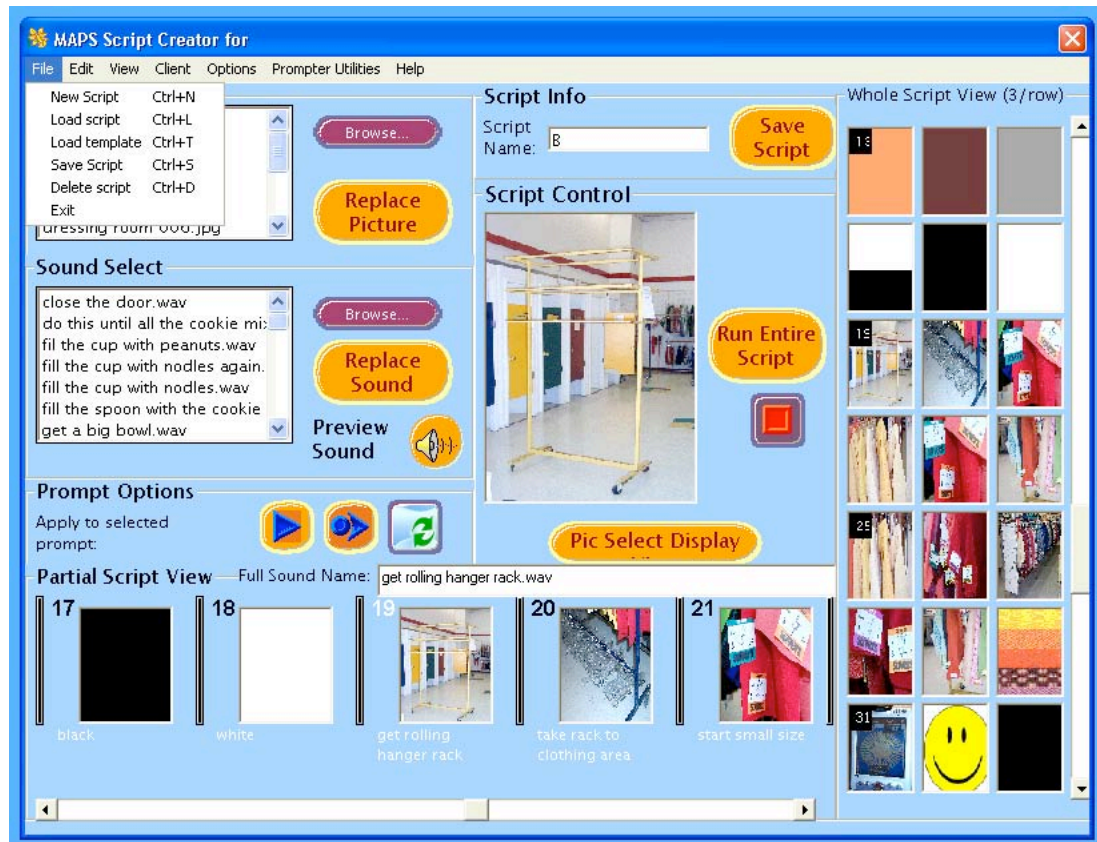
Steve decided that the second script he would make for Doris would be one that would be fun for her rather than a chore. After talking it over with Bjork, he decided on a trip to the library because he had taken Doris to the main library a month ago and she really enjoyed getting her own library card and checking out a DVD about Italy. (Doris’s maternal grandparents were Italian, and she was interested in all things Italian.) Steve thought this outing would be all the more attractive because there was a branch library not too many blocks away, so Doris would use the MAPS-prompter not just to use the library, but also to make the short trip by walking to the library by herself. The next day, he broached the subject with Doris, and she was excited to try it. Bjork had mentioned that for Doris, who could be quite truculent when told to do something she was not interested in doing, it might be an interesting experiment to record the prompts using her own voice (see section 6.8). So when Steve talked about the script idea with Doris, he suggested the idea of her recording the prompts, which Doris was quite keen to do.



**Figure 15 - Switch Database**

When the time came to fill out the template with the library task, Steve was at somewhat of a loss about where to start. Then he remembered that the MAPS system had a repository of scripts that could be used as models for many different tasks. He went to the “video help” to find out how to use the templates. By following the instructions, Steve switched to the networked MAPS database (see Figure 15) and used the file menu “load template” option (Figure 16 shows this option for a different scenario) to look through the 170 template scripts. All scripts were made anonymous by replacing any pictures with a generic stub and with only the words for each prompt step as file names for the wavs, with a stub for the actual sound. To his relief, Steve found several scripts about the library. He looked at them and chose one as being appropriate for Doris. He then loaded it up into the script-design-

environment, switched back to the local database (see Figure 15), and saved the loaded script as “Doris’s\_library.” He used the steps as a framework to build up his task segment for the library part of the script. For the walking part, he took his camera and a clipboard with the script design template and walked to the library, taking many pictures along the way.



**Figure 16 - Load Template Option**

With the pictures and the script prompts written, Steve and Doris together decided which prompts and pictures were the best; for some parts that were particularly unfamiliar to Doris, however, Steve wrote in the steps without her participation. When the form (see the example in appendix F 13.6) was filled out, Steve helped Doris to record the prompts in her own voice. Once the pictures and wav files were in appropriate folders, he proceeded to finish the task of assembling and testing the script.

### 5.1.3.3. Script creation and editing – Multiscripts

Doris had done well with the previous two scripts, so Steve decided to use the system to help Doris at her place of employment. Doris had just started her new job at Second Time Around, a clothes recycler. Doris's main job was to keep the racks of clothes in order and to generally clean up the aisles. Steve had been doing job coaching for many years, and he knew that one of the big impediments to permanent employment for persons with cognitive disabilities such as Doris were not so much the detail of the task, which could eventually be mastered after many repetitions, but the "soft skills." These "soft skills" were problematic in general for this population, and especially for people with autism or Asperger's syndrome, who often just stopped when the current task was completed, waiting for someone to tell them what to do next. Steve had been through several experiences in which the store manager was satisfied with the client's performance but had such significant problems with the client's personality and efficiency (due to the tendency to just sit and wait for the next task) that the manager was forced to fire them.

For this position, Doris should learn two main tasks, and it would be good to have another task so that when she ran out of other things to do she could still keep busy. If Steve and Doris could lick this problem, he was sure she could transition from sheltered employment to a full paid position. Steve called Bjork again and told her his ideas and problem; Bjork told him about MAPS multiscript mode and that he could fashion scripts that would support repetitive tasks and choices.

Steve went back to the MAPS help section to learn about how to do this, and he was pointed to the second section of the task segmentation workbook.<sup>7</sup> From the workbook he learned that he could load up to three scripts in the prompter in multiscript mode and they would be accessible on the prompter by clicking one small icon in the bottom middle, top right and top left (see Figure 24). The icons would be made from the first prompt image of each script, and clicking one of them would cause the scripter to start that script. The trick to restarting the task when you were finished, according to the

---

<sup>7</sup>Help for all the features used in these scenarios has been implemented using video help movies. However, the help support for the multiscripting use of MAPS was provided by the developer in face-to-face meetings with the caregiver (also known as "wizard of Oz" prototyping).

manual, was to put a prompt near the end of the script that instructed the user to click on the icon of the script currently running, and switching to the next task was done by placing a prompt, after the restarting prompt gave the user the choice of either going to another task (clicking on one of the other icons) or continuing to the end of the scripts (proceeding to the next prompt on this script). Steve had to read the material several times and do a bit of doodling on paper to really understand the trick, but he got it eventually. All that was left was for him to fill out a script design template (see appendix 13.6) for each of the scripts, take the pictures and record the prompts, assemble the scripts, and load them into the prompter. Once he did this, he spent part of the afternoon checking out the prompter to make sure it worked correctly and fitted the tasks. He also wanted to be able to effectively teach the new use of the prompter to Doris.

#### **5.1.4. Script usage scenarios**

This section describes how persons with cognitive disabilities and caregivers might use the MAPS-prompter. In this section, the use environments and tasks are anonymous versions of ones that were encountered during the field trials. Similarly, the members of the dyads are based on participants in the realistic experiments.

##### **5.1.4.1. Linear script usage**

**Cleaning the laundry room:** On the day that Doris was to use the prompter to guide her through cleaning the laundry room, Steve showed her the prompter and a demo script (the same one that came preloaded with the prompter) and had her play with it for a while to get acquainted with the simplified controls of the MAPS application. Earlier that week, he had downloaded some of his MP3 files and showed Doris how the prompter (an IPAQ that was running the MAPS prompter software) could be used as an MP3 player, and she had carried it around since then listening to her favorite tunes, except for when Steve had to assemble and load the new script. Satisfied that Doris had become competent at using the MAPS-prompter, Steve loaded the actual cleaning script and told her that it was the time they had agreed upon to try out the new device. Steve handed the prompter to her and sat in a corner of the room watching her use it. Doris had no problems with the controls, but sometimes she would start a subtask of the whole task and not wait for the next step, doing *that step* the old, wrong way. At those

points, Steve intervened and reminded her to just do the step prompted and refreshed her memory of how to replay the verbal script. Doris got to the end of the script with most of the steps done properly. They were not done completely right, but much better than she had done them before. Over the next few weeks, Steve worked with Doris and the cleaning script, and she got progressively better at following the steps precisely. After three sessions, Doris was doing the laundry room cleanup task correctly; so much so that Steve could collapse several steps into single prompts with the script-design-environment. With the new script, Steve felt that Doris had internalized the sequence and detail of the steps as well as she could for the foreseeable future, and with the aid of the MAPS-prompter and his script, she would be able to do this task with minimal supervision.

#### 5.1.4.2. Multiscript usage

**Employment:** Steve took with him the MAPS-prompter onto which he had loaded the three “Second Time Around” scripts in multiscript mode the next time that he was scheduled to work with Doris at her new job. Steve demonstrated to Doris how to start a task and how to switch to another script for a different task, as well as how to restart the task that Doris was doing. Doris started using the prompter for her first task, but she was going slowly because she needed to pick up the prompter, activate the next step, put the prompter down, and do that step. Steve had thought about this and brought along a special holder that Bjork had lent to him for just this situation. Doris slipped the prompter into the holder, put the strap (see Figure 17) over her shoulder and soon was going along just as fast as before the prompter, but with more assurance that she was doing it right. It was a bit tricky to *not* touch the screen in the three hot spots that changed the running script but she mastered it in a little while.



**Figure 17 - Prompter in a Palmtop Computer Holder**

With the prompter in the special holder, Doris could hear and see it as well as activate the controls through the flexible plastic face as she went through the racks, organizing them. Several times she went back to the first part of the script that showed her the correct order of the sorted clothes, and in about an hour she had finished the section of racks that her supervisor had set her up to do. Doris looked around for the supervisor but could not see her; she was told by a fellow worker that the supervisor had gone to a meeting and would be back in a half hour. Doris knew just what to do: she activated one of the other scripts. By pressing the icon with the picture of the shelves on top of the clothes rack, she soon was being guided through the process of arranging all the stock on the top shelf and pulling the ones that did not match into a box to be sorted out later. After Doris had been doing this for about 20 minutes, her supervisor came by and, pleased that she had initiated a task to fill the time between finishing the clothes sorting and the supervisor's return, gave Doris another section of racks for which to sort the clothes.

## **5.2. MAPS Design**

Each of the frameworks discussed above (see chapter 3) has unique insights to add to the design of assistive technology, especially for AT tailored for persons with cognitive disabilities. It is easy to see a prompting system as an example of distributed cognition. What computationally enhanced prompting does for a task is to remap executive function and memory load from internal requirements to the task of following the sequence of prompts.



Activity theory [Nardi, 1996b], although primarily a descriptive method, offers a helpful perspective on mediating objects in a social context, particularly the view that the relationship develops and evolves. The feedback loops, in the short term (as script logs and corrective action) and over a longer span (as collapsed or expanded scripts accommodating learning), express this perspective. A “wrong end of the telescope” insight that activity theory produces is the premise that MAPS script making is actually making operations into actions [Kuutti, 1996] (a sort of running development backwards) to fit the ability and needs of the caregiver and end user. Similar to the situated actions [Suchman, 1987] emphasis on the local emergent property of moment-by-moment interactions in working toward an overarching goal, activity theory’s insight is that development is driven by the contradictions that emerge in the activities processes. These exception states constitute not a problem but the basis for the ability to leverage existing cognitive ability toward independent living.

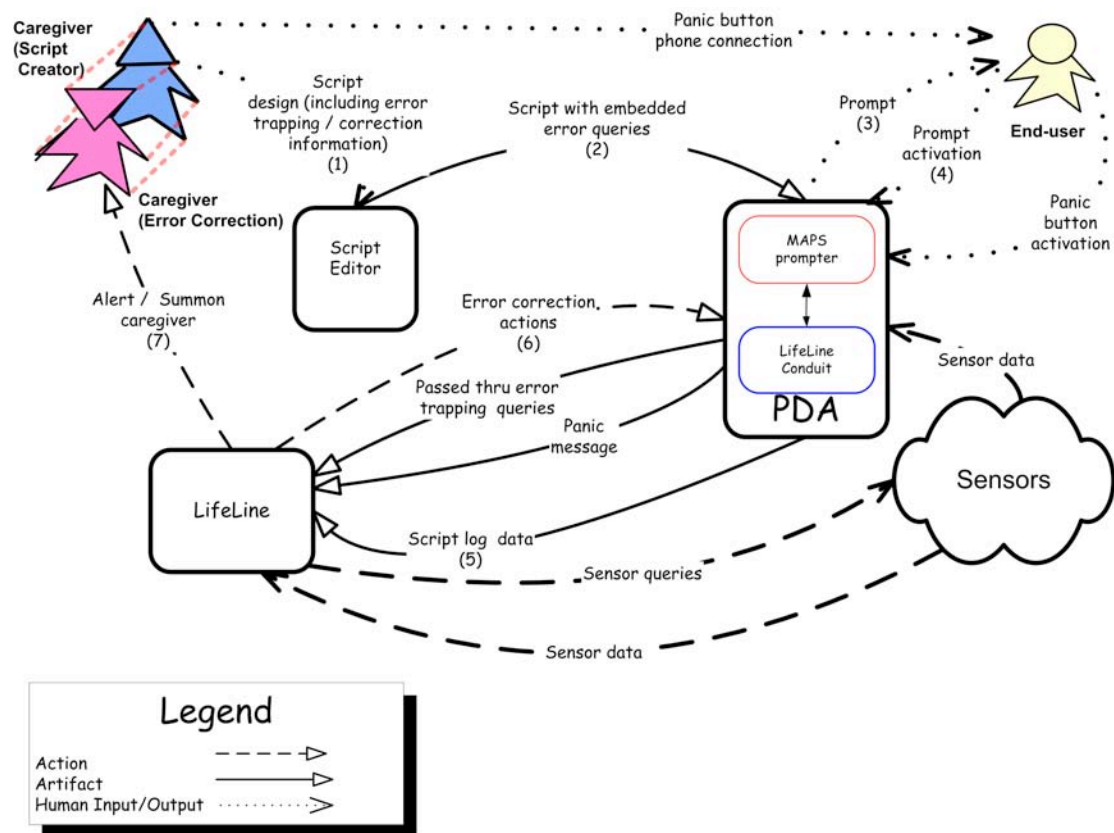


Figure 18 - MAPS and Lifeline Project as an Information Ecology

Nardi's model of information ecologies [Nardi & O'Day, 1999] and its basis in the notions of diversity and co-evolution map well to the notion of symmetry of ignorance [Fischer, 1999b] [Fischer et al., 2002], again in a unique way that has design implications for the MAPS/Lifeline system. An example of this is a software programmer (who knows everything about coding and making applications) and the owner of a hotel reservation company (who knows everything about hotels and reservations); by herself, the programmer would make a smoothly functioning, quickly performing, but *useless* reservation system; similarly, the owner would make a precisely fitted, exactly right hotel reservation system that would immediately crash (if it ever worked at all). It is the expertise that they have between them that allows the creation of a good-fitting, well-crafted application. In a parallel sense (see Figure 18), the end user provides the system information (prompt activations (3 in Figure 18 above), panic button activations, and the state trail of prompt activations) that the caregiver can use to satisfy the caregiver's need to feel that the end user is safe, and the caregiver can produce scripts that the end user will use in the system that allows the client to experience more independence and inclusion in his/her life.

### **5.2.1. Caregiver Script-Design-Environment**

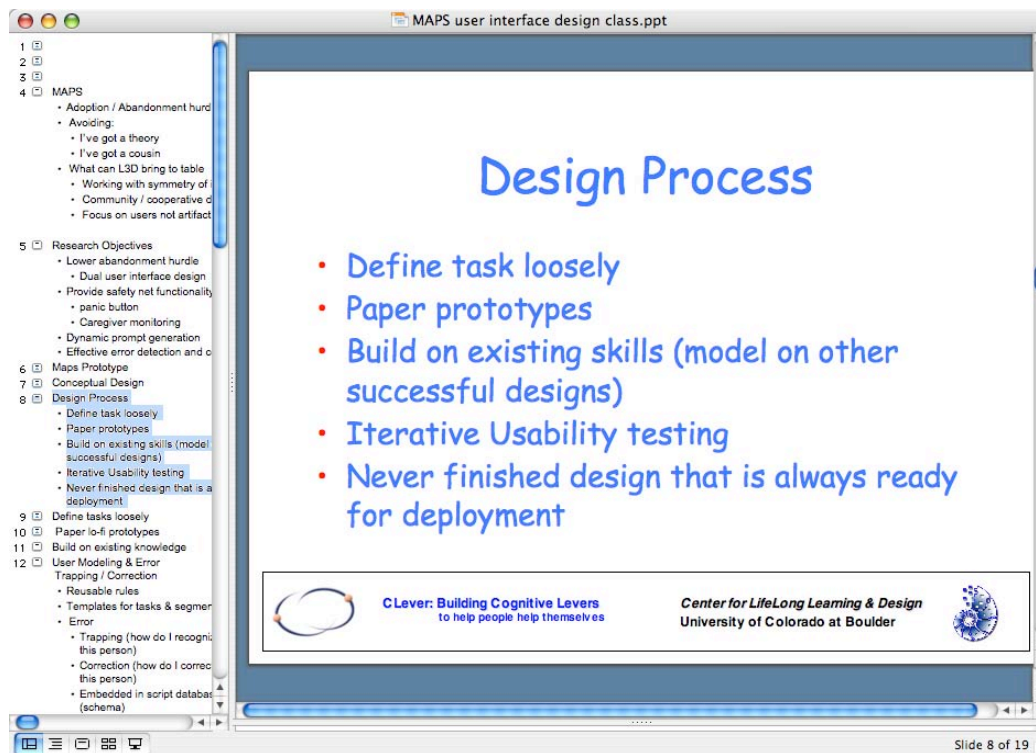
The MAPS script-design-environment (see Figure 19) to be used by the caregiver needs to be immediately usable with a shallow learning curve so that it can provide useful results as soon as possible. By ensuring minimum time spent learning to use the tool and as few as possible breakdowns presented to the user, the whole system will have a higher potential for non-abandonment. By involving the caregivers and AT professionals in the design process as soon as possible, there will be less backtracking in the process of making a good fit to skill levels and the task. The script-design-environment was designed by using tools from Lewis and Rieman's "Task-Centered User Interface Design" [Lewis & Rieman, 1993]. By starting from the task and the user and working inward, the design's focus was on usability and not the underlying technology. Also, by focusing in this way, the internal structure of the scripts schema was more amenable to extension.



**Figure 19 – MAPS Script-Design-Environment Functional Detail**

The MAPS script-design-environment was built on a user model that might be common to the caregivers' population. The design uses a filmstrip as a base metaphor, and is extended by modeling on Microsoft's PowerPoint (see Figure 20), a widely used application with which many of the target caregivers may already have experience. With these metaphors in hand, the user can assemble scripts by selecting image and prompt binary files, represented by the image itself and the file name of the wav file, respectively. The application then would allow previewing the assembled script in a window of the same aspect as the handheld prompter's screen (see the "preview panel" in Figure 19), and through the PC speakers. The user is allowed to save the script to the underlying Sybase database (see the "menu" in Figure 19 and section 5.4.3). The application also allows retrieval of existing scripts from either the local or networked databases (see section 5.1.3.2). A script, once loaded into the script-design-environment, can be changed and saved as a new script or saved as a revised version under the same name. A "preferences" menu option allows the user to turn on various options such as autosave and to change the behavior of the diverse graphical user interface (GUI) functions of clicking and dragging. Help is implemented as a series of short video clips, each one demonstrating an element of a

typical task (see Figure 13). The menu also has an “undo” option, which allows the user to undo the most recent action. Like PowerPoint, there is also an option to display the whole script, in several thumbnail sizes, in columns to the right of the main screen.



**Figure 20 - PowerPoint User Interface**

The MAPS script-design-environment is intended to support both design by modification and design by composition (see section 3.1.1) [Fischer et al., 1998]. To support design by modification, the caregiver is provided templates for many scripts from existing repositories of scripts [Saskatchewan Learning – Special Education Unit, 2003] obtained from rehabilitation and assistive technology professionals. The caregivers can load the scripts, which have empty stubs for images and wav prompt files, but with the filenames<sup>8</sup> taken from the captions of the steps in the existing scripts. These scripts are located on the networked database server, and can be used as templates and stored as completed scripts after modification on the local machine that is running the MAPS script-design-environment.

<sup>8</sup> The filenames for the images are descriptions of the image, and the filenames for the wav files that are the verbal prompts are the words contained in the wav file.

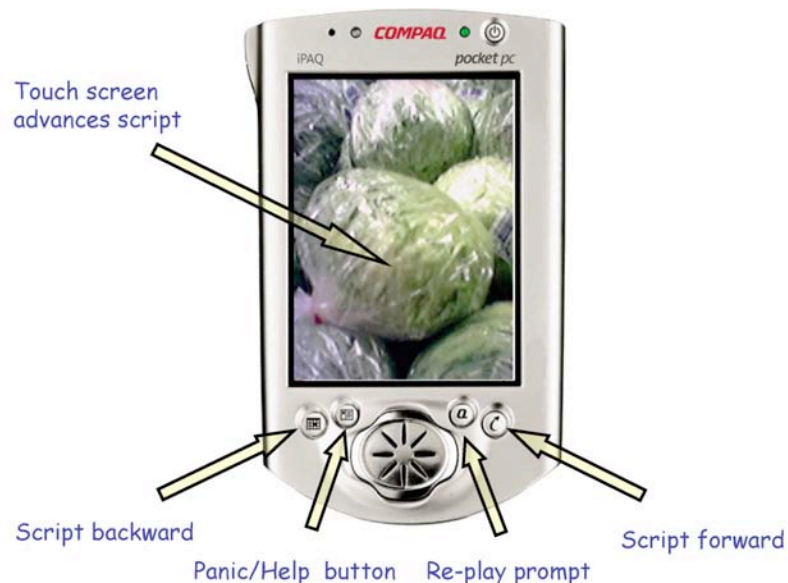
Successful scripts can be made anonymous and submitted to a moderator to be added into the Internet repository database.

### 5.2.2. Mobile Prompter

The MAPS-prompter is the unit that presents to the client the multimedia scripts that support the task to be accomplished. Its function is to display the prompt and its accompanying verbal instruction. The handheld has a few simple controls (see Figure 21): the touch screen functions as one large button that advances the script forward one prompt, and the four hardware buttons on the bottom are mapped to:

- Back up one prompt
- Replay the verbal prompt
- Advance one prompt
- Activate panic/help status (functional only in some prototypes (see section 8.3 ) )

The mapping of the buttons to functions is configurable to the individual user.

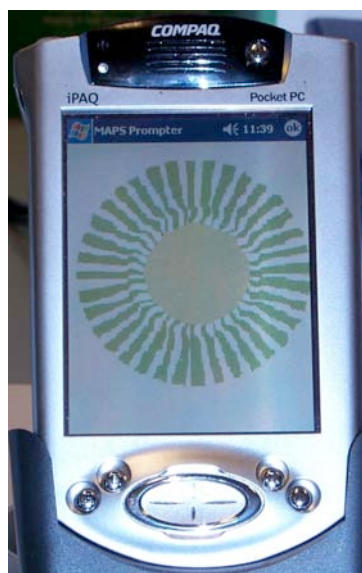


**Figure 21- MAPS-Prompter Functional Detail**

Scripts are transferred to the MAPS-prompter when the MAPS synching application is run while it is in the ActiveSynch cradle. When the new script option is chosen when running the MAPS-

prompter application, upon tapping the “sunburst” logo, the screen adds a smaller sunburst inside the larger one (see Figure 22) to let the user know that a script is being loaded from the PC via the Universal Serial Bus (USB) cable. Upon successful uploading, the application then displays the first prompt of the newly loaded script.

As a script is run in the MAPS-prompter, a log is written of the sequence and order of the script prompts displayed. Written into the log are the time the step is loaded and the details of the loaded step (see appendix 13.9). The log is used for script analysis in the evaluation portion of the research.



**Figure 22 - MAPS Scriptor Splash Screen**

**Design Rationale:** The design of the MAPS-prompter depends highly on the needs and abilities of the part of the target population that has cognitive disabilities, and the context within which it is intended to be used [Carmien & Hudak, 2003]. There are three parts to the design: the image display, the audio output, and the controls. The image size has to fit on a device small enough to be carried in a pocket and capable of being held in one hand while performing a task. It also needs to be capable of presenting medium resolution (approximately 150 dots per inch (dpi)) images sized large enough to recognize image details, on the approximately 2 × 3-inch handheld screen. The prompter needs to be able to play back recorded verbal prompts at a listenable volume. The controls on the MAPS-prompter need only be rudimentary (e.g., a keyboard is not required); however, mapping the controls to

individual requirements is necessary, and this mapping can be changed to fit an individual user's preferences and needs.

### **5.3. The MAPS Socio-Technical Environment**

MAPS is a socio-technical environment that aims to address the abandonment problem of many AT systems [Kintsch & dePaula, 2002]. As such, MAPS presents two separate interfaces, one for each of the core users of the tool: the client and the caregiver. Because the component of MAPS that gives caregivers the ability to adapt the MAPS system to the specific needs of the person with cognitive disabilities who will use MAPS cannot be anticipated by the MAPS designer at tool-design-time, MAPS must be constructed in such a way that it has the flexibility to do the configuration that the “universe of one” environment requires. This document refers to the **MAPS**-prompter and to the **MAPS** script-design-environment to distinguish between the two, and **MAPS** with no modifier refers to the system as a whole, as mentioned in the introductory chapter.

Often the limitation for people with cognitive disabilities is not an inability to complete a task, per se, but rather an inability to manage the cognitive load of remembering the sequence of steps required to complete a task or how to problem-solve when an error occurs. Traditional models require that either caregivers monitor individuals with cognitive disabilities as they complete basic tasks or that the task be performed by memorizing the script sequence, making the number of possible tasks much smaller than the number that could actually be accomplished, and thereby limiting true independence.

#### **5.3.1. Prompting**

Studies of task cuing provide a background for design and study of computationally based prompting systems [Lancioni et al., 1999; Lancioni et al., 2000; Lynch, 1995]. Much of the work on computer-aided task support for people with cognitive disabilities has focused on fairly simple, linear tasks. These can be described as repetitive, unchanging tasks, such as packing boxes, janitorial work, or making breakfast. By using these tools, the user will often learn the task, which reduces the need for the cuing agent [Davies & Stock, 1996].

Existing systems that support prompted task completion by persons with cognitive disabilities typically utilize a computer that provides previously created verbal and visual prompts that guide the user through performance of the task. Typical tasks may be cooking a simple meal; the screen displays the ingredient to be used and the verbal prompt directs the user to perform an action on that ingredient (such as to mix ingredients).

### **5.3.2. Low-Hanging Fruit and Completeness**

In the process of designing MAPS, trade-offs needed to be considered. Designing a system to provide real help to the MAPS population, when looked at as an abstract problem that needs to be completely solved, initially looked like a very large problem. To account for all the possible orders of a correct sequence of actions in supporting even the simplest of activities of daily living (ADLs) would result in a combinatorial explosion of paths. To account for and support even a majority of possible task completion paths would become a daunting task. As an example of how truly expensive “complete” solutions are is the solution of hand washing for clients with Alzheimer’s. This simple task took a team several years and required an instrumented bathroom to work properly [Mihailidis et al., 2004].

The principle that kept MAPS from becoming an intractably difficult problem involved reconsideration of the original goal: to support independence for a realistically large population, a subset of all of the persons with cognitive disabilities. Domain experts considered “low-hanging fruit” of this subset of the population to be large enough [Braddock, 2006]. This subset population needed to be bound further by the limits of “could use the system” and “needed to use the system” (see section 2.1.3). Given that, an existence proof could be made to show that it was possible to create scripts that would guide the user to do real (not trivial) tasks for ADL support. Assumptions had to be made about partial order plans and the resilience of this band of the population of persons with cognitive disabilities as a basis for the design of such a system. The evaluation section (see chapter 6) documents that this goal can be reached. The section on further work (see section 8.2) contains thoughts about completeness and how to approach the hard artificial intelligence (AI) problem of error trapping and correction.



### 5.3.3. Forking and Looping

Most domestic tasks can be modeled in linear scripts; cooking, cleaning, and the like can all be supported with a series of prompts in a script. However, many employment tasks require repetitive iterations over a series of prompts, and other employment situations may require choosing between alternate tasks at the conclusion of one part of the whole job. To accommodate this disparity, MAPS was modified to support these requirements, being designed with the job coach who was the caregiver of one of the realistic field trials. There are two versions of the prompter binary: one for primarily linear scripts and one for primarily multiscript task support. Either can be switched to the other by use of an Easter egg<sup>9</sup> located in the splash screen (see Figure 23). The MAPS-prompter was modified to load the first visual prompt from two or three scripts loaded into the prompter. They are loaded into center bottom and upper right and left on the screen (see Figure 24).

## 5.4. Technical Approach

The technical aspects of the MAPS system are discussed in this section, working from the ground up—implementing the system and starting with the hand prompter. Then the technical aspects to the PC-based script-design-environment are discussed; and finally, the database supporting stable long-term storage of the script elements and the underlying schema is illustrated.

### 5.4.1. Prompter

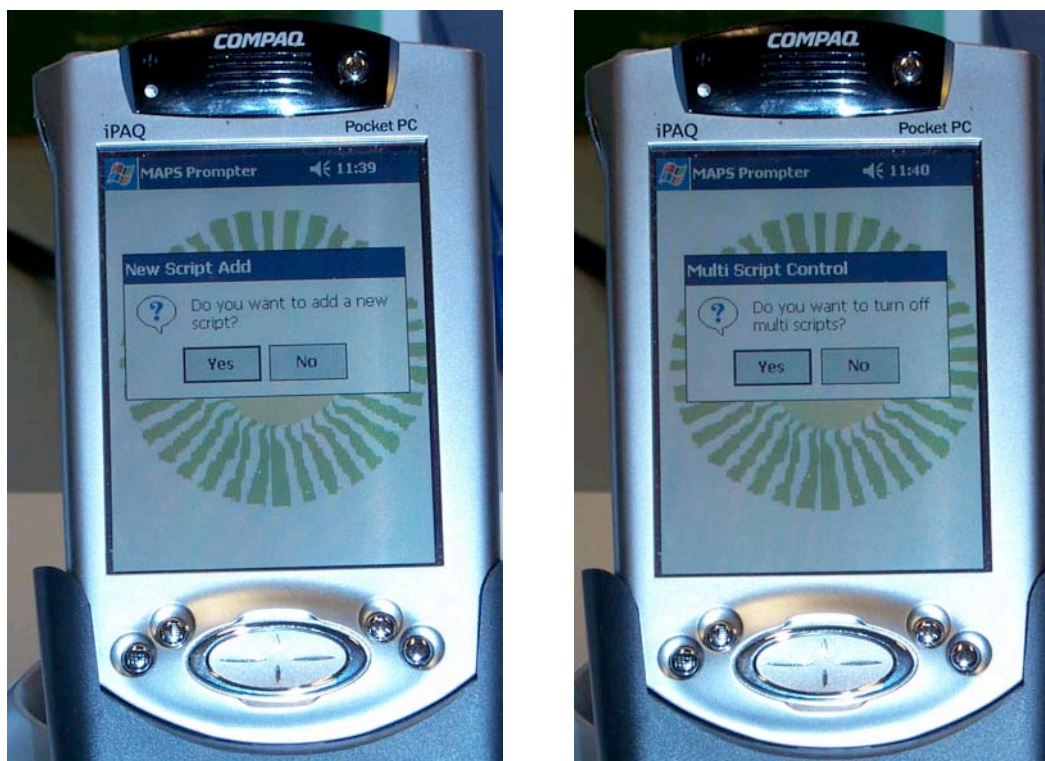
In choosing a hardware platform for the handheld MAPS-prompter, the first requirement was that it possess natively multimedia capability, which narrowed down the choices (in 2002) to devices that ran on Microsoft's Compact Edition (CE) operation system. The current platform for the MAPS-prompter is an IPAQ 3850, but because it is written for any machine that runs the WIN-CE operating systems family, the prompter may have cell phone and GPS functionality. Interestingly, as the IPAQ line matured and developed Wi-Fi, cell phone, and Bluetooth ability, the small speaker built into the

---

<sup>9</sup> An “Easter egg” is a portion of the screen that does not present itself as a GUI affordance (in this case a part of the upper left white background of the prompter splash screen) that when interacted with (i.e., clicked on) provides access to hidden functionality. ‘Easter eggs’ are common in video games.

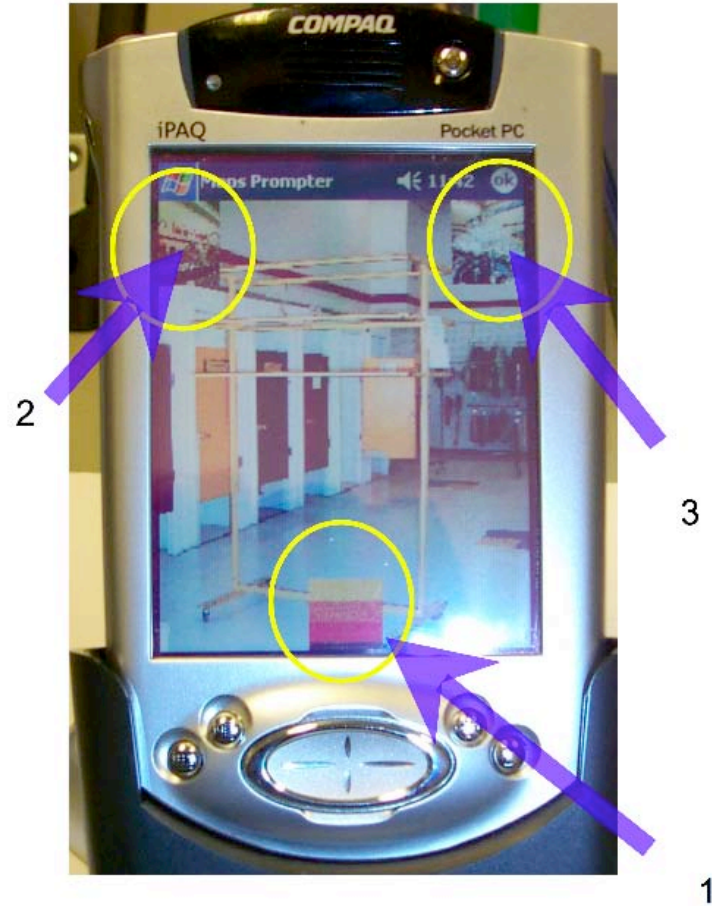
machine shrank and became not as loud or accurate in playing back wav files. Although this was a sound response to market pressure, it forced the field trial units to be IPAQ 3850s. Fortunately for this project, Hewlett Packard made available to the project a box full of these devices that had been obsolete for several years. These devices were distributed to the field testers and to a group in the Boulder Valley School District (BVSD) system who desired to use the MAPS system. The IPAQs were equipped with secure digital flash memory cards with between 256 megabyte (MB) and 1 gigabyte (GB) capacity, which allowed backing up the entire system so that when the main batteries were (inevitably) completely discharged, the prompter could easily be brought back in minutes to its original functionality. One of the advantages of using even obsolete PDA technology is that the user now appeared to “fit in” with the technically savvy crowd, thus avoiding the problem of cosmesis [King, 1999] (or “dorkyness”) that has been anecdotally noted by several AT experts to lead to abandonment [Kintsch, personal communication, 2002] or at least make the MAPS-prompter less easily lost by a client.

The prompter software was originally written in embedded VB (Visual Basic), and then ported to the faster and more flexible C# .net environment. The prompter software comes in two selectable “flavors”: single-task support or multitask support. This selection is made at the initial screen, which has an Easter egg area that allows the caregiver to update the script(s) on the machine and change the scripters behavior from the default (see Figure 23). There are two versions of the compiled software, depending on whether to default to multi or single.



**Figure 23 – MAPS-Prompter Easter Egg Dialog Boxes**

The multiscript version allows the caregiver to present the looping and forking behavior that is critical for employment task support. The caregiver can load up to three scripts into the prompter from the PC-based script-design-environment. The prompter, upon initial startup, takes the first image of the scripts and puts them in alphabetical order, with the first script in the lower middle, the second in the upper left and the third in the upper right. At any point in the currently running script, the client can switch to another script (or restart the current one) by clicking on one of the icons (see Figure 24). The caregiver, by careful script creation, can then support looping with the proper verbal prompt (e.g., “if there are more clothes to sort, press the middle icon”), and forking (e.g., “if you are done with the clothes sorting, you could choose to pull hangers by using the icon in the upper left, or rag the shelves, which is the icon in the upper right”). The programming of looping and forking requires substantially more caregiver training.



**Figure 24 – MAPS-Prompter in Multiscript Mode**

#### **5.4.2. Script-Design-Environment**

The MAPS script-design-environment (see Figure 25) is built on a modern Microsoft OS (Windows 2000 or XP) platform to ensure that possible system testers would be likely to have a PC with the right OS in their homes (for the parents) or office (for the job coaches or group home staff). Because it is based on a Microsoft OS, there is an inherent ability to connect to and support PDAs that run the Microsoft WIN-CE operating system, which at the time of the initial prototyping was the only handheld operating system to truly support multimedia output. The MAPS script-design-environment uses cookie-like initialization files to maintain script-design-environment configuration information from session to session. Visual Basic 6.0 also provided good support (via open database connectivity, or ODBC) for database integrations with the applications. The drop-down menus are consciously modeled on Microsoft's “standard” application menu, as well as the location of the help and

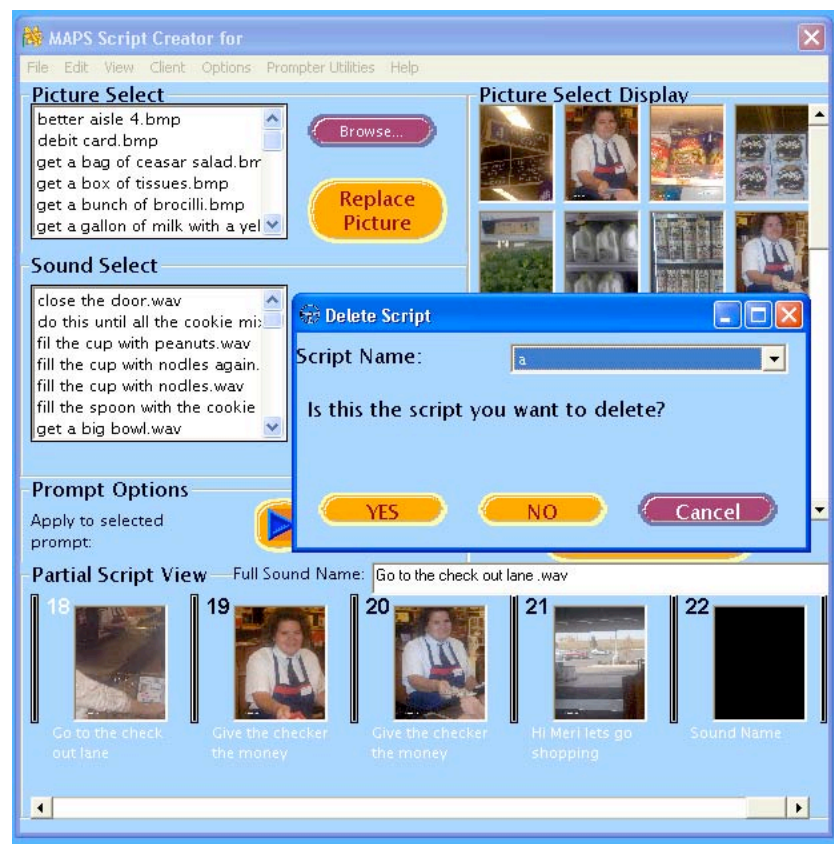
configuration utilities. The initial script-design-environment design's look and feel was typically clunky and squarish, as is common in VB applications, but as the design functionally matured, an undergraduate computer science major who also had experience as an artist redesigned the interface, and the resultant look and color scheme gave the script-design-environment a more contemporary and sophisticated look. During the field trials, small bugs in the system were discovered and corrected. Additionally, two significant functional changes were added. The first was better support for transferring the completed scripts to the prompter. Whereas previously the caregiver had to run several batch files to prepare and start the synching in the changed version, the menu item brought all these into the menu selection, thus lowering, for the caregiver, the complexity and difficulty of loading the scripts. The second functional change was requested by one of the caregiver-trial participants who had found the process of scrolling back and forth over the names of the wav files containing the prompt verbiage for a given step frustrating. The change applet allowed sorting and inserting wav files in a more natural fashion.



Figure 25 – MAPS Script-Design-Environment Detail

### 5.4.2.1. Basic metaphor

As stated above, the user interface for the script-design-environment was consciously based on a filmstrip metaphor. In this model, the bottom part of the screen displayed five prompting steps with a slider bar beneath to move ahead and back. The following description of the user interface (UI) is based upon the default configuration but, dependent on the user configuration of the script-design-environment, there are typically many different ways to do the same action in the script-design-environment (e.g., double clicking on a wav file name in the browser frame may either insert the wav file into the currently selected filmstrip prompt or it may preview/play the sound on the PC speakers) (see Figure 25). The File menu option allows the user to load, delete, or save existing scripts (see Figure 26) as well as to load templates to be used for design by modification sessions.

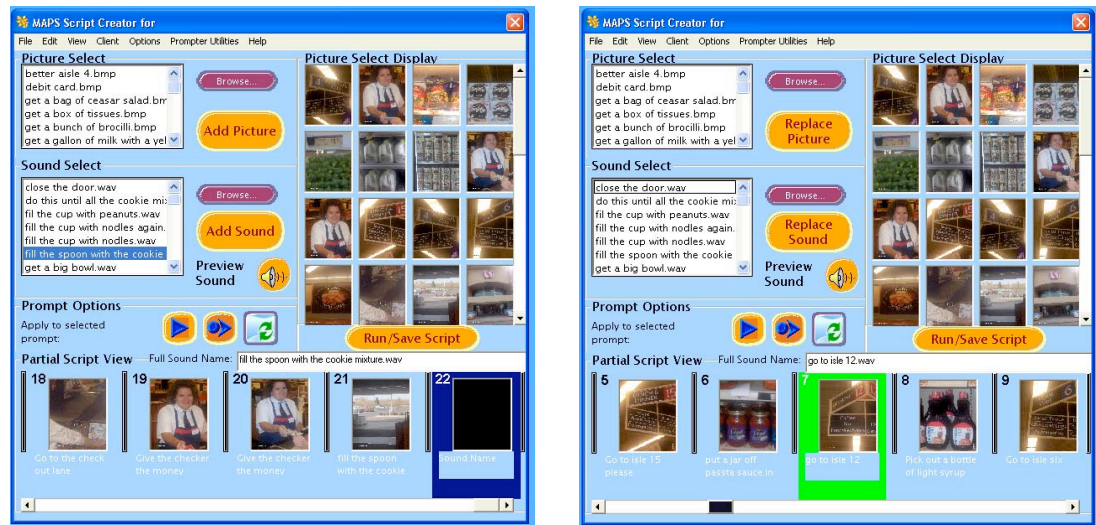


**Figure 26 - File Options: Load and Delete**

The edit menu supports undo and redo functions (for up to seven levels of actions). The menu also supports actively switching between the “append” and “insert” modes. These controls are typically

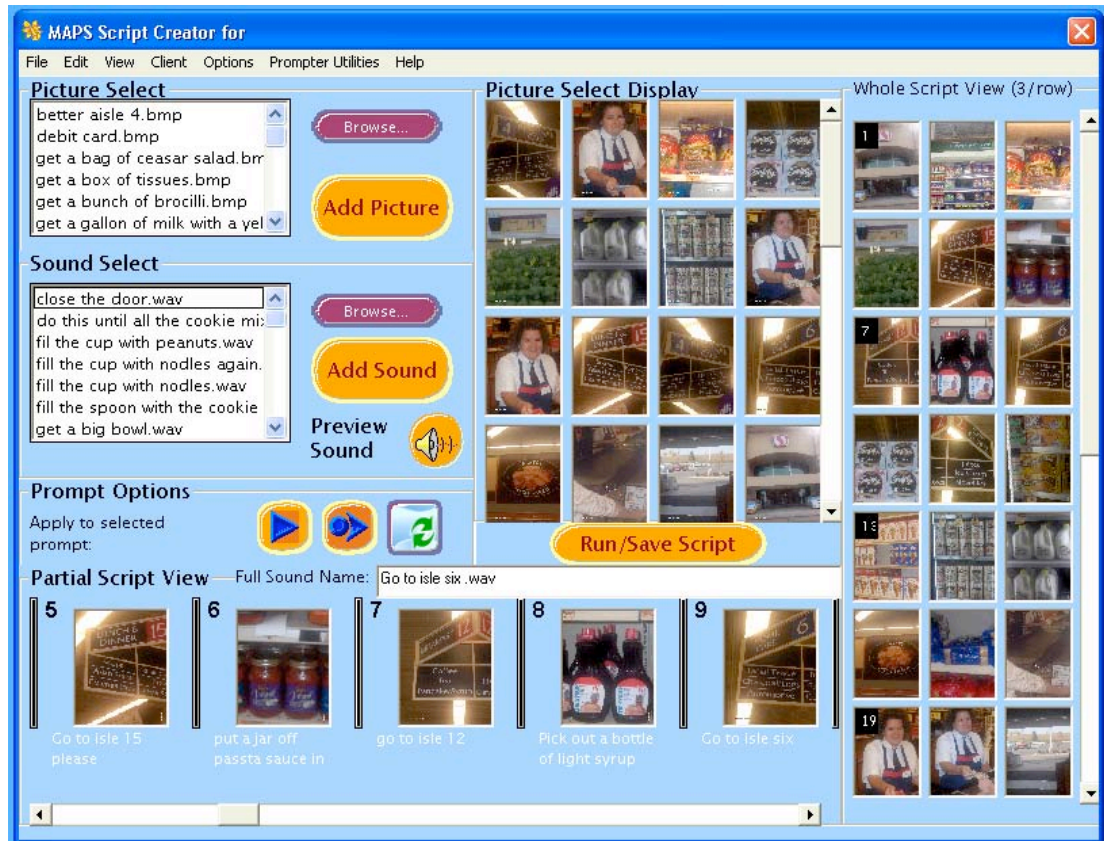


not used very much. In the process of iterative design of the script-design-environment, the append and insert (or edit) modality of the script-design-environment—not unlike the vi UNIX text editor—was pushed into the background, effectively automatically switching from one to another, depending on the actions of the user and the state of the application, by careful testing and UI coding (see Figure 27).



**Figure 27 - Script-Design-Environment in Append (left) and Insert Modes (right)**

The script-design-environment goes into append mode (with a flashing blue border around the current working box) when the cursor (the highlighted box around a given prompt) has no prompt filled with image and wav files to its right. Similarly, if the user wishes to insert a new prompt between two existing prompts, clicking on the vertical bar between them inserts an empty prompt box. Deleting a prompt function can be invoked by highlighting the prompt to be deleted and using the edit menu item “delete” or highlighting a prompt and clicking on the trashcan above the filmstrip bar. The view menu allows the user to display thumbnails of the complete script to the right of the main script-design-environment form—either three or four to the row (see Figure 28). When a user locates the prompt of interest, clicking on the thumbnail repositions the filmstrip so that the selected one is in the middle position and moves the cursor to that prompt.



**Figure 28 - MAPS in Visualize Whole Script at Once Mode**

The client menu does nothing; it is a stub for further development. The options menu has but one element—allowing access to the user preferences form. This form is divided into five tabs (see Figure 29): user preferences, auto-save preferences, add sound preferences, sound path select, and picture path select. Each of these tabbed panes gives the user from one to three options to customize the script-design-environment to the user's preferences and MAPS script-design-environment experience.



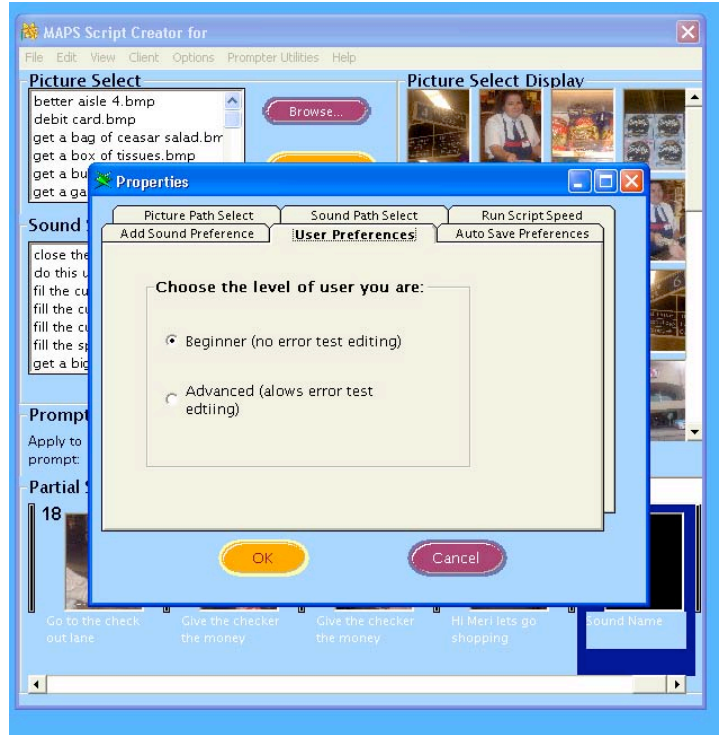


Figure 29 - MAPS Script-Design-Environment User Options

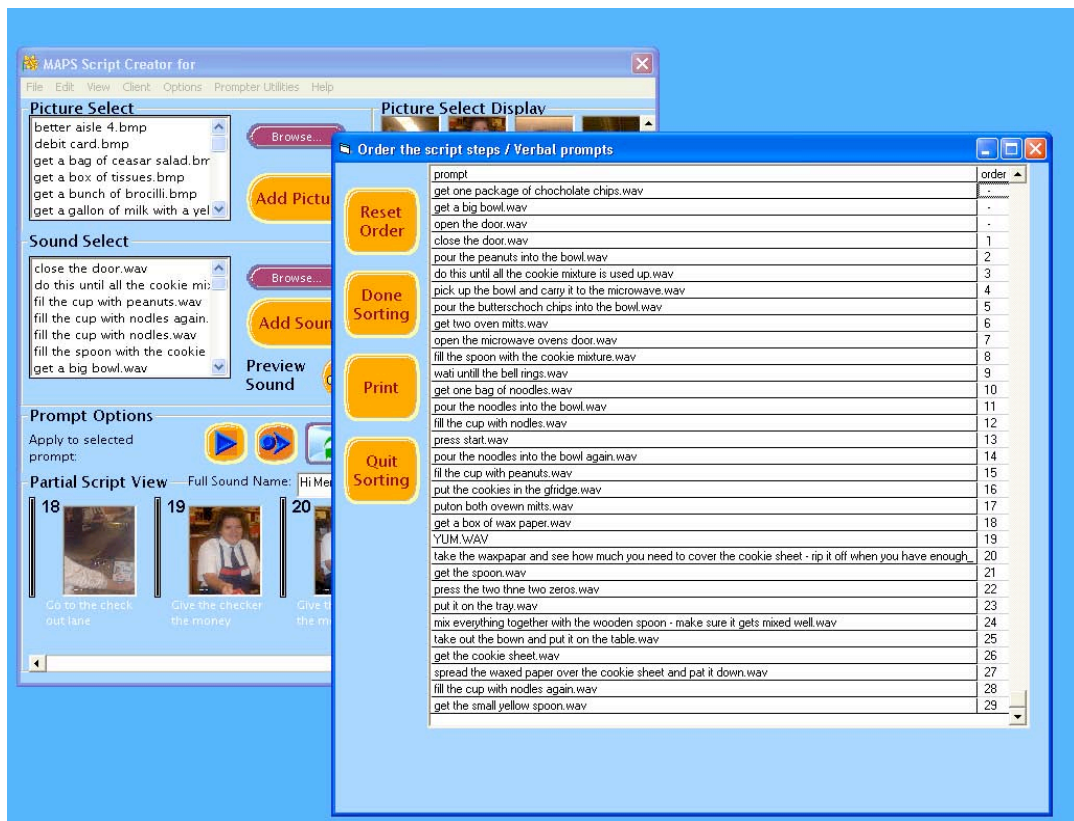


Figure 30 – Wav Filename Sorting and Insertion Tool

The “Prompter Utilities” menu allows the user to prepare a script (and run the database utilities) for loading into the handheld MAPS-prompter; and to activate a utility that allows sorting and insertion of recorded prompts directly into the active filmstrip prompts at the bottom of the screen. This utility was specifically requested by one of the realistic trial caregivers to surmount the problem of meaningful filenames and recorded prompts (see section 8.2.3.) Specifically, she requested an applet that would allow her to stage the wav file names in the order that reflected the script order, and then when the ordering was done, would allow her to insert the files one at a time into the script as she proceeded to assemble the script. In use, various caregivers found this of greater or lesser help in making their scripts (see Figure 30). Finally, the help menu provides the usual text-based help and “about this application” page as well as a video help system (described below). Additionally, the “about MAPS” menu option brings up information about the MAPS script-design-environment application and Microsoft’s standard system information screen. In the lower right of the form is an Easter egg control section that allows one to switch the MAPS database from the local versions (which store finished scripts) to a networked version of a MAPS database (see Figure 15 and section 5.4.3) that holds 168 template scripts.

#### 5.4.2.2. Help system

It is common knowledge that text-based help systems are either not effective or never looked at, by and large. MAPS provides rudimentary text-based Microsoft-formatted help screens, but the majority of help for the user is provided by video clips and tool tips. Video help was created using a video capture tool, Camtasia, and overlaying the short videos with a commentary voice. These short videos, typically from 30 to 90 seconds in length, were cued by the controls in the video help pane (see Figure 13). The pane in Figure 13 shows three tabs: script help, prompt help, and view/preferences help. Selecting a tab presents from two to six videos that can be viewed by pressing on the VCR-like controls. Additionally, “tool tips” contextual help (see Figure 31) was supplied for every control on the main screen and subscreens. If the mouse cursor lingers on the control for more than a few seconds, a short (two to ten words) explanation of the control’s use and function is displayed. The tool tips change, depending on the user preferences selected; the tool tips typically present more information to the advanced user than to the self-flagged beginner.

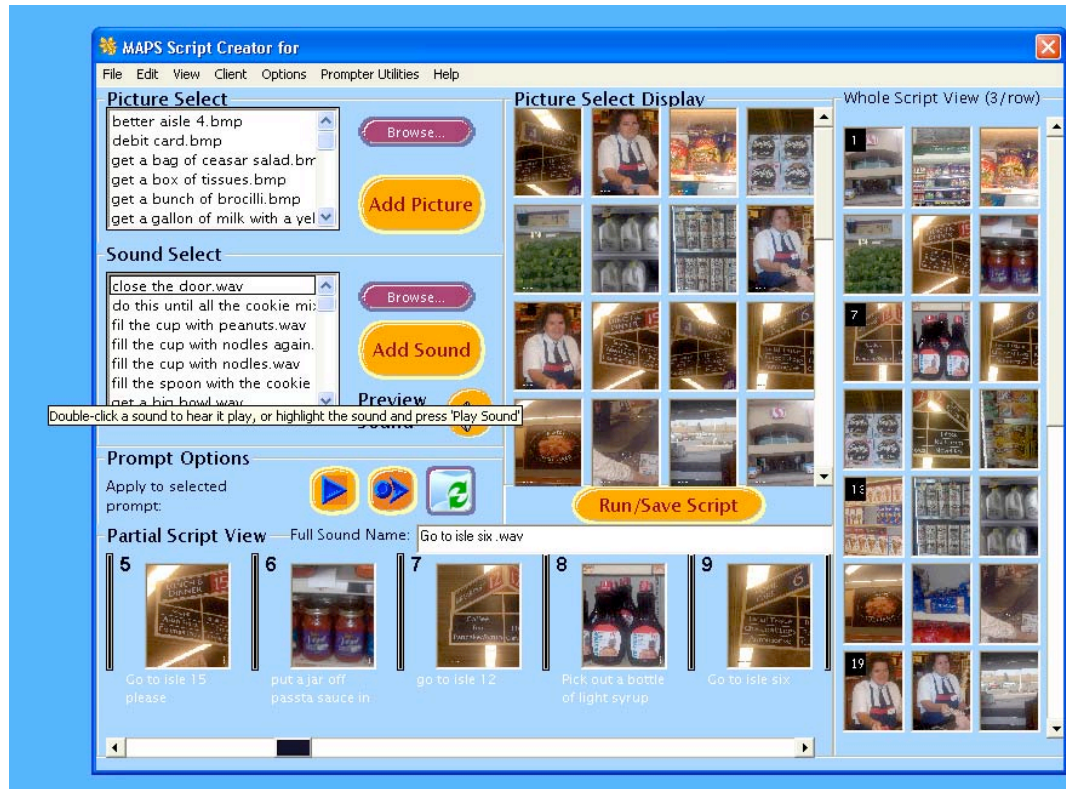


Figure 31 - Tool Tip Help

### 5.4.3. Database

The MAPS system uses a database to store and display scripts and their component prompts. For this, a Sybase iAnywhere SQLAnywhere 9 database was used. Sybase was selected due to its strong support of the same database across platforms, specifically Microsoft XP on desktop systems and various “flavors” of CE on handheld devices. In using a database manager instead of the file system on the various devices, MAPS could ensure that the integrity of the files and the relationship between members of the MAPS stable storage system would be enforced. The schema of the PC-based database tables can be seen in appendix 13.8. A hierarchical view of the components shows, at the top, scripts, which are composed of script nodes and are themselves registered as belonging to a repository (script location) and category (for searching templates). On the handheld side, the Sybase Ultralite database manager holds script nodes for display in a table called “Current Script.” Beyond this basic functionality, the MAPS database has tables to support the wizard-like configuration of creating error trapping and error correction script annotations for communicating with the Lifeline service, allowing

active distributed task support (see section 8.3). This part of the schema allows the MAPS script-design-environment to configure itself, depending on the error traps and error recovery that the caregiver desires.

## 6. Assessment and Field Studies

Many approaches exist for studying a complex computer system [Pfleeger, 2001]. One approach studies its functionality by using quantitative measurements, usually in isolation to its intended users and environment. By isolating the device from its context, a more specific and generalizable set of metrics can be drawn. Another way of approaching evaluation is to dig deeper into the system and test the components with the aim of identifying subcomponents that may be bottlenecks and cause performance problems in scaling up the system to its intended use. Here the issue is neither accuracy nor efficiency, but rather how to understand the process of adoption of high-functioning AT systems [Newell, 2006]. The literature and best practice [Martin & McCormick, 1999] reviews show that a significant contributing factor to AT device abandonment is the lack of initial fit and difficulty in modifying these devices to the current needs and abilities of the end users (persons with cognitive disabilities as well as caregivers). With this goal in mind, MAPS needed to be studied in situ, with real problems being solved by real users.

Good design uses evaluation and usability studies as part of the design process [Carroll, 2003]. The initial design of the hand-prompter was quite close to the final design. This was due to the smaller number of controls that the user needs to use for interaction and the simple operation of the prompter (see section 5.2.2). There were, however, a small but interesting number of changes in the prompter that developed due to the field tests. The script-design-environment was a different matter—there were multiple, incremental usability tests, with re-designs between each one.

Once the MAPS system prototype was close enough to complete for field-testing, ethnographic studies were performed with the whole MAPS system on caregivers' machines, doing real tasks. By and large, computer scientists are not formally trained in doing this sort of fieldwork. However, for adoption, it is critical to understand the people using an AT system as well as the environment and the tasks to be performed with it. There are two goals in doing this: (1) to understand the world of the person with cognitive disabilities, and (2) to understand the process of adoption (or rejection) of the MAPS task support system. Understanding the process implies that the process has to be tested in the real world (field trials), not in laboratory-sized chunks. This kind of understanding can be readily and

naturally gained by closely observing the participants in action and applying an analytical technique to the observations. Ethnographic participant observation [Bernard, 2002] provided the structure for the field trials.

## **6.1. Methodology**

Two methods were used in the design and evaluation of the MAPS system. The first—traditional usability testing and analysis—was used during the design of the script-design-environment with caregivers. The second—ethnographic participant observation—was employed during the field trials of the system with both clients and caregivers.

The participants of the realistic field study (see section 6.6) were in dyads consisting of a person with cognitive disabilities and a caregiver who had an ongoing relationship with the individual with cognitive disabilities. The participants in the initial prompter study (see section 6.2), the images study (see Section 6.5), and the initial script test (see Section 6.4) were persons with cognitive disabilities, as above. Participants in the script-design-environment usability study (see section 6.3) were caregivers and special education professionals. The subjects with cognitive impairments were 16- to 28-year-old young adults transitioning to independence (from school to working and living in less-supported environments), and one 37-year-old person with cognitive disabilities living in a group home diagnosed with mild to moderate retardation (DSM-IV classified as IQ 50-55 to 70 and upper range of moderate as IQ 35-40 to 50-55). The subjects with cognitive impairments worked well with verbal instructions and did not have significant behavioral issues. The caregiver subjects consisted of adult caregivers of the subjects with cognitive disabilities. These cognitively normal individuals included, for example, parents, teachers, and group home staff. The caregivers were selected to have moderate experience with computers (i.e., able to write a short letter on a personal computer) and some experience with prompting in this context, typically for training of persons with cognitive impairments [Carmien & Fischer, 2003-2006].

Students of average IQ and chronological ages similar to the young adult participants with cognitive disabilities were recruited for the image attributes experiment (see section 6.5) to provide a baseline for the collected data. All of the participants for this part of the study were recruited with the

help of Anja Kintsch, a consultant to the CLever project, and an assistive technology specialist with the Boulder Valley School District (BVSD).

Additionally, some of the subjects were recruited with the aid and under the guidance of local experts such as the BVSD special education staff, and Imagine! of Boulder County (see section 6.6.2). Possible subjects were identified on the basis of personal and professional connections between the target population and assistive technology professionals. Parents, guardians, and the individuals themselves participated in deciding that this study was appropriate for them. Specific subjects with cognitive impairments were identified and chosen as appropriate for specific areas of task accomplishment (e.g., someone already familiar with cooking was selected for the cooking task, and similar selections were made for the housework, shopping, and employment scripts).

As one part of the dyad, the person with cognitive disabilities was considered a vulnerable population, due to both cognitive ability and, in some cases, age. The University of Colorado and the BVSD required substantial documentation of the process and projected use of results of any interaction with these persons. The process of getting Human Research Committee (HRC) approval [Carmien & Fischer, 2003-2006], which took several months and extensive documentation, resulted in a research protocol that was well thought out and, in the end, flexible enough to accommodate small but important changes in the procedure

### **6.1.1. Usability Testing**

The design of MAPS, particularly the script-design-environment for caregivers, was based on the techniques of Lewis and Rieman's "Task-Oriented Design" [Lewis & Rieman, 1993]. The initial design came from early scenarios of a caregiver creating and modifying a task-support script. Once the basic functionality was determined, users were brought in to study their relationship with the interface and thereby to improve it. Three kinds of users were selected:

- AT professionals and special education teachers from the BVSD special educational team.
- Parents of students in the BVSD special education and the bridge transition programs.
- Professional caregivers who were staff with Imagine!, an organization that runs group homes for persons with cognitive disabilities in Boulder County.

Participants' recruitment was facilitated by Anja Kintsch, an assistive technology expert who is part of the CLever team, employed half time by the BVSD and half time by CLever as a consultant. The specific techniques used included the "think aloud" protocols [Lewis & Rieman, 1993], in which the subjects are encouraged to speak aloud the thoughts and desires that they are experiencing as they attempt to accomplish a given task. All participants in the script-design-environment usability test were audio and video recorded with the camera focused on the hands of the users being tested and the screens they faced. Participants were given some specific tasks (see appendix 13.3), and input from noting the forms and situations in which breakdowns occurred resulted in improvement to the design of the script-design-environment. Also at this time, participants were led in a semi-structured interview [Bernard, 2000] that started with some general questions about the MAPS system and prompting and allowed participants to go into some detail about their thoughts.

### **6.1.2. Ethnography**

Typical tests of computer systems focus on narrow, controlled behaviors in laboratories, but the research goal for this work was understanding how people interact with their environment, especially with respect to computationally aided prompting. For this goal, ethnographic techniques are eminently suitable [Nardi, 1997]. Two frequently used techniques in ethnography are interviews and observation. Interviews could be either completely open-ended, with no goal other than "getting to know" the target population, or structured. Structured interviews can span interviews that are so structured that the results can be plotted by using a Likert scale (or similar measure) to semi-structured interviews, where the interviewer has a list of topics and steers the conversation toward them one at a time, but without putting constraints on what else is brought up nor how long each topic is discussed [Bernard, 2002]. The goal is to understand the participants' world in more detail and depth than could be obtained by simply observing them. An interview can be a good way of quickly getting to know the terrain of the domain; however, it does lend itself to the pitfall of interviewer bias, both in the choice of topics and in the words used by the interviewer.

Participant observation, the other ethnographic tool used, allows a more nuanced and intimate understanding of the subjects and their world. Interaction with the subjects of this study can take three



different roles, spanning from complete observer, to participant–observer, to complete participant. In this study, there is no way to “become invisible” in domestic situations, so the role of participant–observer was chosen. However, in domestic or employment situations, there is limit to the observer being a “participant”—too much participant activity, and the naturalness of events is compromised; too little, and parts of the “dance” of ADLs become obscured. Hutchins, in his seminal study of naval navigation, *Cognition in the Wild* [Hutchins, 1994], labels this process “cognitive ethnography.” In approaching the caregivers and clients in daily life and as MAPS users, an attempt was made to see events from this perspective. To do this, careful attention had to be paid to how decisions were made and what support was available for plans in the short-term, and how the representations of task support were developed and used.

Data collected in this process of participant observation ranged from field notes, to digital audio recordings, to collection of artifacts that support task completion. In each situation, field notes were taken; however, the desired level of detail could not always be captured with pencil and paper, so digital voice recordings were made [Lofland & Lofland, 1995]. Shortly after each session, the field notes were transcribed to a text document in the computer, and a second set of notes about what was said and what was done was transferred into the computer by carefully listening to the audio recordings. Often, a third set of observations was written, discussing artifacts used to complete tasks and supporting documents that were retrieved during the session [Emerson et al., 1995] .

Finally, the notes were entered into a hypertext database (Hyperresearch) and coded. The Hyperresearch software allows linearly stepping through the observations in a text file, scanning for bits of behavior or information that are an instance of a class that has been selected by the researcher by labeling it with a “code.” Regarding codes, LeCompte and Schensul define coding as “*organizing data into categories related to the framework and questions guiding the research so that they can be used to support analysis and interpretation*” [LeCompte & Schensul, 1999, p. 45]. The goal of the scientist is to let the data speak for themselves, and so this study undertook iterative coding to analyze the data and perceive emergent patterns.

The approach to analyzing data in the present study has had five parts: (1) produce transcripts of observations, (2) identify potential analytic categories, (3) label instances of categories, (4) use the set of labeled instances to build abstractions about the observed environments, and (5) present the resultant abstractions and use *exemplars* (i.e., quotes from the observations) to illustrate them [Bernard, 2002]. This method, however, begs the question of where the codes come from in the first place. Initially, the researcher has questions to be answered and selects objects of observation and situations to observe from that perspective; thus, the initial codes come from that process. The process of allowing the data to speak is important, however, so that in the process of coding, the coder allows space for new codes and unexpected connections to emerge. This is what did happen, in fact: the coding initially started with 30 possible codes but by the time of the last (fourth) dyad, the list had enlarged to 104, and about 10 of the original codes were never used. Observations and recordings of four dyads produced more than 110 pages of transcripts, which, using 104 codes, resulted in 1,298 instances of coded observations. Hyperresearch provides a utility to generate quantitative aggregate data reports about the instances of codes and dyads (see appendix 13.14). In the report, dyads are referred to as cases.<sup>10</sup> In addition to the qualitative analysis of observations, the evaluation of the field trials was based on artifacts collected from caregivers as well as photos taken from observations of script use.

A confirmation of the coverage of these four dyads was that as the last dyad was examined, very few new codes were introduced, demonstrating that, for this limited set of research topics, theoretical saturation was being approached. Themes and common modes of failure and success can be drawn from the coding of the notes [Berg, 2004].

#### 6.1.2.1. Interviews

The initial research into the domain of AT, persons with cognitive disabilities, and support communities consisted of interviewing local experts at the University of Colorado. Their expertise spanned dyslexia and communication disabilities, to assistive technology developers, to special

---

<sup>10</sup> Cases 61 and 71 are the first family dyad; 62 and 72 are the second family dyad; 63 and 73 are the dyad from the transition program; and 64 and 74 are the dyad from the group home.

education assistive technology professionals. Each interview was semi structured (see appendix 13.1). The purpose of this set of interviews was to begin the process of understanding both the world of persons with cognitive disabilities and assistive technology's part in this world. From this exploratory research, the themes of universe of one [Yoshinaga-Itano, personal communication, 2000] and learned helplessness [Olsen, personal communication, 2000], the details of prompting as a support for ADLs for persons with cognitive disabilities [Baesman, personal communication, 1999], and the general notion of assistive technology for persons with cognitive disabilities [Richardson, personal communication, 2000] emerged. The interviews also formed a basis for the some of the extensions of distributed cognition that arose in the form of a discussion about dyslexics and screen-reading applications.

#### 6.1.2.2. Participant observation

Participant observation was chosen as the principal ethnographic technique to study the realistic field trails of MAPS with dyads. Rather than controlled experiments, ethnographic studies were chosen for the evaluation and analysis due to consensus (based on the preliminary studies, the recommendation of the assistant technologist on staff with the CLever project, and literature reviews) that for such a sufficiently complex situation, isolating the system in a lab environment would result in a flawed and biased set of results.

The first and most important reason to do participant observation is that, by doing this, the participants in the trials (after an initial "getting to know you" period) interact and behave close (or closer) to naturally. Without this acculturation period and technique, the details of the introduction to, training in, and use of the MAPS system might be overwhelmed by the Hawthorne effect. The Hawthorne effect refers to situations in which subjects might be particularly vulnerable to changing their behavior based on the introduction of the observer. This effect was observed in the personality of many of the persons with cognitive disabilities who were trial subjects. One objection to the validity of this study might be that typical anthropological participant observation studies take a year or more of fieldwork, and this study was done in about six months, in which observation took only about three weeks of contiguous time. The reason that even this small amount of time spent was sufficient was

based on the specific information that was being looked for and advice from several experts in this technique. The approach taken for this part of the research was *rapid assessment* [Bernard, 2002]. By starting out with a list of inquiries and having a good idea of the contexts that might provide answers, much can be learned in the space of several weeks.

## 6.2. Design of the Handheld

Early in the design process, it became clear that that a preliminary test of multimedia prompting on the IPAQ handheld computer by persons with cognitive disabilities was necessary before proceeding further with the MAPS system at all. The concern expressed by members of the computer science community (in a review of a paper submitted by the CLever group) was that individuals with IQs below 60 might be incapable of using the proposed system. With the help of Anja Kintsch, the AT professional member of CLever, a simple script was constructed and representative participants from the BVSD high school population were recruited for feasibility testing. The script guided the users through assembling a plastic glider airplane toy. The script contained seven steps and guided the user through opening a bag with the parts, counting the parts in the kit, pushing the large wing through the fuselage (see Figure 32), pushing the rear horizontal wing through the fuselage, attaching the nose and launching the glider (see appendix 13.10 for details). The verbal prompts and images were collected and installed on the prototype MAPS-prompter, which was instrumented by logging all interactions with the MAPS controls and the name of the step as well as the time of the event.



Figure 32 - Two Prompting Steps from the Glider Script

Seven young adults with cognitive disabilities who expressed an interest in doing the experiment were selected from the BVSD population of special education young adults with cognitive disabilities with help from Anja Kintsch. The appropriate human research protocols and permission forms were completed, and the test was conducted in a local high school. The participants' ages ranged from 14 to 18, with an average age of 16.16. Their IQ's ranged from the 40s to mid-60s, with an average of 50.<sup>11</sup> Four of the participants were male and three were female. The prototype had two scripts loaded on it: a demo script that illustrated the use of the controls (i.e., move forward one prompt, move back one prompt, replay the verbal prompt, and a button to request help) and a script to assemble the plastic glider.

**Table 3 - Initial Prompter Test**

Participants	Method / Procedure	Data Collected	Result	Interpretation of the Results
8 BVSD special Ed students	Demo of prompting and using a prompter to assemble glider model	Observation, semi-structured interview, audio recording, script log	Seven succeeded six unaided; one succeeded with help	Data led to conclusion that young adults could successful use the MAPS prompter

The test procedure (see Table 3) consisted of a brief instruction in the use of the handheld prompter, including the operation of requesting help, which was implemented by supplying “wizard of Oz” verbal help from the observer when the help button was pushed. After the demonstration of the features of the prompter (see script in appendix 13.10), the bag with the glider kit was brought out and placed on a table. The glider assembly script was loaded into the prompter and the participant attempted to assemble the glider by following the script on the prompter. A log of the users' interactions with the prompter was collected, as well as an audio recording. An evaluation sheet filled out for each participant (see appendix 13.11) recorded specific information about details of the participants' interactions such as their visual and auditory discrimination of the prompt (including which prompt was problematic), fine-motor interaction with screen and hardware buttons, and general

---

<sup>11</sup> IQ and most other quantitative measurements of cognitive ability are *very* rough estimations of real ability; however, they are one common measurement for almost every person with cognitive disabilities, and thus IQ is used here.

observations of the interactions with the system and how well the observer felt that the participant had developed a mental model of the prompting system. The same evaluation form collected requests for help by using the panic button, verbal requests for help, and unrequested intervention. The participants' panic button requests ranged from 0 to 2, with an average of 0.57; their verbal requests for help ranged from 0 to 1, with an average of 0.28; and unrequested interventions ranged from 0 to 6, with an average of 2 interventions per subject. Two subjects required 6 unrequested interventions each, and another required 2 interventions.

All seven test participants were able to successfully assemble the glider—five with little difficulty and two with several interventions to bring them back on task. Although this result surprised both the special ed teacher and our staff AT professional, they surmised that *this* population, from affluent Boulder County, was raised with remote controls, wireless house phones, MP3 players, and cell phones (in fact, several of the young adults in the special education teacher's class had their own cell phones). The situation for these young adults, who had been raised in environments where small computers were ubiquitous, was analogous to the generational difference between those with few automobiles and telephones and no home television, and their children, who had all of these and used them as a completely natural affordance of the world. This point was made again, when, during the realistic trials of the MAPS system, a grandmother of a 20-year-old young adult with cognitive disabilities told me that her granddaughter (with an IQ of 54) had shown *her* how to operate the family's complex home theater system.

### **6.3. Design of the Script-Design-Environment**

The script-design-environment started out as a set of possible designs on paper and scenarios of how the interaction might happen, as well as several metaphors or models that the user might have for the script-design-environment. The script-design-environment was designed to use as much of the Microsoft interface standards as possible. The target user for the script-design-environment was assumed to be someone who was minimally computer literate: novice-level skills at using a common word processor and the ability to read and compose an email message using a common GUI email application. The basic metaphor for the scripts is a filmstrip (for older users) or a PowerPoint

presentation (for younger users). The visual interaction style and affordances built into the application were drawn from these models. Any changes in either the architecture or user interface were documented in a file that was part of the source code—a rich repository of the evolving design rationale of the system spread over three years (see Figure 33)

**Table 4 - MAPS Script Design Environment Usability Testing**

Participants	Method / Procedure	Data Collected	Result	Interpretation of the Results
9 people, 3 iterations (2, 3, 4) and review by AT expert	Think aloud, semi-structured interview, videotape	Compiled list of bugs and interface difficulties	Three iterations of design	Design was simple enough to use with minimal (10 min) introduction

When a working prototype had been coded, the first set of usability tests were run (see Table 4). In all, there were three sets of usability evaluations. The first set involved two typical users, after a short set of instructions in the use of the script-design-environment, performing a simple set of tasks, which were observed and videotaped. This first group exposed several bugs, which led to a substantial redesign of the interface. The major changes resulting from this first series of evaluations were autosaving, a rearranged layout for more intuitive design, and more menus added to access more functionality. The second set of three users was given a more systemic set of instructions and formal introductions to the script-design-environment.

At this point in the design, help had been implemented as a series of videos, so the instructions, besides reviewing basic menu choices, consisted of pointing out the existence of video help and a brief review of the topics available (see appendix 13.2 for the instructions and appendix 13.3 for an evaluation sheet). On the basis of this second set of evaluations, many GUI changes and one major architectural change were made. The script-design-environment had an “insert and append” mode that the user had to be aware of, and the redesign pushed the modality of the script-design-environment into the background, automatically switching back and forth.

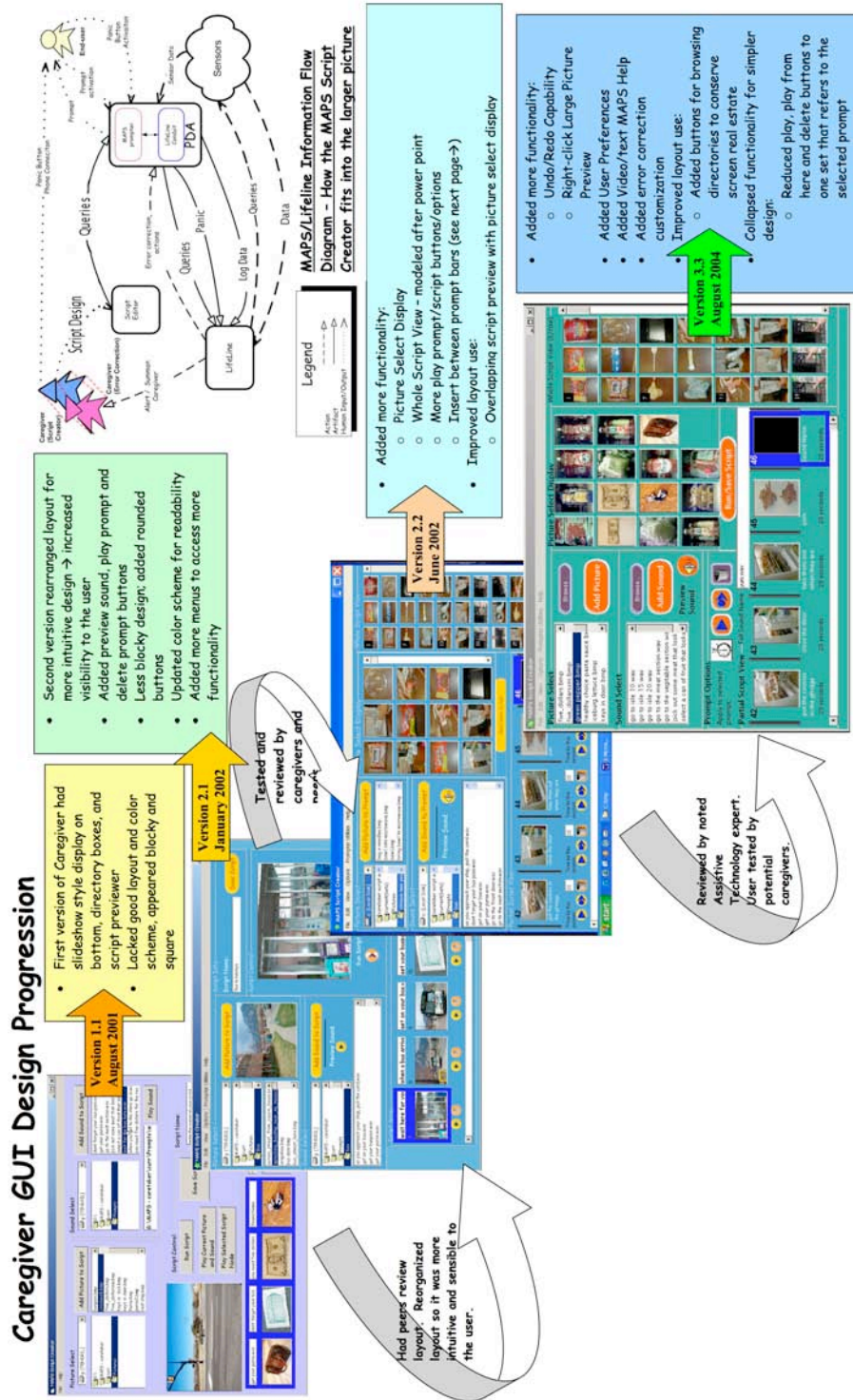


Figure 33 - MAPS Script-Design-Environment Iterative Design



The last set of usability tests resulted in adding an “undo and redo” function as well as making the color scheme and text font and font size changeable at compile time. From this last group’s input, some unused functions were removed and controls for little-used functions were pushed deeper in the menu structure, thus giving a bushier choice tree for the user. Finally, the script-design-environment was demonstrated to an assistive technology product design expert, Cathy Bodine from Assistive Technology Partners [Bodine, 2005], and the look and feel of the labels and wording as well as the menu structure were changed based on her helpful input. Table 4 summarizes this part of the research, and Figure 33 illustrates the changes in the GUI over the trials.

#### **6.4. Preliminary Field Trial of the Handheld with a Complex Script**

The previous steps proved that young adults with cognitive disabilities could use the mobile hand prompter, and the usability testing of the script-design-environment showed that it would be sufficiently easy to use by the caregiver population. The next step in exploring this problem space was to have a person with cognitive disabilities do a typically complex script similar to what would be used in the realistic field trials. A cooking script was found to have the attributes that were sought:

- It was taught in the special education classes to this population.
- It is complex enough to present a real challenge (i.e., it had many steps).
- Timing was involved (i.e., the task had steps that involved doing something for a “while”).
- Objects were acted on.
- Artifacts (tools) were used to act on those objects.

The task chosen was making Chinese noodle chocolate cookie treats (see Table 5). Part of the reason for choosing this task was that the special education teachers had already broken the task down into several levels of granularity for use by various levels of ability, the ingredients and process were fairly simple, but the process took many steps. The special education teacher and Anja Kintsch came up with a suitable experiment participant. She was 19 years old and diagnosed with Agensis of Corpus

Callosum, polymicrogyri, and other miscellaneous anomalies all of unknown etiology.<sup>12</sup> Pictures and voice prompts were recorded by Anja Kintsch and the special education teacher, and the script was assembled as part of the usability test by the special education teacher. The completed script had 42 steps. The test was conducted in the special education classrooms, which were equipped with the kitchen devices necessary for the script. The ingredients and utensils were laid out, and the participant was given the prompter. Panic button functionality was simulated with a wizard of Oz approach so that if she had pressed the prompter, it played an “I need help message.” The participant was also encouraged to ask for help directly.

**Table 5 - Chinese Cookies Script Test**

Participants	Method / Procedure	Data Collected	Result	Interpretation of the Results
19-year-old female with cognitive disabilities	Make and install 42-step script on prompter; give to participant with ingredients and utensils	Fieldnotes from observation, videotape and script log	The subject succeeded in making the cookies, but skipped some steps	Successful accomplishment of real-world task with prompter. Skipping of steps indicated internal scripts that the script builder had missed; log file demonstrated ability to automate collapsibility analysis

The subject was able to complete the task in 29 minutes. Help was requested twice by using the panic button and seven times directly. (This indicated that more training in the use of the help system would be needed in future tests.) There were also several unsolicited interventions to get the participant back on track. Although she was able to successfully accomplish the task, it was clear to the observers that the granularity of the script was too small. This fact was demonstrated by the participant coming to an instruction and then starting the subtask and continuing past the end of the prompted subtask for several steps, at which point she would turn to the prompter and rapidly step through the prompts until she came to the one with which she needed help. This sequence of actions could be read from the script activity log. An example was that when she got to step 15 (“Open the microwave oven door”) at

---

<sup>12</sup> Agensis of Corpus Callosum, polymicrogyri, and other miscellaneous anomalies all of unknown etiology. (Probably metabolic in origin but not for certain.)

10:09:31, she skipped forward to step 18 (“Press the yellow button”), going past three steps in 18 seconds, whereas a typical step was taking between 20 seconds and 2 minutes. This essential test confirmed that prompting as task support was a reasonable approach for young adults with cognitive disabilities, as well as giving encouragement to ideas such as collapsing scaffolding and script reuse.

## 6.5. Image Testing

In the process of crafting the first prototypical scripts for MAPS, and after looking at the sample scripts of commercial applications [Baesman & Baesman, 2000; Davies & Stock, 1996], it became clear that the quality of support for a given step of a task script depended on the image presented to the client. The image displayed on the 2.75- × 4-inch screen of the handheld computer could strongly affect the success of the attempted task. The “best practice” rule that assistive technologists use states “for a successful match, the image fidelity has an inverse relation to the level of cognitive ability” (i.e., the less ability, the higher the needed fidelity of the image; [Snell, 1987]).

Table 6 – *Image Type Experiment*

Participants	Method / Procedure	Data Collected	Result	Interpretation of the Results
15 high school young adults with cognitive disabilities and 15 age-matched high school typicals	Cue image was presented on PDA at same time as three life-sized images were presented on a screen. Participants choose a match with a 10-inch diameter button on table in front of screen. Computer stored results.	Counted accuracy (exact match, close match and wrong) and time from initial image to button push	Data accuracy and reaction time with respect to identifying an object from a small image was tabulated	There were three possible trigger images on the PDA: icon, photo alone, and photo in context. All participants did significantly worse with icons as triggers. The young adults with cognitive disabilities took a significantly longer time to make a match than typicals

A literature review found studies about profoundly retarded individuals and image recognition [Dixon, 1981; Stephenson & Linfoot, 1996] and visual recognition issues pertinent to persons with mild cognitive disability learning reading and recognizing letters. In collaboration with Erica Wohldman, a doctoral candidate in cognitive psychology, an experiment was devised to test how the type of representation (icons, photos of objects in context, photos of objects in isolation) displayed on a

handheld computer affected recognition performance in cognitively disabled young adults (see **Table 6**). Subjects were required to match an object displayed on the computer to one of three pictures projected onto a screen. The experiments tested the opinion widely held by occupational therapists and special education professionals that there is an inverse relationship between cognitive ability and the required fidelity of a representation for a successful match between a representation and an external object. The subjects, selected on the basis of being cognitively disabled, fell into the range of potential MAPS users with no confounding problems with vision. As part of the experiment, the same number (15) of “typicals”—age-matched cognitively normal students—were given the same test. Subjects were required to match an object displayed on the computer to one of three pictures projected onto a screen. The image on the handheld randomly varied between an icon, a photograph of the target in isolation and one in context (e.g., on similar shelves in a grocery store). Despite the widespread use of icons in most learning tools developed for the cognitively disabled, our results suggest that icons are poor substitutes for realistic representations, but among the other representations there was not enough statistical variation to draw valid conclusions. A discussion of the experiment has been submitted to the *American Journal on Mental Retardation* [Carmien & Wohldman, 2006].

## **6.6. Realistic Field Trials**

Because MAPS might be used in various milieus, MAPS was tested by representatives of three groups: young adults living with their parents while still in school, young adults enrolled in a transition program in their local school system (typically 18 or 19 years old), and adults living in group homes with varying levels of caregiver support. This part of the project used ethnographic methods, particularly participant observation and semi-structured interviewing [Emerson et al., 1995; LeCompte & Schensul, 1999]. The goal of the participant observations was to discover and learn about the client and caregiver’s world and their interactions as well as how tasks and learning of tasks were currently conducted (see Table 7). The initial visits with the client were both with and without the caregiver present, designed to gain insight into the general lives of the participants and more specifically about the client’s world with respect to abilities, relationships, learning style, and employment. This consisted of about six hours of contact for each pair.

When the initial participant observation of daily life and employment was concluded, we moved on to watch the process of the caregiver learning (and being taught) how to choose and segment tasks, gather the material for prompts, assemble scripts, and load them onto the prompter. Finally, we investigated how the MAPS-prompter supported learning and doing a new task. For each dyad, this process of designing a script, generating its image and sound components, assembling it, and using it to guide the user through a new task, was repeated three times. Field notes and recordings captured the detail in these sessions.

The process of the caregiver learning about scripts and the MAPS system consisted of starting with a session about script building and task segmentation. This was aided by the use of a video that illustrated the skills needed to properly segment a task and translate it into a series of prompts for people with cognitive disabilities [The Oregon Research Institute, 1988] and a checklist of topics to be covered (see appendix 13.5). From the video and the checklist, the caregiver and the researcher brainstormed to create an idea for the first script and what the steps should be. That done, the study moved onto designing and assembling three scripts.

**Table 7 - Realistic Field Trials**

Participants	Method / Procedure	Data Collected	Result	Interpretation of the Results
BVSD Families (two dyads)	Give dyads MAPS script-design-environment and MAPS prompter. Observe process of use of system	Audio recordings, field notes, secondary artifacts	For all four dyads, the process of adoption of the system was documented and analyzed	Generated heuristics for script designing and use, successfully made script-design-environment easy to use by caregiver, but uncovered OS issues. Understood internal/external script inventory effects on task success.
Imagine! Group Homes (one dyad)				
BVSD Transition Team (one dyad)				

For each dyad, the intent was to make and test an increasingly challenging set of scripts. The content and environment of the scripts was typically from simplest to most complex (see Table 8):

- Controlled environment (e.g., a housekeeping chore), in which neither the task nor the environment is dynamic and the environment is familiar;
- Less controlled script (e.g., cooking), in which the task doesn't change and the environment is dynamic but familiar;

- Least controlled script (e.g., shopping), in which the task and the environment are unfamiliar and the environment changes

The first script was also used to familiarize the caregiver with the script design and composition process and to familiarize the person with cognitive disabilities with the use of the prompter, its controls, and how to follow a script. The performances of each of these scripts were observed and recorded. Analysis of the usage of each script allowed the caregiver to refine the design of the subsequent scripts.

**Table 8: Realistic Field Testing**

	Scripts	
	Objective	Typical
<b>Controlled</b>	<ul style="list-style-type: none"> <li>• Learning the process of writing and creating a script (caregiver)</li> <li>• Learning how to use the prompter and how to follow a script</li> </ul>	<ul style="list-style-type: none"> <li>• Sweeping the floor</li> <li>• Washing and drying a load of laundry</li> <li>• Folding dried clothes and putting them away</li> </ul>
<b>Semi-controlled</b>	<ul style="list-style-type: none"> <li>• Attempting to do a task for which the environment might change and timing issues are critical</li> </ul>	<ul style="list-style-type: none"> <li>• Cooking</li> </ul>
<b>Uncontrolled (i.e. out in the world)</b>	<ul style="list-style-type: none"> <li>• Using the prompter in situations where: <ul style="list-style-type: none"> <li>• Other people may interact with the client</li> <li>• The environment may change</li> <li>• Task requires multiscripting</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Cooking in a unfamiliar kitchen</li> <li>• Going to the library and checking out materials</li> <li>• Shopping for groceries</li> <li>• Performing complex employment tasks</li> </ul>

The process of designing and using a script typically started with a caregiver coming up with an idea for an appropriately sized and relevant task. After discussion with the researcher, the caregiver went back to formally breaking the task into steps and filling the content of each prompt in a script design text template (see appendix 13.6). When the task was segmented and descriptions of images and verbiage of prompts were chosen and entered into the script design template, the caregiver submitted the filled-in script to the researcher and the AT expert for comments. In most cases, this review was done only for the first script; the others were done by the caregiver with no outside assistance. Of the

four caregivers, one had no help, two received assistance and suggested changes for only the first script, and one received suggested changes for the first two scripts. With a list of images to capture and words to record, the caregivers took pictures and accomplished the recording. In one case, the gender of the voice on the prompts was judged by the caregiver to be critical to the success of the script due to power issues between the teenaged daughter and her mother, so the recording was done by a male voice from outside the family. For a majority of the script making, the caregivers had to submit the image and verbal binaries to be processed so that they would be below the 64-kilobyte (Kb) limit for blobs<sup>13</sup> given by the Sybase Ultralite database on the handheld. This limitation was a big issue with the caregivers because they all wanted to be in charge of the whole design and creation process; the constraint was removed in later releases of the MAPS suite. The last two caregivers were able to do the whole process without adjustments to their pictures or recordings. After the script was input into the script-design-environment, it was uploaded to the prompter and ready for use.

When the person with cognitive disabilities was ready and the situation was right (e.g., laundry was dry for the laundry-folding script, or it was late afternoon for the cooking dinner script), the preloaded hand prompter was given to the client. Typically, the observer/researcher would be off to the side with a notebook, and a digital recorder would be placed midway between the client and the observer, then the observer would ask the client to start the MAPS software. One exception to this setup involved the shopping script, where another observer was posted in the supermarket with a copy of the script template and a notebook to take notes and make sure that the client did not get dangerously stuck on a task. After the client got to the end of the script, pictures were taken of any support artifacts used in the script, and a quick debriefing of the caregiver was recorded.

The following subsections describe the participants on the field trials in some detail, but at a much broader level of detail than was noted in the field notes and field recordings. Each dyad was observed, during the various stages, for between 15 and 22 hours. In discussing the highlights of each

---

<sup>13</sup> **Binary Large OBjects** – a data type in databases, in contrast to strings or numbers.

pair, the intent is to give the reader some sense of the world of the cognitively disabled and also the details of incidents that will be discussed in aggregate later in this chapter.

### 6.6.1. Families

The two dyads who represented families with a parent acting in the caregiver role and a teenage young adult with cognitive disabilities were recruited through Anja Kintsch and her contacts in the BVSD. Each family went to an initial meeting at which the MAPS system was presented and expectations for the scope of the experimental trial were explained. The meetings were held at the L<sup>3</sup>D Lab of the University of Colorado, Boulder, and included the family, the young adult with cognitive disabilities, the researcher, and an AT professional. One of the purposes of the initial meeting was to informally examine the young adult with cognitive disabilities and the caregiver parent with respect to their suitability for the field trial.

The first family in the study consisted of a primary caregiver/housekeeper mother, the husband, an older daughter away at college that winter, a 16-year-old daughter diagnosed with “pervasive developmental disorder not otherwise specified (PDDNOS)”<sup>14</sup> with a *WAIS* [Wechsler, 1997] IQ of approximately 69, and a 13-year-old brother. The daughter was diagnosed with autism at an early age. The mother was able, due to strong motivation and being a stay-at-home-mom, to be a strong and effective advocate for her daughter’s situation for both adaptation and rehabilitation goals. The daughter was enrolled in special education classes in the local high school. She held a job working once a week in the kitchen in a local hospital, where she and two other young adults were supervised by one job coach. The client had a third-grade reading ability and fourth-grade math ability. She played violin in the middle school orchestra, even though she was in high school, and during the trials she moved up to the high school orchestra. Her mother had set up a sophisticated and full tutoring schedule for her daughter, using consultants, university students, and several different home schooling type programs.

---

<sup>14</sup> The mother explained that this is really a catchall diagnosis.



The mom was able to send and receive email and became competent at shooting photos and transferring them to the MAPS environment. Similarly, she had no problems making recordings (following microphone setup and learning the MS recording application). However, she chose to have the prompts recorded by a neutral (out of the family) male voice. Because this was the first of the dyads to do the field trials, a fair amount of time was spent fixing small bugs. Notwithstanding that, the mother was able to use the script-design-environment to easily make three scripts. She said that the script-design-environment was easy to use; however, she felt the system did not provide enough help in loading the prompting script into the handheld computer. Consistent across all the dyads was the fact that the actual script-design-environment was easy to use—especially so with the supplied video help—but that the loading of a script to the handheld and the storage of the images and sounds for the script-design-environment were frustrating at first.

The first script chosen and designed by the mother was a simple script helping her daughter to properly sweep the kitchen floor. The mother also added a section of prompts to the beginning of each of her task scripts that explained the use of the hand prompter controls. This “generic” sub-script of six steps was reused by two of the other dyads—an example of script reuse. Task segmentation added to this generic preface sub-script resulted in a 28-step script. Taking the pictures and recording the prompts was not a problem. The intermediate document that the mother used to prepare the script became the blank script template that was subsequently provided to each of the caregivers in prompting training. With the use of the sweeping script, the first scripting heuristic was discovered by noticing its absence: always have someone other than the script author physically do the task with only the script in the task environment. In this case, the script at one step left the daughter holding the broom while being asked to pick up something else.

The next script was cooking an entree for the family dinner. The mother easily accumulated the images and verbal prompts and created the script of 35 steps. This time the mom decided to stay away from the kitchen because she might disrupt her daughter’s focus by being there. The daughter was able to make the “mini meat mounds”; however, one of the script steps asked her to use a utensil that was not taken out earlier like all the others, and another step directed her to use an ingredient that was not brought out at the beginning like all the others. These shortcomings demonstrated two things: (1)

novice script writers need to have a dry run of the script before giving it to a person with cognitive disabilities, and (2) in this case, the caregiver may have underestimated the skills and existing abilities or internal scripts (see section 7.2) her daughter already possessed.

For the last script, the mother decided that her daughter was capable of going shopping for groceries by herself with the prompter. Mom put together a shopping list of eight items, and walked through the local supermarket (the one that she and her daughter have been going to for several years), walking the path that she wanted her daughter to take and taking pictures of way points (e.g., “Next go to aisle 12” combined with a picture of the aisle 12 sign), target items (e.g., “Get a box of tissues” combined with a picture of the tissues), and checkout procedure (e.g., “Give the checker the money” combined with a picture of a hand coming out to the checker they usually went to). The resultant script was 25 steps long.

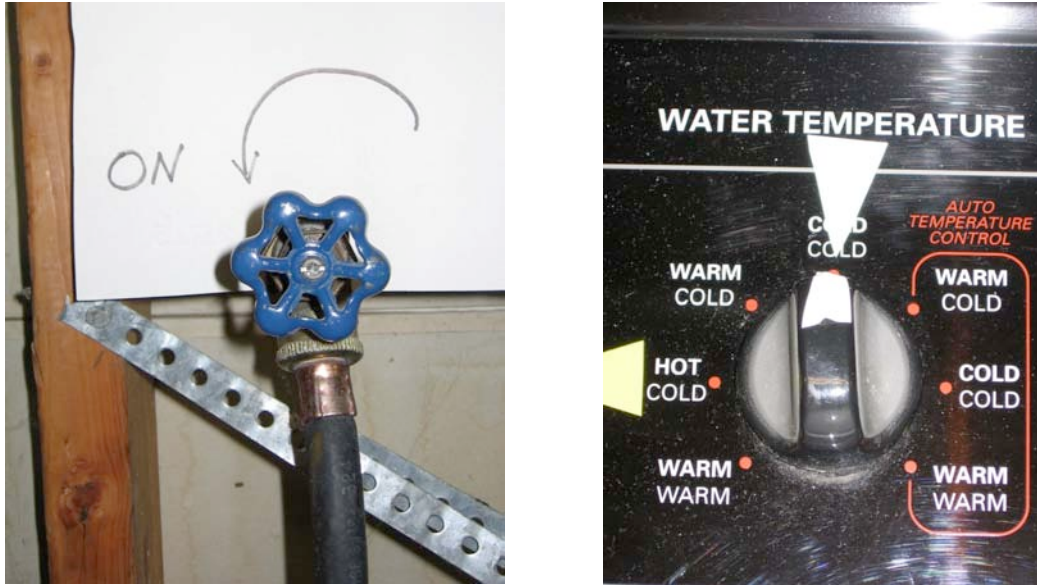
On the afternoon of the test, the client was driven to the store, given the handheld prompter, \$60.00, and a Safeway customer card. Previous to our arrival, a colleague who was not familiar to the client was stationed in the store with a cart and a copy of the shopping list that was in the script. The researcher and the colleague kept in touch by cell phone. The daughter went into the store at 3:29 and came out at 3:45 (from the script log), with every item on the list, all correct except for getting whole milk instead of low-fat milk. The mother said that her daughter would not have been able to do the same task with a paper shopping list without training and repetition. The colleague reported hearing the checker say to the client, “*Oh, you’re shopping by yourself now?*” In the exit interview (see appendix 13.7), the mother said that she was interested in continuing the use of MAPS with her daughter, especially if the requirement of shrinking all binary files to the 64 KB limit was removed, which was accomplished by mid-experiment.

The second dyad drawn from the families was a young adult with cognitive disabilities who was attending special education classes at BVSD. This family consisted of mom, a part-time programmer and full-time mother; dad, a scientist with a local federal scientific organization; and a 19-year-old daughter with developmental and hearing disabilities. The daughter presented as having limited communication ability: she could sigh and grunt to indicate yes or no, and she was frail and friendly.

She communicated with American Sign Language (ASL) with her mother and teachers. She would from time to time get caught in a loop of behavior, often with her hands, and these soothing self-hypnotizing perseverating actions caused problems with tasks that needed a focused flow to accomplish them. Mom had no problems with designing the three scripts, after her exposure to the task segmentation video and exercises.

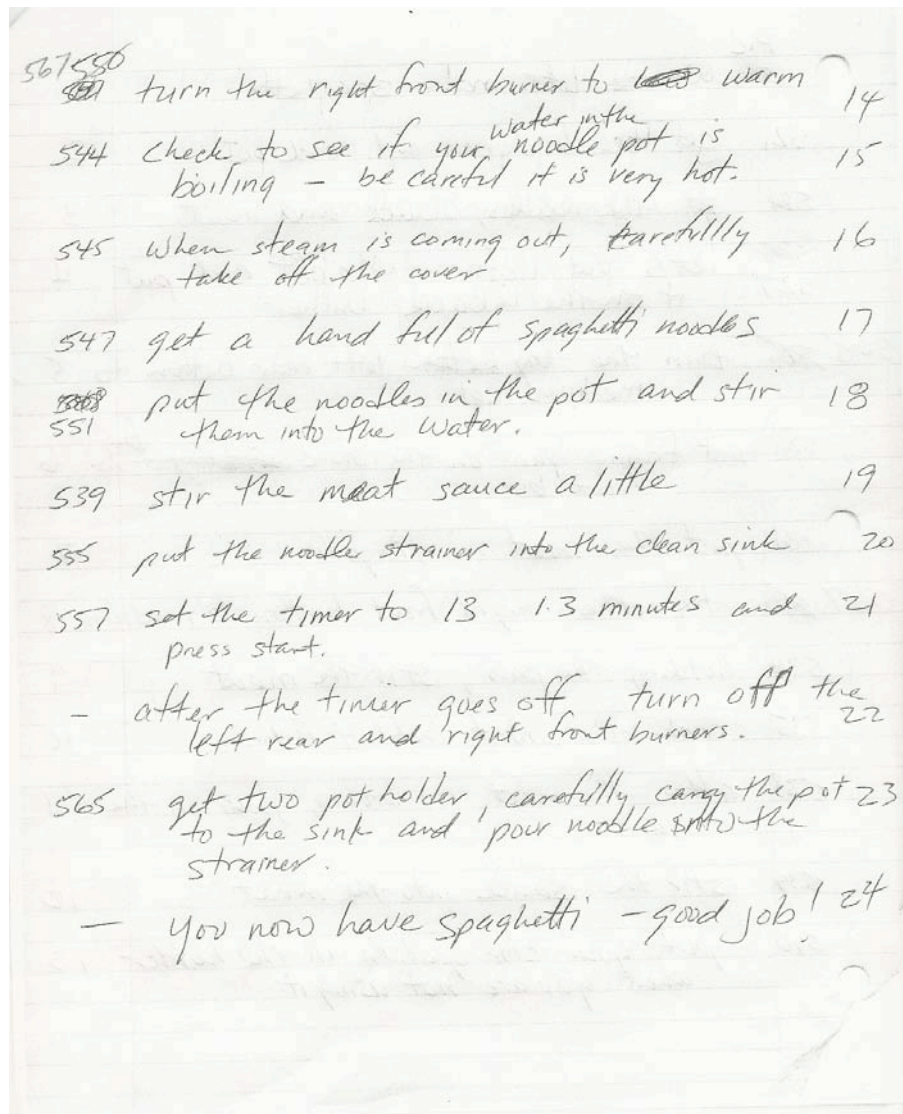
Although the goal for participant selection was for the caregivers to have minimal computer skills, as was the case for the script-design-environment usability/participatory design process, the paucity of potential participants for this part of the research led to selection of this dyad. Interestingly, of the four participating dyads, this pair had the most difficulties using MAPS. In the end, this pair was the one that most probably would not continue using the system after the studies, although they requested continued release support of MAPS and expressed interest in continuing use of MAPS.

The first script was to do a load of the family's laundry, initially both the wash and the dry cycles as part of one script. It became obvious, however, that there was no easy way to keep the daughter on task during the time between the start of the wash cycle and the end of the wash, when she should start the drying task, so the tasks were broken into two scripts. The resultant scripts contained 15 steps (including "generic" control use instructions) for washing and 7 steps for drying. Additionally, after a bit of brainstorming, it was decided to put tape markers on the controls to change the task from "turn the temperature control to hot water" to "turn the big knob on the washer so that the blue marks line up," illustrating the way that distributed cognition transforms tasks and uses alternative representations of pertinent knowledge (see Figure 34). This laundry script helped the client properly wash and dry a load of laundry, but exposed several problems. First the client, once started on a part of the task that she knew, continued on past the stopping point of the prompt and thus ended up needing intervention twice to keep her on track. Also, it became clear that parental involvement in the actual performing of the script was not a good idea. Having spent many years tenderly helping their child with many needs, it was difficult for the mom to stand aside and watch her daughter get stuck on a step.



**Figure 34 - Distributed Cognition Artifacts in the Environment of a Task**

The second script was cooking a simple dish, spaghetti. The creation of the script was again an easy task; however, the mother objected to naming the wav files of the verbal prompts with the actual words in the verbal prompt (a file with the recording saying, “Now get the spatula and put it next to the pot” would be named “Now get the spatula and put it next to the pot.wav”). Instead, she named them sp#.wav, and as a result, she had to use a supplementary handwritten index to put the right sound with the right picture (see Figure 35). The intent in requiring the wav file name be significant is that, unlike images, which can be browsed via thumbnails with several being viewed at the same time, wav files cannot. Additionally, they provide (with some anonymity required) the basis for sharing the scripts because the words can be used as generic prompts.



**Figure 35 - Secondary Artifact for Script Assembling in the Script-Design-Environment**

The actual use of the script demonstrated the need to augment the environment with markings on the stove similar to those used on the laundry script. The need for this adjustment was made clearer when the mom expressed concern that her daughter may not be able to cook unsupervised due to the danger of the daughter turning on the wrong burner, not turning off the burner, or, due to her disabilities, not noticing that she had rested her hand on a hot burner until there was damage done. “I think that the microwave will be her cooking tool rather than the stove” was how the mom expressed her concern at the end of all the experiments. The mom intervened several times to help her daughter get back on track, but in the end the dish was finished. The problems in using the script were exacerbated by the client’s tendency to double click the MAPS prompt player, due to her motoric

disability. As a result, a modification of the MAPS prompt-playing software was made that allowed a user (caregiver)-settable delay after a control was pressed before any control could be used again. The delay was set, after some experimentation, at 2000 milliseconds for this client, and 750 milliseconds for the other users.

The final script for this dyad was another cooking script, this time done in an unfamiliar kitchen without any family present. Mom made a script of the steps needed to make a Hamburger Helper® main dish, with little trouble, especially because the need to shrink the sizes of the wav and jpg files for use in the Ultralite database was removed; this allowed the mom to do the whole process from idea to script creation on her own. This script was performed by the daughter with the same semi-successful results. Because of the anti-double click modification, the daughter did not get confused by out-of-synch prompts and have to backtrack. There was some confusion about the two measuring cups used, resulting in a dish that had three times as much milk than was called for. This underlined the need to annotate the tools in the environment so that the instruction that read “get the four cup measuring cup” would say “get the measuring cup with the blue stripe,” and thus have less detailed understanding of measurements replaced by a simple color selection task. The meal was cooked, but it did taste watery.

The one insight in working with this second dyad is that even though the client was selected by the BVSD professionals, she was a little too low functioning for the prompter system (see Figure 7 and section 2.1.3) to make the big difference in her life that was desired. The client did use the laundry script to learn to do laundry, however, and did the family’s laundry several times on her own after the experiment.

### **6.6.2. Imagine!: A Group Home**

Another living situation that might benefit from the use of MAPS is that of group homes. The Imagine! [Imagine! 2004] organization provides, among other services for persons with cognitive disabilities, several group homes in the Boulder Valley. During the development of MAPS, presentations of the system were made to their residence and technology managers, and they expressed interest in supplying dyads for field testing. After interviews of several likely participants, a dyad of a professional caregiver and a 37-year-old man was selected to participate in realistic field tests of

MAPS. The client had a WAIS-R [Wechsler, 1997] IQ score of 57-63 for the full-scale range. He had lived in the group home for 14 years, and visited his parents approximately every other weekend. The group home had three apartments and a caregiver's office. Each apartment was home to two persons with cognitive disabilities. The caregiver was on site in the morning and in the afternoon, helping the residents with scheduling, getting to appointments, taking medications, and participating in recreation activities.

The client worked at a gymnasium five days a week from 10:30 a.m. until 4:00 p.m. His jobs included caring for the men's changing room and doing other janitorial tasks. He was able to take the bus to work by himself, yet he was consistently unable to do some household tasks, such as folding and storing his clothes properly after doing the laundry. Each member of the group home cooked for the group as a whole every other week or so. To get acquainted with his world, the researcher spent several afternoons with the client in his daily life, as had been done with the other field-trial participants. The setting was an exercise club where the client was a real member of the workplace community, having worked there for many years, and had many friends among both the staff and the members. Following the visit to the exercise club, the client and researcher took a trip to the Boulder main public library and checked out videos and CDs. Later that week, the client was observed while cooking a meal for the other members of the group home.

The caregiver had been taking simple recipes and turning them into scripts for the client to follow for some time, so the introduction of MAPS and the need to properly segment tasks was an easy transition for the caregiver in that respect. She proposed that for the first, easiest script that she make a script to guide him through folding and putting away his laundry. The resultant 61-step script walked him through removing the laundry from the hamper to hanging his dress shirts on hangers. One problem that the caregiver had with creating the script—and one that other caregivers also had—was confusion over how to start up the MAPS application and the locations of the directories where the recorded prompts and pictures went. Eventually, shortcuts were created for these directories, and these should be part of the installation of the application. She took the pictures, recorded the prompts, and created the script, relying on the video help to guide her through the difficult parts. The 61-step script was downloaded to the handheld computer, and the client attempted to run the script. He had failed in

his first pass at the task due to problems with the current release of the MAPS scripter software, which was now repaired, and the timing of the anti-double-click function was correctly reset.

By reloading the application and script, the client was able to do the task. In several instances, he did not stop when the instructions for a given prompt were completed, so the next several times that the folding script was run (he did it once a week), the caregiver focused on that part of using the MAPS-scripted task support. Each time the script was run, the client improved his performance, so by the fourth week the clothes were being folded and stored in a way that was acceptable to the caregiver. The initial notion of the use of the handheld prompter was that the user would be taught how to use the prompter, and perhaps guided through the first use of it with a script, but subsequent uses would be done solo. This caregiver, however, had a different idea of use. She approached the task of learning the use of the handheld PDA *and* the script running on it as being scaffolded by her, the caregiver. As a result, for the first three runs of the script she stayed in the room with him and intervened when she thought it was needed. By the fourth and subsequent runs, she left him in the room alone, and his performance continued to improve. This demonstrates a use of the MAPS system as a training tool, which was not expected at the original design and imagined use.

One of the advantages of the IPAQ/WIN-CE platform of the hand prompter was its ability to be used as an MP3 player. When the user was given the prompter, he was asked what kind of music he liked, and songs by the artists that he liked were loaded into the spare memory of the prompter. He was quite happy listening to this music, and it became part of the laundry ritual to fold the laundry and listen to music on his headphones. There was concern about whether the volume of the built-in speaker of the IPAQ 3850 would be loud enough, especially in outside environments, because this client had some hearing problems. Although he heard the prompts well, there were a couple of exceptions. For the folding script, he would lay the prompter on the bed as he folded the clothes, and for two of the steps he had to lean his head close to the prompter's speaker to make out the prompt. At several points in the script, he would talk to the hand prompter (whose voice was his caregiver) as if the caregiver herself were in the room.



After the first two runs of the folding script, the caregiver decided that some of the sections of the script were too long and that several of the steps were not sufficiently illustrated to ensure success. Fortunately, in the initial design of the script she had made this easy by inserting “collapse points” (see appendix 13.14) so that the steps directly below each collapse point could be removed and the word and prompts of the collapse points would be a sufficient external script trigger for the client to do that part of the whole task. She removed these superfluous sub-scripts in the case where the instructions for folding T-shirts was repeated because he had learned the process to some degree. She also redid several of the steps for clarity and flow. When she did this, however, she decided to stop the laborious renaming of the files with the words from the recordings, and used meaningless numbers instead. When she was made aware of the problems this would cause for her if she wanted to further modify the script and also that doing this made using the script impossible “as is” for a template, she changed back to the more labor-intensive file-naming scheme. This problem of way file names remains to be solved to the satisfaction of all stakeholders. This problem may be due to the dilemma of shared computer-supported cooperative work (CSCW) artifacts (“*who does the work, who gets the benefit?*” [Grudin, 1987]) in that the script-maker gets no direct and immediate benefit for renaming the files; the value of the annoying process of typing in very long filenames only becomes apparent if they rewrite the script themselves or see their scripts reused as templates.

The new script had 51 steps (compared to the previous 61). In reviewing the scriptwriting process, the caregiver said that while making the script, she took five pictures for each step to get the one she wanted, and similarly, she recoded each prompt about five times until she got the one she wanted. This, however, was the extreme end of the effort required to make a script amongst the caregiver population in this study. The client was pleased with the script, the MAPS-prompter, and use of the MP3 player, and had discovered a solitaire game on the PDA by himself. He said to the other staff members that he had “learned how to fold.” His success and the need it was built on are good examples of the “islands of deficits in seas of abilities” (see section 2.1.3); it might seem, to the casual observer that the client might not need task support for something as simple as folding his clothes, but this was a task that he had not accomplished in many years of trying.

The second script that the caregiver designed and constructed was a script to help the client take a trip to the local library. This branch was located within walking distance of his house and the trip was suggested because the client enjoyed going to the library, based on his reaction to initial visits when he got a library card. The caregiver took a camera with her and walked the same path to the library as the script would lead her client. After getting the images, she wrote the script steps and then the wording of the prompts. She then recorded the prompts and assembled the script. The following day, the client and researcher walked to the library, with the client “calling the shots.” Halfway to the library, it began to rain quite hard, at which point the caregiver came by car to drive the rest of the way to the library with the client as navigator. At the library, the client needed prompting for only one of the three goals, the CD rack. He already knew where the DVDs and the rack with the kinds of books he liked were located. The trip was successful, but two things stood out as a result: (1) just as the cooking script aimed too high for the second of the family dyads, this script had aimed too low for this fellow; and (2) a detailed prompt that was wrong would have led the client to be unable to continue at a certain point (the sidewalk did not continue at an intersection).

The last script for this dyad was chosen, designed, and implemented by the client’s parents. For many years they had tried to teach him how to properly hang his nice clothes on a hanger after church. They would show him and prompt him through the process and he would learn, but when he returned the next week, that knowledge was lost. Their hope was to put it into the prompter and use it to bridge the time span while he was at the group home.

The contrast between the too easy trip to the library and the difficult folding and hanging tasks is an excellent demonstration of the variability of skills of persons with cognitive disabilities observed and commented in the literature [Cole, 2006; Snell, 1987]. Other important points presented by this dyad’s experience with MAPS were:

- The MP3 player was a good attractant to involve the clients with the prompting process.
- The client went through several of the pointers that were necessary to use with the IPAQs. The pointers were cheap to replace, but a nuisance to buy.
- The client was positive about the folding script; he told several other staff members of his success.

Because the relationship of one caregiver to many clients and the long-term history of the clients (of the six in this group home, four had been living there for more than 10 years), the MAPS system may be of particular value to the adults with cognitive disabilities and support communities. It will be interesting to see whether the system is more widely adopted in this home and in the other homes run by Imagine!

### **6.6.3. BVSD Transition Team**

Transition programs in local high schools are another milieu that was ranked high in the list of environments that MAPS might successfully provide support for greater independence. In the state of Colorado, the school system provides support for learning life skills and general education until a young adult with cognitive disabilities turns 21, at which point the support must switch to state and federal programs, which are not as generous and focused on transitioning the young adults toward independence. Therefore, the BVSD has a transition program called “Bridges” with the following mission:

This program focuses on the domains of Career/Employment, Community Involvement, Daily Living Skills/Residential, Social/Interpersonal, and Recreation/Leisure. Students entering the Bridges Program must have enough credits to graduate, but must not have received their diploma. Motivation to succeed in adult life is an essential ingredient for success in this program, along with family/parent support for independence. [Boulder Valley School District, 2006]

Bridges provides job support in obtaining employment and in the path to unsupported employment. This can be critical for the young adult with cognitive disabilities about to make the transition from family to another sort of separate living situation. Working with the administration of this project, MAPS was presented early in the test process, and when the time came to run realistic field trials, the staff of Bridges was ready with a list of potential test participants.

The young lady chosen for the young adult with cognitive disabilities role in the dyad was 19 years old. She was diagnosed as having multiple cognitive disabilities, Microencephaly, SLIC (severely limited intellectual capacity) with a WAIS IQ of 54 WAIS<sub>fs</sub>, also with secondary speech and language deficiencies. As an illustration of the skill variability that many of the young adults with

cognitive disabilities have, this client has an amazing, professional-quality singing voice. She was currently involved in supported employment at a local used clothing store. She worked two and one half hours at a shift twice a week, typical of training time. The client's job was, at first, to colorize (put into a specific color order) a rack of clothes (the racks are separated by gender and type and by size and color). Her job coach typically would be standing next to or near her during her shifts.

At the first observation of the client at work, there were two job coaches (one was a supervisor) and the client. The job coaches' task was to monitor the client doing the job and intervene as needed. At the first observation session, the job coach was observing the client with an increasingly large list of notes and evaluations about her and this job on her clipboard. Another job coach at this session shared that it is difficult to evaluate this population and employment because "tests don't tell us so much about" them in real situations. The original job coach said she was writing down what the client was doing and counting how many things she was doing, so they would have a baseline, and noticing what she was doing correctly, such as separating out dirty clothes to clean, etc.

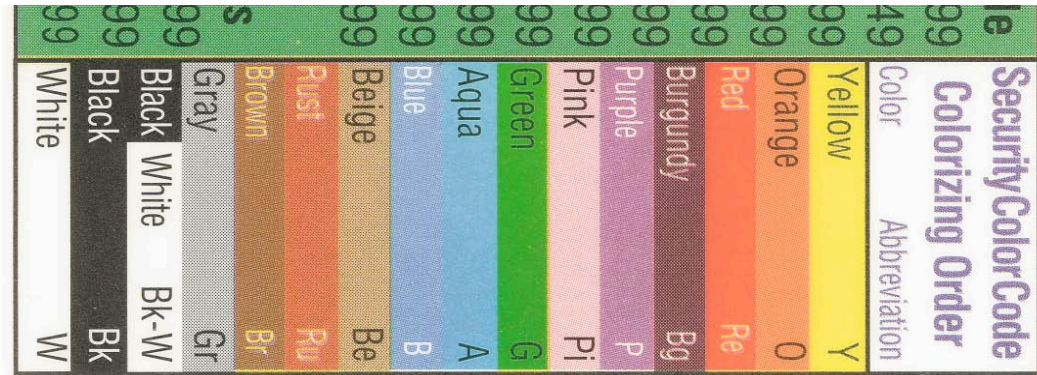
The job coach discussed the two kinds of skills needed to successfully hold onto a job: soft skills [Gilberts et al., 2001] and hard skills. Soft skills refer to the social aspects of employment, such as greeting fellow workers and supervisors, or knowing what to do when the current task has been completed. Hard skills refer to the specifics of the employment task: counting, following a sequence of tasks, sorting, etc. The job coach indicated on the sheet the soft skills and the hard skills identified. The job coach pointed out that it was important to have the ability to come up with another task to do when the client was done with her tasks and she couldn't find her supervisor. Lacking this skill will "*deep six our people*" if they just stand there after completing a task. They need to be able to find something to do until they see their supervisor. The job coach supervisor said, "*90 percent of this is a soft-skill.*" In looking at what MAPS could provide to the situation, she felt that it would be difficult to put soft skills into a script. The other job coach said that the only thing to do with respect to soft skills is to "*catch it and repair it*" on the spot.

The job coach saw potential in supporting and helping to train her client in the soft skills. Supporting soft skills as well as the more easily implemented hard skills in the MAPS script required

modification of the MAPS-prompter software. First, the changes that might support alternate tasks had to be designed. Several ideas were made into paper prototypes, and eventually the layout and semantics of the proposed solution were chosen. The basics of the solution were that the prompter would display three possible scripts as icons in the upper corners and at the middle of the bottom of the display. The script would start by running the “main” script, which was represented by the lower icon. Pressing on any icon would start that script (see Figure 24). By carefully designing the contents of the prompts in a script, a caregiver could implement looping (coming to the end of a given script and re-pushing the icon for that script restarts the script) and forking (coming to a prompt step that implements forking directs the client to “press the upper right picture if you want to XXX and the upper left if you want to XXX”). This was a difficult skill to learn for a caregiver.

The first script (or really, set of scripts) that the job coach wanted to write was for the primary task of “colorizing” the racks, plus two supplementary tasks from which the client could choose at the end of colorizing: removing empty hangers and “ragging” the shelves. The term “ragging” refers to either cleaning up and organizing an area, or pulling clothes that have been on the shelves too long and thus are ready to be sold as rags or given away (the second meaning was probably extended to the first meaning over time). The used clothes store is in a former supermarket, and the clothes are segregated by gender and style (i.e., men’s and women’s sections, with the women’s section then grouped by type, such as all slacks, all blouses, etc.) .Within the type, the clothes are further sorted by size (small, medium, etc.), and within each size, they are sorted by color (a very specific color scheme; see Figure 36). The colorization task requires a multi-pass procedure: the first pass is to remove empty hangers; the next is to pull the clothes that are not of the same gender, type, and size as the section being colorized, and the next is going through the homogeneous rack to reorder the items until they are in the correct order.

The other scripts guided the user through removing all the empty hangers in a rack, and going through all the items displayed on the shelves above a rack and ensuring that they were all of the same kind and all evenly laid out on the shelf (see appendix 13.12).



**Figure 36 - Clothes Store Color Order**

The job coach, being a work-support professional, understood well how to do task analysis and segmentation, so instead of the instruction that had been given to the other caregivers, she was shown how to do looping and forking in the new scripter configuration. After several designs and iterations on the techniques of implementing looping and forking, the job coach understood such ideas as loop invariants and if-then statements without having to explicate and present the details of the abstraction about how to do so. The proof of her new skills was that she was able to produce the second script with no help in designing the conditional prompt steps.

Having designed the set of three scripts, the caregiver had to produce images, which was easy for her. She was reminded to take the photos sideways due to the aspect ratio on the handheld screen. Again, there were problems with setting up the directories for the binary files, which was corrected by the use of alias directories on the desktop. The caregiver had problems recording the prompts until she got set up with a microphone at the PC in the Transitions office. The need to support forking and looping caused a substantial amount of recoding of the code running on the handheld, which required a fair amount of debugging. In the end, getting the first set of scripts loaded in the correct order and connected to the appropriate icons required designer intervention.

With the scripts loaded on the MAPS-prompter, the caregiver introduced the prompting system to the client. The client understood the mechanism of the system and proceeded to go through the script sequence, color by color. She had to be given instructions in the navigation of the controls for the prompter each time, and in the end the buttons were labeled with arrows (← for one step backward;

and → for one step forward), and the name of her job coach caregiver was placed over the middle button (as the voice of the prompts meant a lot to the client), which replayed the last verbal instruction. After a few minutes of use, it became apparent that the client could not hold the prompter and do the task at the same time, and she was alternating between holding the prompter and looking at the prompt and (having to put the prompter on a shelf) doing that part of the task just described. For the next session, she was provided with a nylon holder for the PDA with a transparent face (see Figure 37), flexible enough to accept input on the screen (although the on/off button was difficult to access) and sufficiently transparent to audio so that she could still hear the recorded voice of her caregiver.



**Figure 37 - ChatPC II Carrying Case**

The dyad was observed using the script two times, although the script was used for a month before moving on to the next script. The client's use of the prompter with the colorizing script successfully reduced the number of times that the job coach would have intervened. Additionally, after the client went on vacation with her family during spring break, the caregiver said that she resumed her job faster than she would have without the computationally enhanced task support.

Because the client was a gifted singer, the prompter had 200 songs preloaded on it. The caregiver said that the client cared for the handheld very carefully, monitoring the battery usage and often carrying it with her. She now had a device that the other peers would not see as "dorky," but as

positively “cool,” and thus there was a higher probability that she would not damage or lose it [King, 1999]. In one of the observation sessions of her second script, one of the customers in the store came up to the client and told her what a neat player she had.

During one of the observation sessions at the client’s workplace, the client’s mom and grandmother came by to see how things were going, and her grandmother told a story about not understanding how to use the family’s home theater, and the client (her granddaughter) showed her how to do this, illustrating the fact that although these young adults with cognitive disabilities may be cognitively limited (this client’s IQ was 54), their integration into twenty-first century household appliances was common and pervasive.

Having successfully integrated the first set of scripts into her work routine, the client received a second script designed by the caregiver for the next task she was to be assigned—taking care of the dressing rooms. The store had five changing booths in the west end of the building. Her tasks were (1) to make sure that customers did not take more than six items at a time into the booths, (2) to remove any of the empty hangers left in the booths, and (3) to take all clothes that had been left in the booths to be restocked on the racks.

After the client used this script for a week, the management of the store offered her a position with pay. The job coach agreed that the client was ready to leave supported employment. The coach found this unusual in that they offered her client the job after a shorter time than she thought it would have taken without the prompter. The coach also found it unusual that they offered so many hours at the start; she was offered 25 hours per week employment, when typically “*our kids get offered 5 hours on their own or 20 hours subsidized with support.*” She also felt the voice on the prompts would be sufficient support for her client so that she could rely on that rather than needing a job coach with her all the time. The client was excited but also apprehensive to move into nonsupported employment. On the last scheduled day of supported work, the client asked her job coach many “what if” questions; the answers to many of these questions was “listen to your prompter.” Afterwards, the coach emphasized that she was comfortable with the client moving on because she would have her voice to comfort her



and give her direction. From this observation emerged the notion that, for many clients, a prompt consisted of not two but three components: the image, the verbiage, and the voice of the prompt.

**Table 9 - Summary of MAPS Testing**

Trial	Participants	Method / Procedure	Data Analysis	Result	Interpretation of the Results
Initial PDA field trial	8 BVSD Special Ed students	Demo of prompting and using a prompter to assemble glider model	Observation, semi-structured interview, audio recording, script log	Seven succeeded, one succeeded with help	Data led us to the conclusion that young adults could successfully use the MAPS prompter
Usability / participatory design of MAPS design environment	9 people, 3 iterations (2, 3, 4), and review by AT expert	Think aloud, semi-structured interview, videotape	Compiled list of bugs and interface difficulties	Three iterations of design	Design was simple enough to use with minimal (10 min) introduction
Preliminary field trial of the handheld with a complex script	19-year-old female with cognitive disabilities	Make and install 42-step script on prompter, give to participant with ingredients and utensils	Field notes from observation, videotape and script log	Successful accomplishment of real world task with prompter.	Skipping of steps indicated internal scripts that the script builder had missed, log file demonstrated ability to automate collapsibility analysis
Image experiment	15 high school young adults with cognitive disabilities; 15 age matched high school typicals	Presented cue image on PDA at same time as three life-sized images were presented on a screen. Participants choose a match with a 10-inch diameter button on table in front of screen computer stored results	Counted accuracy (exact match, close match, & wrong) and time from initial image to button push	Data of accuracy & choose/ reaction time with respect to identifying an object from a small image	There were three possible trigger images on PDA: icon, photo alone, and photo in context. All participants did significantly worse with icons as triggers. The young adults with cognitive disabilities took significantly longer to make a match
Realistic MAPS field trial	<u>BVSD Families</u> <u>Imagine! Group Homes</u> BVSD Transition Team	Give dyads MAPS script design environment and MAPS prompter. Observe process of use of system.	Audio recordings, field notes, secondary artifacts	For all four dyads the process of adoption of the system was documented and analyzed	Generated heuristics for script designing and use, successfully made script design environment easy to use by caregiver, but uncovered OS issues. Understood internal/ external script inventories effects on task success

The transition team intends for this client to continue using the system and to migrate the responsibility for the caregiver's role to her dad. They are also interested in getting several more MAPS systems for use with other job coaches and young adults with cognitive disabilities.

## **6.7. Evaluation of Field Trials**

There are two ways to extract usable results for an ethnographically based study. One is to place the insights into an abstract framework, and the other is to relate anecdotes. By themselves, the framework insights become uncontextualized data-bits; in contrast, the anecdotal insights become trivial stores with little ability to generalize. Presented next is a categorized set of insights drawn from the study followed by a number of particularly meaningful anecdotes. These sections contain results of analyzing the qualitative data. Note, however, that the underlying observational data were made without the intention of producing quantitative results and so should be viewed as a broad brushstroke depiction of trends and more of a starting point for a genuine qualitative, rather than a definitive, analysis.

### **6.7.1. MAPS and the Person with Cognitive Disabilities**

An attribute of scripts that is not an immediately obvious component to their success or failure is the voice that records the verbal prompts. This can be subdivided into negative and positive components; the voice of the script can confound as well as enhance the task support process. Young adults with cognitive disabilities are not just persons with cognitive disabilities—they are young adults. In the case of the first family dyad discussed in this chapter, the client is a teenager, and although developmentally delayed, she and her mother experienced all the difficulties that daughters experience with their mothers when transitioning through adolescence. As a result, when the first script was being designed, the mother suggested that she not be the voice on the prompter. The mom realized that prompting instructions coming from the MAPS-prompter would be entangled with the issues of power and independence that the two of them had been experiencing. So a neutral male voice was provided for the recordings

On the positive side, for the Imagine! group home and Transition Team dyads, an interesting phenomenon was observed in the execution of the supported tasks; at certain prompts, the clients would talk back to the prompter, as if talking to the caregivers whose voice was prompting them through the tasks. This interesting projection of the persona of the caregiver onto the recorded script may indicate a certain level of comfort with the prompting support and a “loyalty” to the artifact. This loyalty could be reinforced by the use of the PDA as an MP3 player, possibly leading to promoting success and careful safekeeping of the device.

In discussing this dimension of the MAPS system, Michael Tomasello of the Max Planck Institute for Evolutionary Anthropology [Tomasello, personal communication, 2005] suggested that it might be an interesting experiment to have the prompts recorded by the client that the script was designed for. His logic was that small children, when performing a task that they have not fully mastered, especially by themselves, are often observed directing themselves through the task. He suggested that clients hearing their own voices might reinforce the steps to be done in the same way as for young children. In the four dyads, the opportunity to perform this experiment did not come up, but Tomasello’s idea may shed some light on clients relating to the prompter as if it were the caregiver. The lesson here is that voice counts. It may be important enough that, as the transition team caregiver said, there are three components in a prompt: the image, the verbiage, and the voice.

Several of the dyads put task support markings on the prompter itself. The markings fell into two categories: (1) additions to support general MAPS-prompter navigation and (2) additions to support specific task scripts. Two of the dyads (the first family and the Transition Team) chose to add markings to the front of the MAPS-prompter, above the hardware controls. There are four small buttons and one large button on the lower half of the prompter; the far right and left correspond, respectively, to going forward and backward one step in the script, and the large button replays the verbal prompt for the current prompt. Both dyads added arrows pointing to the right and left above the right and left hardware buttons, and, in addition, the Transition Team labeled the large center button with the name of the job coach, with two goals: (1) to remind the young adult with cognitive disabilities that this replays the prompt, and (2) to remind her that the job coach could always be

“virtually” there. One dyad, the Transition pair, glued onto the back of the handheld the color sorting scheme (see Figure 36). The intent was to provide an “overhead view” of the task to be done.

The script-use observations coding revealed 20 instances of learning the navigation controls of the MAPS prompter by persons with cognitive disabilities, but also 10 cases in the second and third scripts for each dyad where the clients became so befuddled with scripter operation that an intervention with respect to PDA navigation was required. This underlined the need for additional navigation training. However, the majority of the cases requiring intervention were with the clients whose scripts did not have an introductory “generic” section reviewing the affordances of the MAPS scripter, leading to a possible heuristic for script design.

### **6.7.2. MAPS and the Task to Be Done: The Task Perspective**

For some dyads, every script must have some environmental artifact support. For example, the first client on the families study needed the valves and controls marked on the washing machines and on the burners on the stove. Part of the reason for not completely successfully accomplishing the second cooking task (Hamburger Helper) was that the measuring cups were identified by verbal label (“big four-cup measure” and “two-cup measure”) and not marked with colored tape. For the dyad from the Transition program, the job coach pointed out the external aid to color sorting that was posted on racks and in the back sorting room. In the first family study of a young adult with cognitive disabilities at supported employment at a hospital/rest home, the job coach pointed out an auditory cue for properly putting the tops on the water mugs in each patient’s room is a pop when it is seated correctly. She said to listen for the pop. Marking up the environment can become quite tricky; for example, the first family client had problems with separate prompts requiring the same stove control to be set in different positions, requiring more than one color match.

There was also an instance of opportunistically using artifacts existent in the task environment in making the scripts. The caregiver of the second family dyad used the supermarket-supplied listing of the contents of the aisles in selecting the waypoint steps for the shopping scripts. By integrating this into an extension to the MAPS system to support multiple shopping trips, it could be possible to use the list of aisles and groceries to automatically generate item prompt order and waypoint selection. It is

easy to imagine the script-design-environment selecting and inserting prompts for the appropriate waypoints, similar to MapQuest® driving directions.

### **6.7.3. MAPS and the Script that Changes over Time**

In three out of the four dyads, the scripts were rewritten after initial use. In some cases, the rewriting consisted of replacing pictures; in others, re-recording the prompt instructions. In the group home, the initial folding script had both pictures and sounds replaced as well as deleting redundant steps, and steps that had been learned by the client in the first usages. The same lessons that led to rewriting a script can, over time, lead to re-estimating the granularity of each step in the script or internal scripts (see section 7.2), and resizing the following scripts to fit the user's current ability. For two of the scripts, a MAPS distributed cognition became a training script due to the clients' internalizing the task (see section 7.2.2 on tools for living and learning). Caregivers were encouraged to make changes (e.g., expanding, contracting, rewriting) to copies rather than the original scripts, so that earlier versions could be loaded if the need existed. Examples of needing to use earlier versions include returning after a vacation, daily ability variation, or to use as a template for a slightly different task.

### **6.7.4. MAPS and the IPAQ Prompter**

An anticipated problem in using a PDA task support system was that one hand was always not usable for the task because it was preoccupied with holding the prompter. In the realistic trials, this was not as much of a problem as anticipated, but it was still a problem. Some clients put the scripter down between steps (the laundry folding, sweeping tasks, and all the kitchen tasks); some rested it on a natural affordance (the shopping tasks). One found it so cumbersome to switch between looking at the prompter and doing the step that a special IPAQ nylon holder was provided (see section 6.6.3 and Figure 37). This problem needs to be considered with any new adoption of the MAPS system,

Two of the concerns that were expressed had to do with the IPAQ platform and the WIN-CE operating system that ran on it and within which MAPS ran. The first was that the volume and clarity of the verbal prompts might not be adequate for prompting purposes, especially in public places. Second involved the fragility of the WIN-CE operating system, which often locked up during

application development, requiring soft rebooting, and several times requiring a hard reboot, resetting the system to the basic operation system and requiring a restoration of the entire system.

In use, the volume and clarity of the MAPS prompts were a problem in only two prompts in one of the 14 scripts that were built. The two prompts were recorded quite softly and the client had a hearing deficiency, which caused him to incline his head to the prompter at those two steps. The workaround for this problem is to teach the caregiver to monitor the waveform displayed in the middle of the application when tutoring clients in using the Microsoft sound recorder application (see Figure 38). The important part of the tutorial demonstrates a poor recording and a good recording; in one case, a sample script was made of the two different levels of recordings and synched to the handheld so that the caregiver could see how critical the recording settings could be. Although all the clients used the tiny internal speaker to play the scripts, a fallback in the case of really difficult listening environments or moderate hearing loss was to use headphones, but this also leaves the impression that the client is listening to the MP3 player, a solution that AbleLink uses for their “Pocket” series of applications (see section 4.2) on similar PDAs.



**Figure 38 - Microsoft Sound Recorder**

To prepare for either the battery becoming totally discharged or a corrupt operating system, each MAPS-prompter given out was supplied with a secure digital memory card with between 256 MB and 1 GB capacity. In the case of battery discharge, when the PDA is recharged, it uses the OS and applications that were stored in ROM when it was manufactured. To correct both the corrupt and missing OS functions, the caregiver was expected to restore the system from a system backup stored on the secure digital card. Also, the caregivers were shown how to do a “soft reboot” to reset the system as an initial response to a locked-up system. Among the four dyads, use of the hard-reset and

restore fix had to be applied in several script-playing sessions. However, over the long run, the WIN-CE OS has proven itself more robust than was originally expected.

In the process of the caregivers learning and using the script-design-environment, it became clear that these PC novices (the group that the system was designed to support) needed more support than simply installing the application and demonstrating the video help system. There were many instances of “getting lost” in the intricacies of the Microsoft Windows file system, which were solved by making shortcuts to the base directories for the images and for the recorded prompts, as well as for the MAPS application and the OS-provided voice recorder. This should be made part of the installation script for MAPS. Additionally, several of the caregivers had problems with file names for the images and sounds, there was no structural fix for this, and a half hour on the telephone cleared up the confusion.

### **6.7.5. MAPS and the Caregiver**

Perhaps the biggest initial obstacles to success in using the MAPS system were the caregivers’ task choice, segmentation, and script design. Three of the four caregivers in this study had problems with task segmentation in the initial script. This manifested in several ways:

- Overestimating already held skills or internal scripts (see section 7.2) (i.e., too small a granularity of steps)
- Underestimating already held skills or internal scripts (see section 7.2) (i.e., too large a granularity of steps)
- Not encoding tacit knowledge
- Missing or misstated steps
- Not enough prompter navigation support

The second family and the group home caregivers both overestimated the internal scripts for their clients. In the case of the second family, the client’s cooking ability and ability to use stove controls caused the spaghetti script to require multiple interventions. Similarly, the ability to manipulate a hanger and the proper method of hanging a shirt on it caused the caregiver in the group home to rewrite the script with expanded support for this part of the task.

Underestimating internal scripts can cause the client to skip forward and not stop and cue the next prompt at the end of following *this* prompt. Underestimating can also cause the client to experience frustration at the slowness of the process. Both families and the group home client in the trial experienced skipping ahead due to underestimated internal scripts. This could be observed when the client did one prompted action and then immediately did the next prompted action without advancing the prompter. Often this continued until eventually the client had gotten to a state that did not match a step in the tasks script, requiring backtracking, and at times in this study, intervention. The solution for this problem was to rewrite the scripts with larger chunks for external script cueing. Also physically doing the task following the script explicitly sometimes caused “Ah ha!” moments when it became obvious to the caregiver that they were breaking the task down into too-small steps; this last solution can also apply to overestimating internal scripts.

Sometimes the task would be encoded improperly by missing tacit knowledge or missing a step. In the beginning sections of cooking scripts, caregivers usually prompted all the utensils needed, but in the middle of a task a spatula magically appears. Similarly, a sweeping script did not move a piece of furniture with the explicit removal of the other furniture, and by the middle of the script it had mysteriously “disappeared.” In the 14 scripts, there were coded instances of 14 missing steps.

Another class of problems concerned the navigation of MAPS and the use of the prompter’s controls. In every case, some of the client’s knowledge of the operation of the MAPS-prompter faded between sessions—typically a one-week period. In actual use, MAPS would be used more often than in the trials. The first family’s caregiver created a set of prompts to review the operation of the prompter, which some of the other caregivers inserted into some of their scripts, but the clients still had problems with the scripts without the reminders.

All of the caregivers expressed that using the templates helped them in script design, either using it as an actual template for the task or just seeing how other tasks are segmented.

An interesting phenomenon occurred when the caregivers wrote and recorded the prompts in the initial scripts. Several caregivers wrote and recorded what we ended up calling “robo” scripts, in which the script prompts would lack personal references as directly addressing the client by name or not



using personal pronouns. The general tone of these “robo” scripts prompts would be clipped and abbreviated. In both situations, this was pointed out to the caregivers and they were shown examples of scripts that were more personal and tailored to the client.

Rough statistics were kept about the time spent making and loading scripts to the handheld. Typically, it took about between 1.9 and 4.0 minutes per step to collect and assemble a script. The variation between 1.9 and 4.0 reflects the difference in the caregivers’ self-labeled “perfectionism.” To load a script from the script-design-environment to the MAPS-prompter took on average about one minute for every 10 script steps, so that loading a 60-step script would take six minutes.

One of the assumptions driving this study was that a significant cause of high-functioning AT abandonment is difficulty in (re-)configuration (see section 2.2). All of the caregivers expressed that one of the easiest parts of using the MAPS system was the script-design-environment. The video help system and the availability of templates were mentioned as significant in making script creation easy. When the script creation task was enlarged to include the prerequisite tasks of preparing the OS file system and naming the files appropriately, however, a slightly different picture emerged. The percentage of all coded interactions with the caregivers that were related to script-design-environment problems, counting requests for new functions but not counting bugs in the script-design-environment that were fixed as of the current MAPS release, and including the aforementioned operating system issues, was 19 percent. Without the OS issues, the percentage dropped to 14 percent. Most commonly mentioned was that they had not really understood the coverage of the video help until requesting intervention, something that needs to be incorporated into the half-hour introduction to the MAPS script-design-environment.

Some caregivers (the first of the families and the group home) decided not to follow the instructions for using the script-design-environment to design scripts, and decided not to name the wav files (recordings of the prompts) with the verbal contents of the files. Instead, they assigned sequential numbers to the files. As discussed previously (see section 6.6.2), this behavior is not unfamiliar in computer-supported collaborative endeavors. A possible workaround could be providing an integrated applet that used the script template text to assign the names to the files from the notes column. All of

the caregivers in the study used the template, and the verbiage in the notes section was close enough to the actual words to allow reuse and anonymous use as templates.

All of the caregivers expressed concern about the steps required to copy a finished script from the PC-based script-design-environment to the MAPS-prompter. In every case, they needed intervention and a demonstration of how to do the transfer, which required copying the chosen script to a table in the database on the PC, running a batch file on the PC, and, after ensuring the active synch application was running on the PC and PDA, running an application on the PDA, triggered inside the MAPS script prompter (see Figure 23). To alleviate this problem the script-design-environment was modified to run all the PC-based database actions and the batch file. One request by the caregivers was either to add guidance for this process to video help or, at minimum, to write a “crib sheet” similar to the examples that the caregivers made in appendix 13.16.

Over the course of observing 17 sessions involving the use of the MAPS scripter by clients, there were many instances of caregiver interventions. Of the 55 coded cases of observed interventions, 16 percent involved supporting the learning of the prompter use, 16 percent brought the client back on track for the scripted task, and 5 percent ensured the safety of the client (i.e., cooking task with stove). These interventions demonstrate a need for further initial training in MAPS-prompter navigation and script-following for clients. However, 61 percent were interventions by caregivers who interfered with the process of learning to use the prompter on the task. Most of these happened early in the field trials, and the caregivers were mothers of the clients. In further similar trials, the protocol should control for these unnecessary interventions that also interfered with observing breakdowns and therefore losing the opportunity of insights that breakdowns often expose.

In the case of using MAPS to support employment, the job coaches immediately saw a need to elaborate on linear prompts. With rare exceptions, employment tasks that are appropriate for this population involve repeating a chunk of behaviors until there are no more objects left for the behavior. Also common are doing a subtask until a decision point is reached, and then choosing one of several options to continue. The initial task for scripting that the transition team was working with required both. This brought up two themes in the study: caregivers becoming co-designers of the script-design-

environment and prompter, and the need to somehow encode soft skills in employment-supporting scripts.

One interesting extension of the metadesign (section 3.5) perspective was the request made by caregivers for additional functionality from the MAPS script-design-environment. The first family caregiver, irritated at scrolling up and down a directory box to locate the next sound recording for each prompt, requested a better way to preselect the sounds all at once and then support the serial choosing and inserting the wav files one at a time in the preselected order (see section 5.4.2.1 and Figure 30). This was a natural consequence of the difference in the affordances provided by image files, which can be looked at 16 at a time on a screen, whereas the sounds can either be looked at six at a time in the directory box and requiring lots of annoying scrolling to find the right next file, or worse—if the caregiver used meaningless file names—listened to one at a time. A prototype was delivered to the caregiver, and upon further refinement, added as an option in the utilities menu bar (“Sort prompt files for script”). Another requested MAPS modification was the group home caregiver’s addition of forking and looping functionality to the system (see section 5.3.3). Again, a prototype was brought to the caregiver, and with some small modifications it was folded into the release code of the MAPS system.

#### **6.7.6.MAPS as a System**

The goal of MAPS was to support independence, and several instances of this happened in the process of realistic field trails. The first family dyad’s shopping task script was the most clear example of supporting independence: the young lady using the script to successfully shop for family groceries was a complete example of how computationally supported prompting could provide a lever to extend the existing skills of a person with cognitive disabilities. Even the first script attempted, the “sweeping the kitchen” script, which in many ways was not a success, was retained by the daughter. Regarding her daughter’s new chore, the mom said, *“She did not learn how to sweep the floor the way the script showed her, my way; she sweeps the floor, but she does it her way.”*

The second family dyad was not as clear a success, and this may illustrate an attempt to introduce MAPS to a person with cognitive disabilities who may have been on the bottom edge of “able to use

the MAPS-prompter to accomplish tasks” (see section 2.1.3). During the exit interview, however, the mom said that the client had done the family’s laundry several times as a result of learning to do so with the script.

The dyad from the group home, while perhaps providing an example of MAPS being used by a person with cognitive disabilities who was on the edge of “not needing to use MAPS for task support” (see section 2.1.3), was also an example of the “an island of deficits in a sea of abilities” principle (see Figure 9). Although he did not need the scripter to support a trip to the library, the large script for folding clothes was a big success, both for his esteem and for his day-to-day living skills.

Finally, the dyad from the BVSD Transition Team used MAPS to support real employment being offered to the young adult with cognitive disabilities in a much shorter time than her job coach expected. When the client of the transition team dyad expressed anxiety about being able to successfully do her job without her job coach, her job coach told her that she did not *need* her support, that “all you need is either up here [gesturing to her head] or in here [gesturing to the MAPS-prompter].” Perhaps the biggest indication of the success of not only the technical design but of the system’s integration into a socio-technical environment that works is the fact that all four of the dyads requested to keep the MAPS system.

In reviewing the script sessions during the exit interview, the caregivers cited several instances of the failure of the MAPS system that either caused the caregiver to state that they would not adopt the system unless remedied or were sufficiently onerous that adoption might be made more difficult. From the coding analysis of observations, of the 13 instances of “failure,” 30 percent were due to the requirement for file manipulation to conform to the 64 Kb limit of binaries stored in the Sybase Ultralite database on the prompter, which made it necessary to submit the image and sound files for processing before they could assemble the script themselves. Caregivers really wanted to do the whole process, from task segmentation to image collection and sound recording to loading the script on the prompter completely by themselves. Midway through the field trials, the 64-Kb limit was programmatically solved, so that requirement was removed. Of the rest, 30 percent were a result of a failed script design, half of which were corrected in subsequent scripts made by that same caregiver;

the other half of the failures were a result of a mismatch between a client’s skills and needs (specifically, the library script was provided to a client who already had the ability to do the scripted task on his own)—a situation that could be labeled as “shooting too low.” Confusion about locating and accessing directories for the image and sound files accounted for 15 percent of the problems, which was solved by placing aliases for the directories on the desktop, and 15 percent of the issues were about the file-naming requirement for wav files, which was addressed by education (see section 8.2.3 for a proposed solution). The remaining 7 percent were caused by problems with clients getting “off track” by requiring them to wait long enough between tasks to become distracted, which was addressed by breaking the task into two tasks. There is some discussion of a technological solution to this problem in future work (see section 8.2).

## 6.8. Realistic Field Trials and Heuristics

As previously discussed, some of the insights gained in this research are applicable only to AT design and use, and others bear ramifications to the larger HCI world. The following is a short list of heuristics for the design of a tool with high-functioning AT for persons with cognitive disabilities, and then a set for script design. The following sets of suggestions came primarily from the realistic field studies, but the usability tests for the script-design-environment contributed several ideas as well.

### For general application design:

**Leverage existing PC skills.** By using the existing skills and models that the users bring to the device designs means needing to challenge only the users with new functionality [Lewis & Rieman, 1993].

**Produce “video help”** to provide an alternative to the widely unused text or hypertext help for high-functioning AT. Each segment should illustrate exactly one idea, allowing the user to dip into help and immediately get back to the application. Video help can be easily implemented by using captured video for each small use segment and annotate it with a voice over. Don’t assume that the user will use text-based help. Try to provide innovative help that gives the users only the help that they need in a form they will use. MAPS video help was consistently praised by the caregivers for these two reasons.

**Ensure low entry cost to use the system.** Don't use "innovators" as usability testers who would struggle to make the application work; instead, use "late majority" users [Rogers, 1995] to ensure poorly designed interfaces are highlighted

**Take responsibility for as many operating system interactions as possible** by moving them into the installation. The failure points of the system are often the places where the user is forced to interact with difficult-to-use OS affordances, or ones that are based on a different model than the application. Microsoft's assumptions about how the user interacts with the file system ("My Documents," etc.) may conflict with how the application deals with persistent data. Take responsibility for the way that the user accesses the system functionality. The MAPS installation eventually set up the caregiver's PC with aliases for key parts of the application. Lack of this kind of support was the primary place the early MAPS system broke down.

**For AT tool designing:**

**The designers of metadesign tools to be used by dyads need to be explicitly aware of the underbuilt/immediate satisficing result trade-off.** Make sure that the user can immediately produce something useful; the user may give the system only one chance to succeed.

**The tool needs to be underbuilt and general enough that it might support artifacts as ambitious as possible.** Scenario design is helpful here, to be explicit with ideas as far out as the end user can imagine.

**For any sufficiently complex AT system, the system needs to incorporate the dual user interface framework.** Identify stakeholders and provide appropriate interfaces for each.

**Make the application as "toaster"-like as possible.** The added functionality of the general computer that the AT application is based on can be confusing and make the system more brittle than necessary. Disable as many controls and features as possible and reduce functionality to a minimum necessary to support the desired goal.

**For designing scripts:**

**Provide explicit training in task selection and segmentation.** The lack of training in breaking up tasks was a pronounced difference between the professional caregivers (Transition team and group home staff) and the family caregivers. Improperly segmented scripts were the second most frequent cause of breakdowns

**Show examples of well-formed task segmentation.** The introduction to MAPS included a film [The Oregon Research Institute, 1988] designed for special education teachers.

<p><b>Discuss the material</b> the film presented (topics such as chaining prompts and error correction). The MAPS introduction used the workbook that came with the video as a springboard for discussion.</p>
<p><b>Present script design support tools to the caregiver.</b> In the case of MAPS, a MS Word-based script worksheet form was included (see appendix 13.6).</p>
<p><b>Do the task yourself</b> before attempting to segment the task. Mentally doing the task is not a replacement for physically doing the task in the actual environment where the task will be performed.</p>
<p><b>Write down each step, especially the words.</b> By being explicit about the steps used to build support for the task, missing holes and overwriting (underestimating the internal scripts of the client) can be avoided.</p>
<p><b>Having written out the script, have someone else review it.</b> This is particularly important during the initial plotting of scripts in the MAPS system. This can be done from the filled-in MS Word script worksheet or by using the script on the handheld, but in either case have someone else attempt to do the task. If no one else can help, doing it yourself can substitute; however, the caregiver needs to be careful that tacit knowledge does not slip into the performance of the task. In any case, do the task using only the prompt words and pictures as guides.</p>
<p><b>Use the templates as supports for making scripts.</b> The MAPS system had almost 200 templates on its web-based database server, which the caregivers found useful in the initial script design.</p>
<p><b>Don't overestimate the prompter navigation skills of the person with cognitive disabilities.</b> Remember that MAPS is remapping the memorizing of the steps, constituting a task to learning the use of the MAPS-prompter. Always add the short generic navigation sub-script (see section 6.6.1) for the first several scripts.</p>
<p><b>Back up the PDA frequently.</b> PDAs are fragile as are their operating systems. Additionally, the batteries on the handheld are limited enough so that there certainly will be a need to restore the MAPS-prompter. When the batteries exhaust themselves, the MAPS-prompter, after recharging its batteries, returns to the "factory" version of the OS (that was stored in ROM when it was produced), and to be used as a MAPS-prompter requires restoring it from a backup that is located in the Secure Digital memory card on the MAPS-prompter. Practice restoring the system when a dyad initially learns to use it, so that restoring it doesn't become a breakdown point. After every new script is installed on the prompter, a backup needs to be done, which typically takes about 15 minutes. Of the four dyads evaluating the MAPS system, three had to have the system restored from the memory card due to battery discharge. Occasionally, the operating system can become corrupt, and the only solution in this case is to restore the MAPS-prompter to the point where it is was working properly; the fourth dyad needed this first aid.</p>

**For making scripts from an existing design:**

The first three guidelines refer to the process of capturing the images of the prompts for use in the MAPS script-design-environment:

<p><b>When taking pictures for the script, always preview them in the script-design-environment.</b> If there is some concern about lighting and the items depicted (e.g., you want them to use the frying pan, yet it appears only in the edge of the image), put it on a script and transfer it onto the prompter and look at it.</p>
<p><b>Take all images in portrait orientation,</b> not the usual (landscape) way that cameras are normally oriented. This needs to be said only once at the start of the process of a caregiver creating scripts; the compression of any image with the normal wider than tall images will make the need obvious.</p>
<p><b>Shoot the image photographs at the lowest resolution that the digital camera will allow.</b> PDA screens are relatively low resolution devices, and displaying an image at the higher resolution does not make a perceptible difference, but takes a much longer time to unload from the database on which the script is stored in the MAPS-prompter—it takes so long (greater than a minute) that it appears that the system is broken.</p>

The last four guidelines refer to the process of recording the prompts for use in the MAPS script-design-environment:

<p><b>Pay attention to the fidelity and volume of the accompanying verbal prompter.</b> Notice volume levels and clarity (i.e., lack of background noise). Some scripts will be done in public places with background noise levels higher than indoors. In certain situations, the use of headphones is recommended</p>
<p><b>Record the prompts by using the Microsoft sound recorder</b> (see Figure 38) provided with its operating system. For the same reason as the guidelines for the image photography, set the recorder to 11.025 kHz, 8 bit, mono (telephone quality).</p>
<p>When recording the prompts, <b>do not leave too much room in the front of the recorded voice</b> before you hear words; the user may think that the application is broken if nothing happens right away (i.e., the broadcaster “dead air” rule).</p>
<p>When writing and recording scripts, <b>be careful to be as natural as possible and use the clients’ name in the prompts.</b> Several of the caregivers wrote or recoded prompts that sounded “robotic.” There is no need to try and sound mechanically authoritative or clipped in composing prompts.</p>



**Carefully choose the voice to record the verbal prompts.** As has been previously discussed (see section 6.7.1), the voice that records the prompts may have a large impact on the success or failure of accomplishing the task. Some, but not all, of the clients had a definite preference, whether negative or positive, on whose voice recorded the prompts. Be aware that the voice may affect performance.

## 6.9. Key Design Contributions

In the process of performing realistic field trials with the MAPS system, several important work practices were noted. These practices were supported by specific design elements in the MAPS system that were included in anticipation of their evolving out of actual use. In addition several secondary physical supports were created by the caregivers from their experiences using MAPS. The three practices were:

- Collapsing of scripts
- The use of multiple perspectives of the script as it was being created / modified the script design environment provided ability to go between seeing the whole of the script as well as the details in the design process.
- The reuse of scripts and fragments of scripts

These three elements were considered important elements of the scenarios of MAPS usage for the beginning of the design process. Confirming their relevance and generalizability was the fact that collapsing and reuse were each independently “invented” by caregivers. However, complete and useable functional support for either was not implemented by the time that the realistic trials of the system commenced.

### 6.9.1. Collapsing Scripts

Collapsing scripts describes the adjustment of scripts over time to reduce the number of prompts for a given task. Scripts are initially created to reflect the client’s existing internal script inventory. As chunks (a contiguous set of prompts) of the script become internal scripts, the script is modified to reflect that new inventory state. An example of this was the clothes folding script of the client/caregiver dyad from the Imagine! group home. The initial script design called for a 61-step

script, presenting in great detail the tasks of folding and storing in a dresser or on a hanger various different kinds of clothing such as pants, T-shirts, underwear, etc. As the client used the script, the caregiver noticed that for certain chunks of prompts, once the client hit the beginning of the set of prompts covering the subtask, the client put the prompter down and continued until he did all of the steps for the subtask and then picked up the prompter and rapidly stepped through the prompts that detailed the subtasks he had just done until he got to the prompt for the next subtask, “labeling.” This practice was reflected in the prompter log for that instance of running the script.

This collapsibility was first noticed in the use of the chinese cookies script described in section 6.4, “Preliminary Field Trial of the Handheld with a Complex Script.” The client doing this cookie script took an average of 38 seconds per step; however, at step 21 she took 3 minutes and 8 seconds, and then after that the next five steps took 5, 4, 5, 4, and 11 seconds each, and after that the next step took 30 seconds. This reflected the fact that she reached a set of steps that she knew and when the first of this sequence came up, she continued to do the following five steps without using the prompter to tell her the next step until she “ran out” of her internal scripts and had to go back to the prompter’s next step of the script to continue. In this case, the script running log would have enough data to flag the caregiver to consider collapsing the script at the initial (3 minutes, 8 seconds) step, by deleting the next five prompts and perhaps changing the 3 minute, 8 second prompt to explicitly “contain” the five following prompts.

The current MAPS system does not support the sort of automatic script log analysis described above. The two examples of collapsing a script, the folding and the cookie scripts, were triggered from caregivers’ observations, but automatically analyzing the script logs would produce the same analysis that the caregivers’ observations led to. In the case of the folding script, the caregiver used a script design paper template that supported collapsing the script by marking some steps as collapsible (see appendix 13.14). By guessing where the task would become memorized (in this case, the T-shirt section), she was able to provide a subtask label prompt before the set of prompts that would constitute the new internal script. When the client memorized the subtask of folding the T-shirt, it was a simple matter of removing the newly memorized steps and saving the script with a new name. By keeping the old script and not overwriting it with new set of prompts, the caregiver had the option of using the old

script with the details of T-shirt folding whenever the client might forget the details of the memorized subtask.

### **6.9.2. Multiple Perspectives of the Script and Prompts**

In the initial design sessions for the script-design-environment and after looking at other commercial offerings, it became clear that a script could succeed or fail depending on the clarity and applicability of the images used as prompts on the handheld prompter. The problem was driven by the “universe of one” problem of the person with cognitive disabilities not being able to abstract and also the small size of the image on the PDA (280 x 210 pixels). To illuminate possibly problematic images, the MAPS script-design-environment provides the ability to preview any given prompt in the image review frame (see section 5.4.2). The caregiver could also preview the entire script at the same size as on the prompter in the same preview pane of the script-design-environment. These functionalities were typically not used for the first script, but by the time the third one was designed, three of the four caregivers had used them.

The script-design-environment originally provided only two views of the prompts: the filmstrip at the bottom of the screen and the individual prompt image preview in the preview pane (see Figure 25). By using these views alone, it became clear that for most real-world scripts (which ranged from 20 to 60 steps), it would be easy to become disoriented with respect to where this set of five prompts displayed in the “filmstrip” view in the bottom of the script-design-environment were in the whole script. An overview perspective was obviously needed, and the three- and four-thumbnail-wide image script preview was added to the script-design-environment to allow designers to orient themselves and to quickly go from one place in the script to another that may be 20 or more steps away (on Figure 28 , the right-hand side of the screen shows the three-wide full script view). After using the “preview at a glance” function, the caregivers, when modifying the script, no longer used only the paper template to orient themselves but were observed to “fast forward” to the desired script step by clicking the preview thumbnail. Again, initially this was not used by the caregivers, but as the scripts became larger, they all used this functionality.

### 6.9.3. Reuse of Scripts and Fragments of Scripts

The reuse and sharing of scripts was part of the earliest scenario script design sessions. In the process of researching best practices of script creation, about 170 examples of relevant scripts designed for persons with cognitive disabilities were accumulated [Saskatchewan Learning – Special Education Unit, 2003]. Implementing support of design by modification (see section 5.1.3.2), these examples were converted into MAPS scripts and placed on a MAPS database server reachable from the Internet. Caregivers could switch between the local database and the template database (see Figure 15) by downloading a template script, which had generic images and the content of each prompting step identified in the name of the file containing the verbal prompt. Three of the four dyads took advantage of the stored templates to support them in the designing their second or third script. Interestingly, the caregivers' final scripts (*going to the library, shopping for groceries*) retained few of the specifics from the templates. In exit interviews, they expressed that the main help that template scripts provided was a guide to start *thinking* about the task to be segmented.

There was another type of script reuse amongst the dyads. The first caregiver to use the system created a set of three prompts that she inserted at the start of every script to help the young adult with cognitive disabilities remember the controls of the MAPS prompter. This was an interesting recursive bit of distributed cognition; remember that MAPS allowed the client to transform the skills needed to accomplish a task from memorizing the steps and their correct order to knowing how to use the MAPS prompter and follow the instructions encoded into the prompts [Norman, 1993]; in this case, the instructions were how to use the MAPS prompter. This set of prompter instructions was passed to each of the following dyads and was used in two other cases. It is this sort of horizontal sharing of scripts that the MAPS database server was designed to support.

### 6.10. Concluding Thoughts about Assessment of MAPS

As had been said at the start of this chapter, there are many ways to analyze a system, and it is unusual to find a system analysis for which the majority of the probing and analysis is focused on the participants in a system rather than on the system itself. So what can be said about MAPS from this perspective? Socio-technical environments are never complete [Mumford, 2000a], so any analysis of

MAPS must be an analysis of the system in real use. Emergent from the ethnographic study was a rich pattern of detail, showing flows of artifacts and information from tool designer to caregiver to client and from client to caregiver to tool designer. This flow consisted of scripts, script usage information, the rewritten scripts, and the script-design-environment and its modifications. An example of this flow and its analysis is shown in a MAPS script log with a long time between one step and the next and then almost no time at all for the next five steps. An analysis of this might be that the caregiver overestimated the need for task support for this section of the task, and that the person with cognitive disabilities had a larger internal script for this topic than the caregiver knew. Then the caregiver's response might be to rewrite the script by collapsing four steps into the first of the five skipped steps by re-recording the prompt for that step to trigger the whole internal script that the client had.

To document and analyze the complete process of adoption of a high-function AT device as a cognitive orthotic<sup>15</sup> would take many more months than this small study. To properly understand the change in relationships and in day-to-day life, however, would require not just a longer qualitative study but also a deeper understanding of the nature of what independence optimally means in the complex of the dyads that are adopting the system. That said, this short study in a semi-controlled environment produced many insights that extend both ability and knowledge in AT design as well as providing a good foundation for further studies.

Another difference between controlled quantitative analysis and this, more situated, probing study is that the investigator experiences personal changes in the process of studying the dyads. By spending time with the young adults with cognitive disabilities, the investigator's sense of what is "normal" stretches. In addition, the investigator experienced the loss of a certain elitist perspective about people that look and act very differently than his family and professional associates. And this is a good thing.

---

<sup>15</sup> An orthotic device or orthosis is a device that is applied externally to the limb or body. The purpose can be to provide support or extension of a function. A *cognitive* orthotic, as in this case, supplements (not replaces) cognitive function.

## 7. Theoretical Contributions

### 7.1. Overview

The theoretical framework described in the third chapter provided support for this research in several ways. Initially, the perspective of distributed cognition gave hints about what to look for in studying existing practice and the target population. Then, following in the footsteps of Hutchins [Hutchins, 1994] in examining the existing practice of prompting by special education teachers, the investigator looked for artifacts and behaviors to fill slots that had been elucidated in his writings and those in Salomon's collection of studies [Salomon, 1993]. As the landscape of the domain became more familiar and ideas expressed as scenarios were explored, discrepancies arose between what was expected to be seen and what was observed. Further, there were roles and relationships that were not discussed in the canonical literature. From these breakdowns and similar ones that came from using the principle of symmetry of ignorance to investigate stakeholders and proceeding to prototyping using the design methodology of metadesign, insights occurred. These tentative conceptualizations, noticing the importance of external scripts' reliance on internal scripts or the dual-user interfaces became strong elements in the design and analysis of the MAPS system.

This chapter elucidates the insights and abstractions that derived from the application of this troika of theory, principle, and design methodology that comprised the theoretical framework supporting this research.

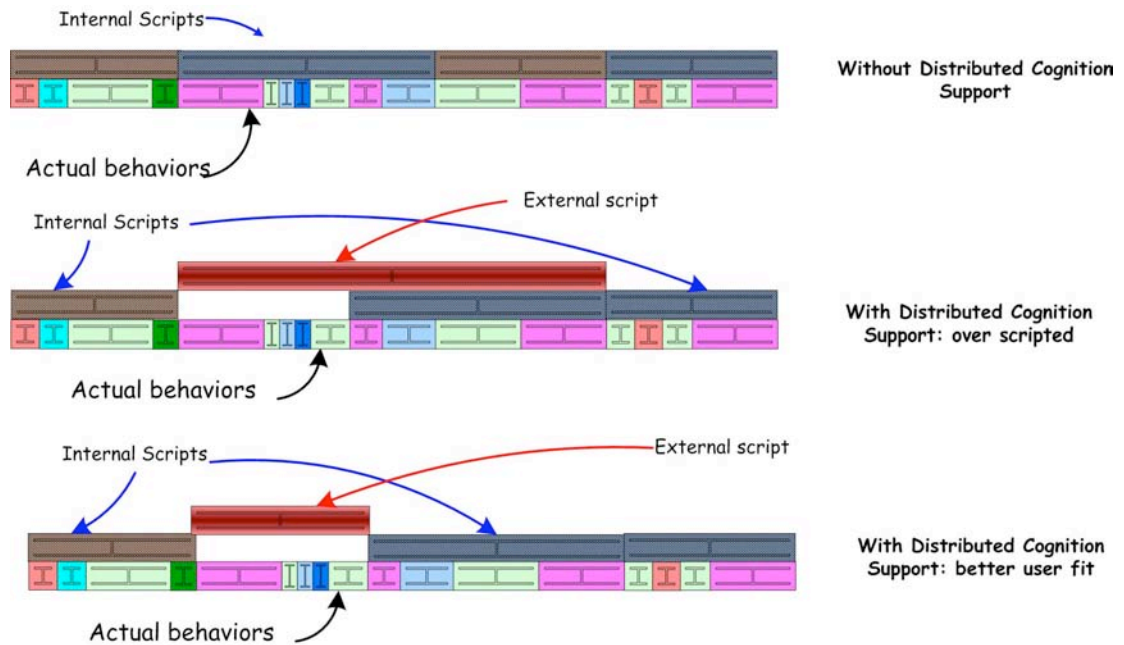
### 7.2. Distributed Cognition

It is initially obvious that prompting is a case of distributed cognition, particularly prompting instantiated as a permanent exterior task support. However, in answering Pea's three questions (see section 3.3) the issues of what is distributed led to considering the nature of the existing knowledge about the task-at-hand and its distribution (see section 7.2.1). Further thought about the dynamics of the distribution of the parts supporting the act of distributed cognition led to thinking about and

studying learned scripts and permanent support scripts, and how they might transform from one to another, as well as the properties of each kind (see section 7.2.2).

### 7.2.1. External and Internal Scripts

Scripting can be seen as an instance of distributed cognition. Cognitive scientists look at knowledge representation, particularly operational knowledge, in terms of scripts and frames [Schank & Abelson, 1977]. In the MAPS systems view of scripts, however, they are regarded as exterior and supportive rather than as internal structures. Traditional rehabilitative use of scripts is intended to lead to the memorization of the script steps, thus tying together these two perspectives. In the MAPS system, scripts are designed to be external supports when the internalization of the sequence of instructions is not possible. From this, one can discuss *internal* scripts as sequences of behavior that have been memorized and can be appropriately evoked to accomplish a desired task, and *external* scripts as the distributed cognition artifacts to simulate an internal script [Carmien et al., 2006]. Figure 39 illustrates the external cueing of extant “atomic” behaviors by an artifact or human support. The top portion refers to a person with sufficient internal scripts to accomplish the whole task, the bottom two sections of the illustration refer to a person with all the “atomic” behaviors to accomplish the task but not having the internal scripts to tie them together in sequence and detail. The middle demonstrates a case in which the external support is more than the person needs to accomplish the task, therefore possibly creating confusion or boredom (see section 1.1 on optimal flow). The bottom section shows the right level and fit of external support (see the second script in section 6.6.2).



**Figure 39 - Internal and External Scripts**

As an example of the process of internal scripts, consider when children become old enough to dress themselves that the various executive and mnemonic tasks involved with selecting, donning, and fastening clothing become part of an internal script that can be appropriately “run” when required. For some persons with cognitive disabilities, the various internal scripts involved in the task of going to the store to buy milk may not be available; perhaps all the components but the travel component exist and are appropriately accessible. MAPS may provide an external script, in the form of prompts to use the bus to the store, to accomplish this whole task.

Table 10 demonstrates the relation of these internal and external scripts. Even people suffering from severe cognitive disability have functioning internal scripts for simple functions such as eating or walking. The MAPS system envisions its external scripts as bridging the gaps where the internal scripts do not support the complete task behavior.



**Table 10 - Overview of Scripts, Internal and External**

Topic	Internal Scripts	External Scripts
<b>Definition</b>	Sequences of behavior that can be elicited and performed without outside support	Cues of sequences of internal scripts
<b>Examples in AT World</b>	Getting bus pass from purse	Taking a bus trip (which at some point entails getting that pass)
<b>Examples in Non-AT World</b>	Selecting an item from a applications menu	Setting up an application to print on a specific printer
<b>Distributed Cognition Perspective</b>	Knowledge in the head about the world	Knowledge about the world derived from the juxtaposition of knowledge in the head and artifacts in the world

The cognitive act can, then, be seen in an individual accomplishing a particular task being guided by (a) the internal scripts individuals are holding with respect to the target activity, (b) the external scripts that are provided in the external surroundings of the actor(s), and (c) an interplay between those internal and external scripts. From the internal script perspective, the term “script” describes the knowledge and strategies that an individual possesses and that guides actions and understanding in a specific situation. In cognitive psychology, *“a script is a structure that describes appropriate sequences of events in a particular context. A script is made up of slots and requirements about what can fill those slots. The structure is an interconnected whole, and what is in one slot affects what can be in another”* [Schank & Abelson, 1977, p. 41]. Schank and Abelson use the term “script” predominately with respect to rather well defined situations, the knowledge of which is acquired through repeated experiences with similar situations and that can be assumed as being culturally shared to a certain extent (e.g., a “restaurant script”).

In contrast to cognitive psychology, computer science uses the term “script” to describe guidelines in the surroundings of an individual that provide procedural support for accomplishing a specific task or a class of tasks. These external scripts can take on very different forms; that is, they can be represented in many different styles and provide *affordances* [Norman, 1993] for desired actions and constraints for undesired actions, and they can do so in an explicit or a more implicit

manner. In the case of MAPS, the external script is the script resident on the hand prompter, each step of which triggers (when properly constructed) an internal script that is already possessed by the person with cognitive disabilities.

Concrete examples of the interaction between internal and external scripts were plentiful in the realistic field trials. The young adult with cognitive disabilities and his caregiver from the group home provided two interesting examples of such interactions. In the case of their first script, the folding script (see 6.6.2), the caregiver progressively collapsed the script steps as the client internalized the various types of folding skills. In the case of the second script, the trip to the library, the caregiver underestimated the internal scripts the client had in this domain, and wrote a script that provided too much support. In the end, this client's experience tended toward the boredom extreme of Csikszentmihalyi's flow diagram (see Figure 4). A similar experience in the other direction was experienced by the first of the family dyads in their cooking scripts (see section 6.6.1). The mother overestimated her daughter's ability in the kitchen, and both cooking scripts fell short of success. These examples underscore the need for caregivers to accurately inventory the internal scripts of clients.

### **7.2.2. Tools for Learning, Tools for Living**

In the process of reviewing the literature and doing research into existing AT with experts and available systems, an interesting taxonomy emerged. Some of the devices were clearly designed to replace lost functionality (e.g., wheelchairs and screen readers), and yet another group of tools were designed to aid in "rehabilitation" [Scherer, 1996], to help restore (or in many cases implement never existing) abilities [Carmien & Fischer, 2005]. More interestingly, when looking at the field from this perspective, it became obvious that some of the mismatches in implementation and adoption of AT were the result of a mismatch between the client's needs and the mode of the tool that the device or system was intended to support. An example of such a misfit is providing a bike with "temporary" training wheels when an adult tricycle might be more appropriate for a person who might be incapable of learning to balance on only two wheels (see Figure 40 and Figure 41). Thinking about systems and artifacts in this way affords insights into distributed cognition, the design and use of artifacts, and

educational decisions about learning and teaching skills with and without tool support. Table 11 summarizes these insights.

**Table 11 - Overview of Tools for Living and Tools for Learning**

	<b>Tools for Living</b>	<b>Tools for Learning</b>
<b>Definition</b>	Doing tasks with tools	Learn to do tasks without tools
<b>Examples</b>	Eyeglasses, phone, radar, cockpits, scuba-diving gear	Spelling correctors, handheld calculators, tricycles, wizards
<b>People with Disabilities</b>	Spelling correctors, handheld calculators, tricycles, wizards	Learning Braille, learning how to use prompts, learning a bus route
<b>Prompting Systems (MAPS)</b>	For people with memory problems (disabilities, elderly)	For training the client
<b>Distributed Cognition Perspective</b>	Resource rich (professional life)	Become independent of external resources (school)

Rough descriptions of these two tools emerge:

- A tool for living is something I need to use to do what the “normal” population does. Eyeglasses are a good example: “As long as I wear my eyeglasses or contact lenses, I can see as well as anyone else; when I don’t wear them. I cannot [excluding laser surgery].”
- A tool for learning is something that I use at some point that leaves me changed so I can do, by myself, what I could not do before. A good example of this would be reading (learning to read): “After I learned to read, I could access the collective knowledge embedded in books (and web pages).”

Some things, such as spell checkers, fall into both categories, but this simple dichotomy seems to provide a way of discussing artifacts and, more interestingly, design of artifacts that can be helpful

A primary motivation for using tools for living is, for example, for a person who does not have the ability to do many ADLs unaided and does not have the capacity to learn (say, how to take a bus), and is thus forced to rely on the help of a caregiver to do these tasks. A simple example of this is the

jar-opening tool that people use to help to get the tops off of jar lids. Without this tool, one must ask someone stronger than they are to help. With the tool, however, one is independent of asking for help in doing this ordinary task. By providing socio-technical environments that would aid persons with cognitive disabilities to accomplish what they would otherwise need someone to help them to do, that person's sense of independence is increased. Both kinds of tools, tools for learning (TfLea) and tools for living (TfLiv), extend our independence, but TfLiv devices provide support that is more separate from our innate abilities.

The motivation and outcome of the use of either of these tools result in a greater ability in the world. A perspective on this intimate use of artifacts follows:

*The invention of eyeglasses in the twelfth century not only made it possible to improve defective vision but suggested the idea that human beings need not accept as final neither the endowments of nature nor the ravages of time. Eyeglasses refuted the belief that anatomy is destiny by putting forward the idea that our minds as well as our bodies are improvable!*[Postman, 1985]

#### 7.2.2.1. Tools for learning

As stated previously, a tool for learning is “something that I use at some point that leaves me changed so I can do, by myself, what I could not do before.” An example of this would be map reading; after learning this skill, people can navigate their way through places that they had never seen before and estimate travel times for future trips.

A device/technology can be categorized as a tool for learning if it affords internalization of what was an external ability/function, if it existed previously at all. An example of this is reading and memorization. At one point in history, it was not unusual for some individuals to memorize enormously long epic poems or sacred texts, such as the Odyssey or the Koran; the development of reading (and writing) allowed anyone who had mastered this tool to reproduce the same feat.



**Figure 40 - Bicycle with Training Wheels — A Tool for learning**

Another good example is bicycle training wheels (**Figure 40**), which some parents provide for their children to help them learn balancing skills at an early age. These devices assist in learning the dynamic balance skills that they need to ride a standard two-wheeler. One of the attributes of these tools is that successful use of them leads to their relinquishment. Having successfully developed the skill, one no longer needs the device, and often the continued use of the technology impairs further progress. Imagine trying to ride a mountain bike in the hills with training wheels. Most often, the tool is used as “scaffolding” [Guzdial, 1994; Pea, 2004] and is dismantled, either gradually or altogether, as the user internalizes the function it aids doing.

One can create an operational definition for a TfLea so that classification can be more easily done. A technology or device can be classified as a TfLea based on the answers to the following questions:

- *Is your skill in the tools domain significantly greater (such that you no longer need the tool to perform meaningful work) as a result of using/experiencing this tool?*
- *Are you different as a result of using this tool?*
- *Is there a method of having this tool disappear (i.e., scaffolding)?*

At a minimum, the first two questions together must be true for the device to be considered a TfLea. TfLea systems typically involve some specialized resources that differentiate them from TfLiv devices, most of them being related to training issues. The instantiation of a TfLea *is* training; after the accomplished training the TfLea becomes part of the user. Often the decision to utilize a TfLea system is based on the hope that it will be powerful enough to effect a substantial enough change so that the user will have the same abilities as the “normal” population; an inappropriate application often leads to disappointment and abandonment of both the specific tool and that class of tools. This can lead to giving up on trying to solve a problem altogether, caused by disappointment when reach exceeds grasp.

#### 7.2.2.2. Tools for living

As stated earlier, a tool for living is “something I need to use to do what the ‘normal’ population does,” or to do what cannot be accomplished by means of training or skill-acquisition. Eyeglasses are a good example: as long as people wear their eyeglasses or contact lenses (excluding laser surgery), they can see as well as anyone else; when they don’t wear them, they cannot. No matter how hard they try to see better without them, they will never gain better sight merely by “learning” to do so. No matter how long or skillfully they use their glasses, their vision stays the same. Other examples of TfLivs are radio-telescopes and electron microscopes, external tools that enable a range of “visual” acuity that can never be obtained in a human body (at least in non-Kryptonian bodies). The tricycle in Figure 41 is functionally similar to the bicycle with training wheels, but is designed for elders who no longer have reliable balancing ability; this tool is not meant to be scaffolding to learn a skill, but rather a permanent aid to normal functioning.

A TfLiv device can therefore be defined as an external artifact that aids in affording abilities that exist in the “normal” human population; an alternate definition could be an external object that enables users to do what they will never be able to do on their own. TfLivs are typically specifically tailored to do *this* task for *this* person; for example, one person cannot wear someone else’s glasses. The need for TfLivs does not change over time, and they are rarely abandoned.



**Figure 41 - Elder Tricycle**

Similar to the operational definition for TfLeas, several questions can discriminate TfLivs:

- *Do you require **this** tool to accomplish **this** task (otherwise you cannot perform the task)?*
- *Does the tool rarely change? Is it rarely abandoned (in the sense of “graduating”)?*
- *Is it specifically tailored to you?*
- *Does your use of this tool typically not change over time?*

In contrast to TfLeas, TfLivs often require ongoing support, especially when they are computational devices. For example, a screen reader (a TfLiv that enables people who have sight or learning deficiencies to use a computer) needs to be aware of the changes in the “raw” documents it is trying to translate into spoken words. Note, however, that the settings specific to the user do not typically change over time. For applications that are more sophisticated, a TfLiv device may use distributed cognition techniques to effect its tasks; think of a cognitive prosthetic that delivers just the right information in the right fashion to an end user [Cole, 1997]. This ongoing need for robust behavior (what happens when the network or a server goes down and you are in a *very* unfamiliar part

of town?) and specific tailoring of the tool to the user provide many challenges for the assistive technology researcher.

Inappropriate, too extensive, or too early reliance on a TfLiv can cripple its existing potential. Providing a screen reader to a young person with dyslexia as a way of “mainstreaming” her may prevent the difficult but ultimately self-sufficient adaptations that might be possible. Indeed, the ill-considered or lazy application of TfLiv can lead to “learned helplessness,” leaving the user both inadequately prepared and psychologically crippled to engage with the world [Olsen, personal communication, 2000]. For all the necessary good that TfLivs provide—wheelchairs, prosthetic limbs, and sensory accommodations such as hearing aids and eyeglasses—there is a subtle danger to self and society inherent in TfLivs. TfLivs, being not human, do not impart the same depth of experience to the user that developing the skill from crude to refined accomplishment does. Doing a task with an automatic tool can remove us from the possibility of the “right livelihood”<sup>16</sup> type of internal growth as we develop the skill to get as good as the automatically produced artifact would already allow us to be.

#### 7.2.2.3. Design implications

Analyzing a project or proposed system for its TfLiv and TfLea qualities can assist in the process of design. For instance, if you were developing a new way of teaching reading of sign language to a specific population, your abstract, high-level design concerns would be quite different from the requirements for a glove that translated hand positions into words on a screen. Of course, attention should be paid to the elements of design in any system, but each type of system will require special care in certain areas. TfLivs, for instance need particular care for initial fit or customization, but less so with respect to dynamic adaptation of the tools interface to the user over time. Therefore, although you want the contact lens prescription to be exactly correct, you do not need to check it every month. Conversely, in the process of learning to program in SmallTalk with a TfLea tool [Guzdial, 1994], you want the device to check the students’ progress and adjust the level of support accordingly. TfLeas are

---

<sup>16</sup> Right livelihood refers here to the Buddhist notion of *samma ajivo*, where the process of working in the world causes a corresponding development in the worker, ethically, intellectually, and morally. This perspective sees one’s inner development directly connected with one’s daily work.



most often used in controlled environments (swimming schools, programming classes, driving schools), so there is less concern for robust behavior than for the design of TfLivs such as wheelchairs, augmentative and alternative communication devices for persons with cognitively disabilities, or prompting devices that aid in taking a bus trip.

To some extent, all human artifacts can be divided into two categories (see Table 11). Some artifacts seem to have both attributes, depending on use and expectations; interestingly, those artifacts for which the expectations for use differ from the actual use (such as hand calculators) have generated some degree of confusion and unsuccessful use history.

Let us look at some examples of these tools. A tricycle is an interesting example. From one perspective, it is a TfLiv—it allows a child to do what he/she could not do unaided, namely, roll around in driveways and on sidewalks while seated. It is also a TfLiv from the perspective of it not changing, and being specifically tailored to the user (tricycles have afforded any number of vertical, horizontal, and seat positioning adjustments to exactly fit the user's frame).

From another perspective, however, the tricycle is a TfLea, designed to be an entry into the more mature world of bicycles, a world that accommodates much more freedom of travel (to school, to the mall) and a much better fit to the environment (e.g., 21 gears, special mountain bike tires). Tricycles are certainly much easier to use in the sense that they do not require the balance and fine motor skills that a two-wheeler needs. However, *“If ease of use was the only valid criterion, people would stick to tricycles and never try bicycles”* [Engelbart, 1988]. The point is that the adoption of a new tool is driven by a combination of perceived ease of use, available affordances, and a match between needs and what it can accomplish. So the youngster aspires to graduate to a two-wheeler, like the big kids. Interestingly, this new tool, the bicycle, which is truly a TfLiv, is, for an aging population, becoming a tricycle again. As fine motor skills and balance abilities wane, many older persons are returning to tricycles. As with any TfLiv, they are appropriately fitted to their adult size and they allow seniors to ride where they might have found walking unaided problematic. So here is an example of a tool that, in

different contexts, is used in both senses, and more interestingly acts as a dynamic scaffolding, fading and expanding over time (over a long time), to accommodate changing contexts and needs.

#### 7.2.2.4. Ambiguities

Another interesting phenomenon is that the use of a TfLiv in a specific domain often drives out the skills (perhaps acquired using TfLeas), and in seeming compensation, another, “higher” skill blooms. This remapping of skills is analogous to the remapping of tasks that is a characteristic of implementing a distributed cognition approach. One perspective on the TfLiv concept is that it is an instance of distributed cognition (see section 3.3), and looking at distributed cognition as a TfLiv may afford new insights. Using a pocket calculator instead of pencil and paper can cause the atrophy of the earlier acquired skill of arithmetic calculation in the head, but at the same time the user is often acquiring higher mathematical abilities. The juxtaposition of freshman calculus classes and the wide availability of inexpensive calculators that can integrate and differentiate has a new generation of teachers worrying about these very issues (see Figure 42). There are several possible responses to this challenge:

- **Position 1:** Ignore the existence of the gadget; we are not interested in technology, but in important mathematical skills; ban handheld calculators from schools
- **Position 2:** Keep the curriculum the same, make children learn arithmetic, multiplication tables, long division, finding square roots by hand, and *after* they have it all mastered, allow the use of handheld calculators.
- **Position 3:** Create new calculators that make learning these skills more fun and create a deeper understanding of underlying concepts so that by using these handheld calculators frequently, the learners would acquire the skills and the knowledge and eventually become independent of the gadget.
- **Position 4:** Find new ways to distribute responsibilities between humans and machines such that humans do the qualitative reasoning, use estimation skills, and relate the mathematical result to the real world, and machines do the detailed quantitative computations

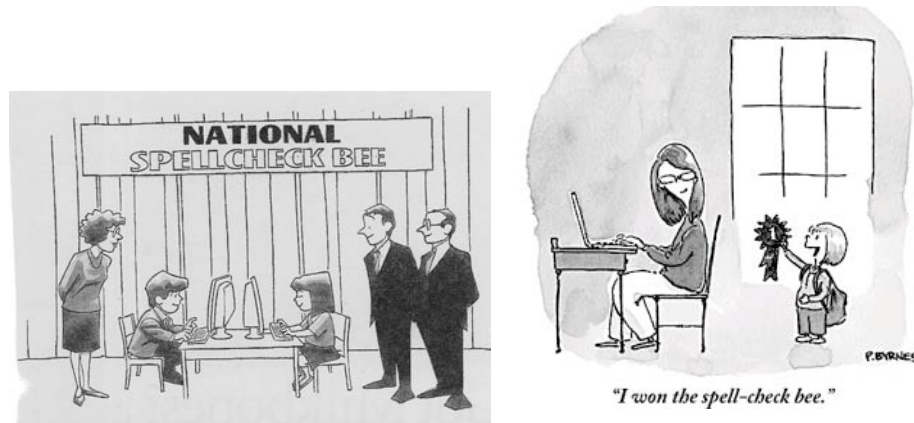


Figure 42 - A TfLea Used as a TfLiv?

What actually did happen was that the National Council of Teachers of Mathematics (NCTM) issued a position paper stating that their recommendation in response to calculator technologies is to dramatically transform the objectives and timing of the entire course of mathematics education [National Council of Teachers of Mathematics, 1989]. For example, in K-4 mathematics, a focus on long-division operations and paper-and-pencil fraction computation has been diminished, the availability of calculators is assumed, and attention is shifted to estimation activities and a focus on the meaning of operations and the selection of appropriate calculation methods [National Council of Teachers of Mathematics, 1989]. The NCTM standards go on to note that “*calculators, computers, courseware, and manipulative materials are necessary for good mathematics instruction; the teacher can no longer rely solely on a chalkboard, chalk, paper, pencils, and a text.*” Note here that “chalkboard, chalk, paper, and pencils” are all themselves TfLivs, which generated similar uproar when introduced into Athenian culture two thousand years ago [Norman, 1993] [Plato, 2003]. This shift from the detail of mathematical operation to focusing on the meaning of higher-level concepts is not uncommon with TfLiv displacement of internalized TfLea skills. An argument could be made that the widespread acquisition of reading and writing skills, which effected the end of the once common bardic feats of recitation, led to the much wider dissemination of culture in general.

Many tools and classes of tools exist that, depending on the user, the users’ needs, and the context, provide TfLea or TfLiv functionality. Spell checkers are seen as educational tools by many who use feedback to both autocorrect the document on which they are working and learn the proper spelling of

the word, and the scaffolding support in the word processor automatically retracts (fewer errors, therefore fewer spell checker corrections). Others see spell checkers as a tool they are wedded to for life—they have made the trade-off of intrusion into composition in exchange for not extending their knowledge of spelling nor touch-typing accuracy. Spell checkers can be used by the same person as a TfLiv, in English, and as a TfLea for learning French. MAPS, while being designed primarily as a TfLiv, has, with some script adaptation (see section 6.6.2 below) been successfully used as a TfLea.

#### 7.2.2.5. Thoughts about the tools and MAPS

What is it about the tools concept that gives better support to the design of AT based on distributed cognition? MAPS was primarily envisioned as a TfLiv, and thus much effort was spent configuring support for a tight fit among the client, script, task, and prompter. In realistic testing, MAPS was primarily used as a TfLiv, but one person in the study clearly used it somewhat as a TfLea. The caregiver redesigned the folding-clothes script after each run for the first four trials. Fortunately, the script was designed with “collapsing points” for each subtask that could be collapsed (see appendix 13.14). Therefore, each script must be saved with a separate name, but if the system were redesigned to better support TfLea, the expanded and collapsed versions of the same script would be that same script but with a parameter to allow the prompter to auto-size itself in use.

TfLivs can transform into TfLeas by “retracting” scaffolding [Pea, 2004]. As an example from the scenarios, in using MAPS to support employment, after several weeks, a young adult with cognitive disabilities could learn the fine detail involved in restocking the linen closets at the nursing home where she is working. Gradually, she has relied less and less on the MAPS-prompter, or at least the fine granularity of prompts it originally presented, to support her task. Her job coach noticed this improvement when she stopped in for her periodic evaluations, and created a shorter employment script by “collapsing” several detailed prompts, which the young adult with cognitive disabilities had internalized, into one cuing prompt. After the job coach gave the revised prompter back, she observed the young adult with cognitive disabilities doing the task just as well with the shortened script, and using a lot less skipping forward to get to the parts for which she really needed support.

Moreover, it is also possible for a TfLea to transform into a TfLiv through “dynamic extending” of scaffolding. For example, someone experiencing the onset of Alzheimer’s and starting to progressively forget little things may need an automatic extension of the external script support to retain his/her current level of independence. Accordingly, MAPS could be designed to allow implementation of extending and contraction scaffolding (see section 8.2.5).

How could the tools concept be extended into the AT world, and beyond that into the larger human-computer interface (HCI) world? If an AT device is primarily designed to be a TfLiv, the designer must focus more on getting the initial fit right and less on scaffolding issues. However, if it is primarily a TfLea, the design should focus on providing a mechanism for retracting and expanding scaffolding and the generic use of the technology.

#### 7.2.2.6. Conclusion

Design and implementation of tools for living are defined by their availability—what counts are the access characteristics (for example, how transparently the needed knowledge is represented and how readily it can be retrieved) [Perkins, 1993]. Motivation for use (and creation) of these tools is driven by an implicit calculation of how difficult it is to use and how much energy must be expended to gain access to the needed knowledge. For instance, it is more convenient to do simple arithmetic sums in my head; however, I would always use a calculator for formulas that involve trigonometry or logarithms. You may be able to calculate the square root of a number by hand, but the time it would take far outweighs the effort involved in using a ubiquitous hand calculator. Infrequent use and complexity of process drive these functions into artifacts. Certainly, you can memorize a text, but the time it takes to do so far outweighs the effort involved in purchasing the book. A more “physical” example is the use of Velcro® fastenings on shoes. A person with fine motor skill deficiency due to cerebral palsy might be able to tie shoelaces with a tremendous amount of exertion and time, or just use Velcro. Nickerson, in discussing this, says:

I suspect there is a law of memory according to which we tend to store information in our heads that we need relatively frequently and that is relatively difficult to access from an external store or to infer. Stated slightly differently, we are the more

likely to commit something to memory the more important it is to us and the greater the effort involved in accessing it from other repositories. [Nickerson, 1993]

Both tools for living and tools for learning share the attributes of distributed cognition; the TfLiv is obvious, but the connection with TfLea is that the skill or ability that is acquired as a TfLea is culturally mediated—it comes from human culture and is part of the web of distributed cognition that makes us human [Hollan et al., 2001]. These tools disappear into the background of day-to-day life; many of these objects have become “mythic,” as Barthes [Barthes, 1972] uses this term, in that they have become so deeply a part of our consciousness that we do not notice them. Turned from artifact into nature, they are invisible, unremarkable aspects of our world. Moreover, as such tools become invisible, it becomes more difficult to see them as bearing intelligence, and we see the intelligence “residing” in the individual mind using the tools [Pea, 1993]. Think about reading, driving, reading a map, and riding a bicycle.

The tools both for living and for learning are artifacts that leverage our existing abilities to act and understand in ever-larger spheres of independence. The ability to perform arithmetic, a TfLea, extends our ability to predict and compensate for future events and to more effectively utilize current tools and resources. TfLivs are independence agents when TfLeas cannot be applied. A wheelchair, a TfLiv, provides mobility when walking is no longer possible. Of course, there are always trade-offs, such as overreliance on calculators, which can fade out arithmetic ability. In an interview, a dyslexia researcher [Olsen, personal communication, 2000] pointed out that for some persons with dyslexia, screen readers are deleterious (i.e., the user does not even attempt to learn compensatory skills that would have allowed reading without assistance), whereas for others, they do not inhibit development (as there can be no development in that respect). TfLivs are like distributed cognition (perhaps distributed cognition is the premier abstraction of TfLiv) in that they remap the task [Norman, 1990]. In the example of writing down information, a TfLiv (in contrast to *learning* to read, a TfLea), the task becomes creating documents, not performing mnemonic “tricks” [Yates, 1966].

### 7.3. Symmetry of Ignorance

The use of the principle of symmetry of ignorance naturally leads to the study of stakeholders in the MAPS socio-technical environment. One of the results of this study was to see the tool designer as having a dynamic role in the evolution of the system design. It was clear the design of MAPS was not, at this level (in contrast to considering only caregivers and persons with cognitive disabilities), an event but rather a process. The system is not just designed and given to the user, but periodic redesign of the system was seen and seemed potentially to be an ongoing, if episodic, part of the evolution of the MAPS environment. This was not obvious at the time of the initial design considerations, but became clearer in the process of studying the metadesign possibilities in this domain, using the principle of the symmetry of ignorance. The details of MAPS from the perspective of symmetry of ignorance led to looking at the roles involved in the MAPS socio-technical environment not as discrete separate entities (as a standard requirements analysis would do [Pfleeger, 2001]) but as participants in a process.

Thus, there are two sets of members to symmetry of ignorance in the MAPS system: (1) caregivers and clients and (2) tool (AT) designers and caregivers. Thus, there is a further iteration of the notion of symmetry of ignorance at a higher design level: the caregiver and the script-design-environment together have the solution to an adoptable script-design-environment, but neither has all the relevant skills and knowledge to construct a script-design-environment on their own. From this division of knowledge that together could provide support for independence comes the concept of *dual-user interfaces*. The first interface is the MAPS prompter for the person with cognitive disabilities task support; the second is the MAPS script-design-environment that the caregiver uses to create scripts and to update them with the data of the completed script logs and observations.

#### 7.3.1. Stakeholders

Wicked problems are most often “solved” (here the notion of satisficing emerges) through group efforts. Further, the solutions are not true or false, but instead are better, worse, or good enough. Task support through computationally based multimedia prompting is one such problem. The starting-off point for a designed solution is to do a stakeholder analysis of the problem space [Overseas

Development Administration, 1995]. Several types of stakeholders are involved in prompted task-support:

1. Key stakeholders – Those who can significantly influence or who are key to the success of the activity (in this case caregivers and clients);
2. Primary stakeholders – Persons who are directly affected by the solution; in this case, parents, employers, group home staff;
3. Secondary stakeholders – All others with an interest in the activity; in this case, members of state and federal organizations who concern themselves with AT, insurance companies, and HIPAA bureaucrats.

Having identified the stakeholders, the designer then can proceed with a better assurance that the system can be adopted as the solution that must satisfy each class of stakeholder. Additionally, this analysis identifies the (sometimes orthogonal) requirements and identifies the symmetry of ignorance members of the problem space.

**Table 12 - Symmetries of Ignorance**

<b>When</b>	<b>Domain Expert</b>	<b>Technology Expert</b>	<b>Notes</b>
<b>Pre-tool design time</b>	AT professionals, special ed teachers	Computer scientist, programmer	Typically done once – produces studies
<b>Tool design time</b>	Users, communicating via the results of prompting studies	Computer scientist, programmer	Typically repeated only in yearly intervals – produces script-design-environment
<b>Script design time</b>	Person with cognitive disabilities	Caregivers	Initially done once and updated after usage – produces scripts
<b>Script use time</b>	Caregivers	Person with cognitive disabilities	Produces script use logs and observed behavior

### **7.3.2. Domains of Expertise**

AT designers and caregivers (caregivers being one-half of the complete end-user persona) have reciprocal and symmetrical domains of expertise that together support the design and implementation of a script-design-environment. The technical skills that are required (software architecture to coding to hardware) are typically absent in caregivers, but can be supplied by the AT designer, and the



detailed knowledge of tasks, task segmentation, and the specifics of individual persons with cognitive disabilities can be supplied by caregivers, thus completing the required skill sets for script-design-environment creation. Similarly, but with a somewhat lesser degree of difficulty, the AT designer can bring the same skills as mentioned above, and the client (a person with cognitive disabilities) brings his abilities and broad user interface requirements to the task of creating an effective handheld prompter.

The caregivers and clients that comprise the dyad that is the end user each have symmetrical domains of knowledge that together comprise the single persona for the metadesign process of MAPS task support. The caregiver provides the ability to choose and segment the task for which MAPS will instantiate task support, and the client provides the internal scripts and proclivities in the use of the MAPS scripts that allow the caregiver to tweak the scripts to fit the changing exigencies of the changing details of user and task. The design of the tool (MAPS) had several iterations, based on the feedback of the caregivers, whose experience at script design time exposed new needs for the tool's functionality. One example of this was the wav sorting and insertion tool (see Figure 30 and section 5.4.2.1) that was built into the MAPS script-design-environment at the request of the first family dyad (see section 6.6.1). Table 12 shows a summary of the relationships inherent in symmetry of ignorance in designing AT for persons with cognitive disabilities.

#### **7.4. Metadesign**

By using a metadesign design methodology with respect to the MAPS socio-technical environment, the unique nature of the design process and the flow of knowledge and information between the caregiver and the person with cognitive disabilities becomes clearer. This flow is mediated by scripts (from caregiver to person with cognitive disabilities) and logs and observations (from person with cognitive disabilities to caregiver), and leads to supporting the dynamic growth of specific scripts by the MAPS system. This dynamic growth manifests as re-writing scripts to expand a script (build more task support into a script) and contract a script (removing the internalized parts of scripts as the user makes them into internal scripts), the re-use of scripts (the script navigation segment that several of the caregivers shared), and the use of scripts as templates (as was done for visiting the library and

shopping for groceries by several of the caregivers). Details of examples of these four processes: scaffolding contraction and expansion, script reuse, and template use are discussed in section 6.6.

Metadesign is a relatively new concept, but not really such a new process. Perspectives on metadesign come from the worlds of collaborative art [Giaccardi, 2004] and end-user-programming [Fischer & Giaccardi, 2006]. Notwithstanding its relative newness, the existing literature provided enough similarity between described instances of the metadesign approach and MAPS design problem space to provide a structure to measure against and build with.

**Table 13 - Overview of Metadesign Attributes**

Stage	Initiator	Recipient	Artifact
<b>Tool-design time</b>	Tool designer	Caregiver	The script-design-environment
<b>Artifact-design time</b>	Caregiver	Person with cognitive disabilities	Scripts and redesigned scripts
<b>Artifact-use time</b>	Person with cognitive disabilities	Caregiver	Script use logs and observed behavior

#### 7.4.1. Definition – Design Over Time

Metadesign refers, in this domain, to the process, framework, and problems inherent in designing a tool that will be used by one population (caregivers) at script-design time (in contrast to script-design-environment-design time) for users with cognitive disabilities at use-time (see Table 13) [Giaccardi & Fischer, 2005]. Because the problem domain that MAPS addresses is classically wicked, that is, the solution cannot be specified at tool design time and the solution needs to satisfy *all* stakeholders [Rittel & Webber, 1984], the use of metadesign principles is particularly appropriate. Metadesign problems are never solved, their goal is one of ongoing *satisficing* of the current configuration of the problem space (i.e., the solution is good-or-bad, not true-or-false) [Simon, 1984].

Metadesign traditionally views two time domains as relevant to the metadesign process: tool-design time and artifact-use time. When approached from a collaborative art creation perspective, the

time often collapses into one process, and the creation of the tool and the use become one process.

Here, however, there are three relevant time domains:

- Tool-design time—the time when the initial MAPS script design-environment (and prompter) is constructed;
- Artifact-design time—the time when the caregiver designs and assembles a task-support script for use by a particular client on a specific task (see Figure 43);
- Artifact-use time—the time that *this* client uses *this* script to do *this* task (see Figure 44).



Figure 43 - MAPS at Script-Design Time

The fundamental design requirements of devices for a “universe of one” (see section 2.1.3) make it impossible to design scripts at tool-design time. The script must be changeable by the user at script-design time; the inability to do so, in fact, caused the abandonment of the Visions (see section 4.2) system in an installation [B J Sample, personal communication, 2000]. So the design of the MAPS system has to be grounded in the basic assumption that future users and user scripts cannot be

completely anticipated at initial (tool) design time, when the system is developed [Fischer & Giaccardi, 2006] ,thus the need for the metadesign perspective. The disparity between the powerful tools of the computer programmer, which are too complex for the user with domain expertise to master, and an inflexible tool designed by using these tools to precisely fit the needs of a user at *this* time and in *this* context forces a successful designer to apply a metadesign approach to design high-functioning AT.

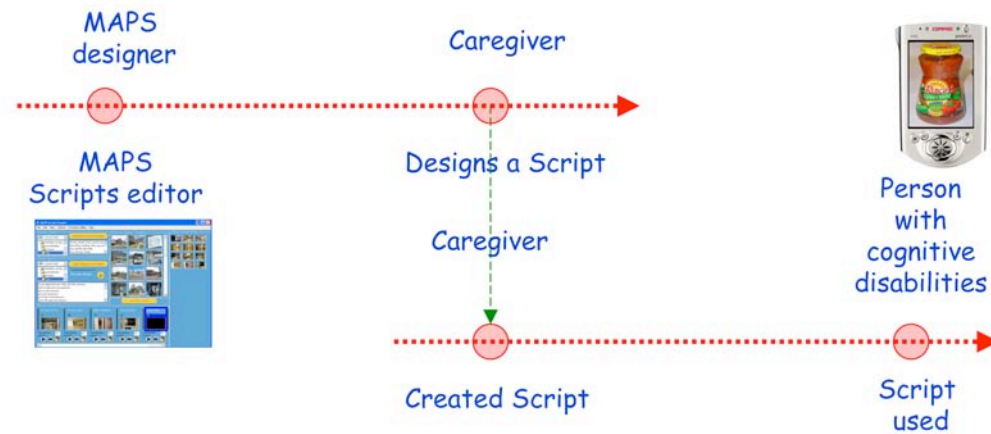


Script Use Time

**Figure 44 - MAPS Prompter at Script-Use Time**

Metadesign has to do with situatedness in that the use of the final artifact is strongly dependent on the new needs of the user at use time, and with embeddedness in that the computer artifact is tightly integrated into the user's daily life [Fischer & Giaccardi, 2006] . Therefore, the MAPS-design-environment needs to reflect the changed environment at actual use, and the perspective of the whole system at use-time of the artifact and the user. Similarly, the MAPS script-design-environment must be, as delivered to the caregiver, underbuilt [Brand, 1995] in the sense of providing the affordances for creating scripts but not the exact content. This underbuilt quality extends to the use of existing,

successful scripts, either as templates or from other similar users, which will be transformed into this user's script by replacing the image, verbal prompt, and error information.



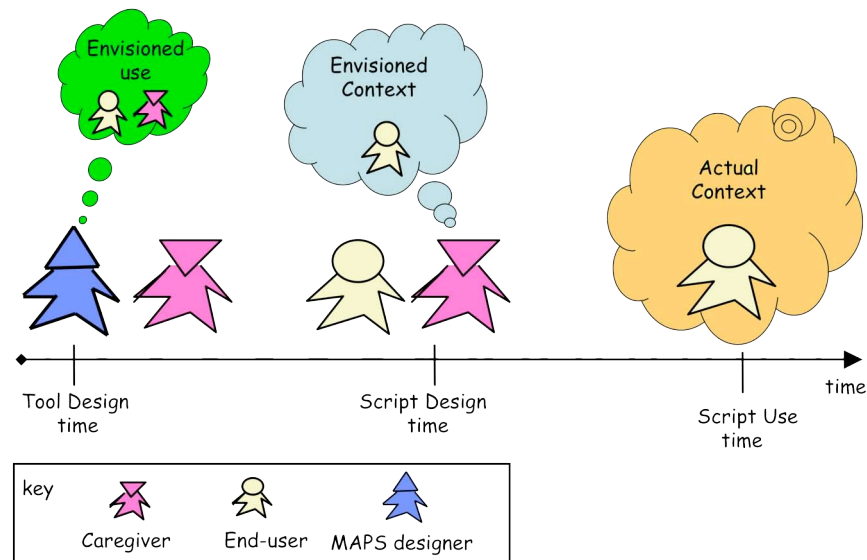
**Figure 45 – Metadesign – Design Over Time and Role**

A metadesign analysis of the MAPS system requires knowledge, at tool design time, of how much domain knowledge (about prompting for persons with cognitive disability) was needed to encode into the design and how much the caregiver needed to know at script design time about the use of the script, and its context (see Figure 45). The MAPS script-design-environment needed to be specifically tailored enough so that the task of making scripts was made tractable for non-programming caregivers (avoiding the Turing tar pit in which “everything is possible but nothing of interest is easy” [Perlis, 1982]), but not so tailored that many different types of scripts could not be designed (i.e., the script-design-environment would create scripts easily only for bus travel).

#### **7.4.2. One End User, Two Roles, Three Persons**

In its original context, metadesign’s users were the same as the designers, resulting in two roles: the meta-tool designer (a programmer) and the designer-user. However, in the world of assistive technology, where the concern is not so much an evolving tool as changing configurations and adding content, there are three stakeholders: the meta-tool designer (the tool’s initial programmer), the designer (caregiver), and the end user (client). Due to the intimate relationship and information flows between the caregiver and the end user, the two can be seen and worked with as one “persona.” The

tool designer must know the domain in which the tool is to be used—both the actor (clients) and the context (tasks)—well enough to provide *underbuilt* [Brand, 1995] support for the tool user to create good-enough scripts. The caregiver, the user of the system to create scripts, must know the script-design-environment and the client as well as possible tasks and environment. The client must understand how to use the prompter and have an inventory of internal scripts sufficient to support the requirements of the script’s task-support. Figure 46 demonstrates the environment that must go between the roles in the metadesign process.



**Figure 46 - Metadesign, Plans, and Situated Action**

Interestingly, in a commercial offering of a script-design-environment, the AbleLink Corporation provides a similar role breakdown. Documentation bundled with the AIMS development package (see section 4.2) discusses these three roles: *content user* = person with cognitive disabilities; *content developer* = caregiver, and *technology developer* = tool designer [Davies et al., 2005]. So we have one end user comprising two roles: the client role and the caregiver role. The whole is composed of three persons: client, caregiver, and tool designer. The next three subsections of this chapter detail the qualities and relations between the three roles in the MAPS metadesign environment.

#### 7.4.2.1. Person with cognitive disabilities

The end user whose interface is the handheld MAPS prompter uses the MAPS system to supplement deficiencies in executive and mnemonic ability caused by cognitive disabilities. What end users contribute to the metadesign system is their own set of internal scripts (see section 7.2). Their internal scripts will be cued by the script, and together they form the actions that, when done, constitute a task successfully done. What they provide for the caregiver, in the caregiver-script-designer role, is an inventory of internal scripts, a set of abilities and deficits, and interface preferences. The interface preferences consist of image, verbiage, and voice preferences. In the realistic testing (see section 6.6), a person's voice records the prompts. In some cases, the person whose voice is recorded makes a difference in whether the script is successful. In the process of using MAPS and with respect to the script artifact, the person with cognitive disabilities provides feedback to the script designer (the caregiver) in the form of logs of script use (see section 5.2.2 and appendix 13.9). As result of the log of use and personal interaction with the caregiver, the caregiver, in the script-designer role, may expand the scaffolding of the task-support script by adding details to the script (in the case of breakdowns) or contract the scaffolding by replacing multiple steps with a single exterior-script prompt (see section 6.6.2).

#### 7.4.2.2. Caregiver

The caregiver, in the role of script designer, brings to the metadesign system knowledge of the person with cognitive disabilities, knowledge of the tasks they want to have supported, and knowledge of the MAPS script-design-environment on the PC. The caregiver provides to the client (the end user with cognitive disabilities) custom tailored scripts, fitted with respect to the client's interface preferences, abilities, and needs. Based on information and experience on the client's use of scripts (logs and observations), the caregiver may expand or collapse the script to make a better fit (see section 8.2.5). Script-designers could, in the future, interact with the script-using community by

reseeding<sup>17</sup> [Fischer & Ostwald, 2002] the repository of stored template scripts. The caregiver can also expand the functionality of the scripts for the client by adding annotations to a script to support context-sensitive and self-repairing scripts (see section 8.3).

#### 7.4.2.3. Designer

The tool designer needs to know the actors: the caregiver who may be barely computer literate, the world of persons with cognitive disabilities, and their possible needs in terms of affordances. The tool designer needs to be well grounded in the domain of AT design including existing AT systems and devices, existing use of prompts for training, the environment in which the prompter will be used, and understanding the use of affordances in this domain. To do a satisficing job with the design and implementation of the script-design-environment, the tool designer starts with samples of properly fitted scripts and works backward. Beyond that, the designer must accommodate the possible annotation of scripts that might enable context-sensitive prompting (see section 8.3). This is done to ensure that the designer (caregiver) is able to immediately create useful scripts, and have a tool that can accommodate complex, real-world exigencies.

The designer brings to the metadesign process a technical expertise that includes computers and other technologies that support distributed cognition. The designer must have expertise in the causes and details of the process of AT adoption and abandonment, and needs to gain expertise with the world of persons with cognitive disabilities and caregivers. This expertise may be obtained through ethnographic studies of the day-to-day life of these populations. The designer provides to the caregiver and client the script-design-environment and handheld prompter application. The script-design-environment must be underbuilt enough to accommodate the specifics of various clients and tasks but have enough and appropriate support so that the caregiver could, with little training, immediately create and install prompter scripts that can be used in the real world.

---

<sup>17</sup> Reseeding: a coordinated effort between MAPS tool designers and caregivers to collect exiting scripts and provide them to the MAPS community as templates; reseeding is a periodic and formal restructuring of task support content.



The designer supports the evolution of the system by providing template scripts for the script-design-environment and faculties to reseed<sup>18</sup> the repository of templates with successful scripts periodically. The designer also needs to provide the ability to attach annotations to the scripts so that as the prompting/scripting environment becomes *context aware* [Dey et al., 2001], scripts can be self-correcting and appropriately bring in caregiver support when needed.

### 7.4.3. Thoughts About Dual User Interfaces

When does a configuration issue become a dual interface requirement? If configuration and end-user programming defined an axis, at one extreme would be the C language, and at the other would be a light switch. Configuring takes something and *adjusts* it; programming *makes* something out of the parts. So from that perspective, a script-design-environment is not clearly one or the other, ignoring, for this example, the separate roles of caregiver and client and calling them one person. In (computer) programming, a (computer) programmer makes things through which a user can make choices; at a minimum, a simple on/off functionality allows the choice of on or off; below that, the artifact is not a program, it is a fact or a “thing.” Configuration would allow deciding what switching on and off would do, but not the bare fact that it has an on/off switch.

From that perspective, “linear” scripts are a result of configuration, and multiscripts (see section 5.1.4.2) are programming tools. Where a linear script fades into end-user programming is in determining how many steps the script will be as well as the notion of rewriting scripts as a result of observed or logged script usage. Another place that it is easy to see where MAPS approaches end-user programming is in the possible error trapping and correction script annotation (see section 8.3). The benefit of determining whether the MAPS system is an instance of end-user programming or not is that there is a mature set of knowledge about the principles of programming languages (see section 8.3).

---

<sup>18</sup> Currently, seeding is done by hand by the tool designer.

#### 7.4.3.1. Challenges in development

The first challenge in doing metadesign is that the designed artifact cannot be properly evaluated as a success or failure independent of actual use in the real world with real users and tasks. Therefore, at a minimum, a second round of design after the first implementation should be anticipated. Prototypes must include the whole of the application, not just the parts that may be tested separately in controlled conditions. Second, design of the script-design-environment involves high risk because the end-user dyad must have a good user experience from the first time it is used.

#### 7.4.3.2. Challenges in application use

The biggest challenge in implementing this metadesign, design-over-time, system is properly implementing the loop from the client end user to the caregiver script designer. The design needs to formally incorporate the feedback loop for how well the script fits this user and task and to present to the script designer just the information that would support script redesign or scaffolding collapse or expansion. This is driven by evaluation of both *this* task performance and also this one in the context of all performances of this task, so the design needs to have a way to extract both relative and absolute performance information. Further, the information has to be processed in a fashion that supports non-computer-savvy caregivers to change the script appropriately. This functionality was not incorporated into the MAPS system as of this study (see section 8), and the caregivers who did modify scripts during the realistic trials did so on the basis of informal observations of the script in use (see section 6.6).

### 7.5. Conclusion of Contributions

The canon of HCI has provided the concepts upon which MAPS was built, but in order to use them, they had to be tailored for this special population. Are the insights that were gained by this process transferable back into the larger matrix from which they came? Certainly the tools concept, extending distributed cognition, is easy to see as usable in other contexts. User fit and scaffolding, the importance of initial experience being a success, and the support for training—these trade-offs are easier to define with the perspective of the two tools.

Similarly, the extension of metadesign to multiple persons and roles as well as across time allows the designer to anticipate odd kinds of information flow in a metadesign environment, as well as multiple players. Because metadesign is a relatively younger topic, perhaps the issue here is not what this research can give back to it, but rather that this is part of the process of defining what metadesign is in the first place. The specifics of the differences with the “official” metadesign model are that multiple persons in the role of user illumine the feedback process of the changing metadesign tool.

The unique way that the design of MAPS used the concept of symmetry of ignorance, in cascading pairs (see Table 12) can provide insight into complex collaborative group processes. These cascading pairs—script-design-environment designer and caregiver, caregiver and client, and client and caregiver—allow the division of roles and the highlighting of dataflow that may aid in understanding similarly complex systems.

In demonstrating this system to an interested fellow CHI’06<sup>19</sup> attendee, the comment was made that this was not different from a slide show and that there are any number of slide show programs in the world, so what was the difference between MAPS and one of those programs? Also, the challenge was made several times that there was no difference between MAPS and instructions and pictures on a set of cards (see Figure 47)—how is MAPS not just high-tech gift-wrapping<sup>20</sup> [Fischer, 1998] with a set of prompting cards? These are valid questions and deserve some consideration.

First, let us discuss the slide show argument. MAPS is different from a simple slide show with sound in two ways: affordances and persistence. The design of the script-design-environment has been driven by a requirement that caregivers with low PC literacy be able to make satisficing scripts immediately, without any learning curve. In other words, the user experience cannot be “the first one kind of works, but as you continue to use the script-design-environment you will eventually make a script that is actually useful.” The application must be capable of giving the caregiver multiple ways to see the script: an overview of the whole script (the multiple-prompt view on the right side of the script-

---

<sup>19</sup> CHI’06 is the 2006 annual Conference on Human Factors in Computing Systems.

<sup>20</sup> Gift-wrapping here refers to pouring the old content into a new media without consideration for either the problem to be solved nor the affordances of the new media. Naive use of new technology is regarded as a “silver bullet.”

design-environment; see Figure 2), a high visibility view of the portion of the script currently being worked on (the filmstrip five-slide segment of the script on the bottom of the script-design-environment; see Figure 2), a single-prompt view in real size (the prompt preview panel; see Figure 14), and finally a preview of the script in action as the client will see it (the prompt preview panel with the script review control activated; see Figure 25). The other part of the argument is that the MAPS system supports saving scripts and allows modifying the script and saving it as a different script as well as using parts of one script in assembling another. The ability to reuse scripts provides the caregiver with templates of successful scripts used by other persons with cognitive disabilities. Template scripts are available from the MAPS Internet-based database server after identifying items removed (items made anonymous for privacy). As clients' needs change the persistence of scripts allows their expansion and contraction to accommodate the skills and needs of the clients, thus keeping them in the zone between boredom and anxiety (see section 1.1).

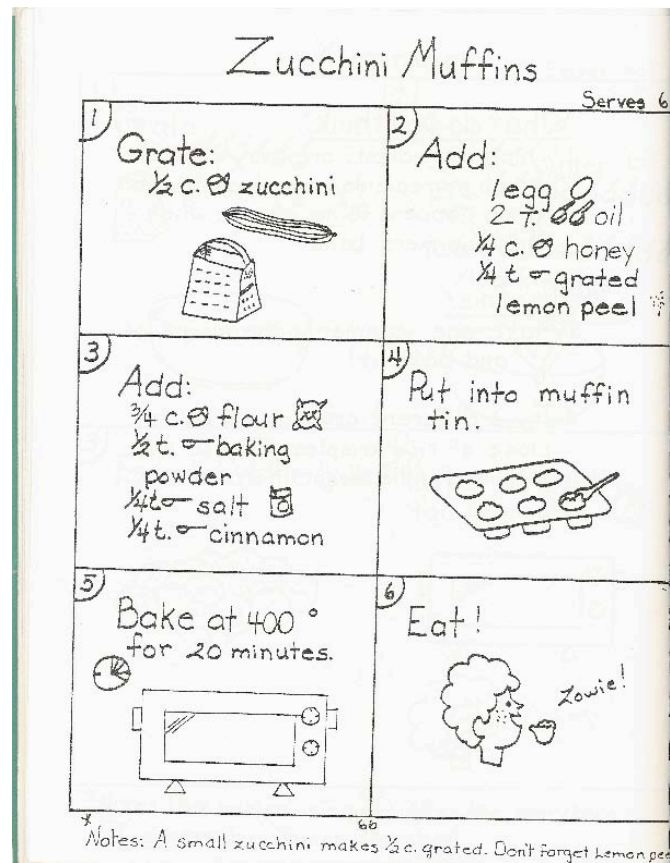


Figure 47 - Prompts on Cards

As for the question of whether MAPS is not just gift-wrapping of a paper-based prompting system, this question is similar to but different and deeper than the slide show one. MAPS is more than gift-wrapping in that the affordances of the multimedia presentation provide support in ways that a passive presentation cannot. One of the things we discovered in the field trials is that the voice is important; perhaps for some young adults with cognitive disabilities, the voice counts as much as the words and images (see section 6.7.1). Like the slide show argument, persistence counts; contracting and expanding scripts and being able to return to the original script is not do-able without an addition to the paper system. Multimedia is “hot” in McLuhanesque [McLuhan, 1964] terms—it’s a “push” rather than the “pull” that is required of the client with paper-based prompting. This engagement may make the difference in successfully doing a step that is not pleasing to the client (versus paper’s pull requirement). With proper fitting the active task support of MAPS avoids the two failure modes of boredom and anxiety (see section 1.1 and Figure 4). Finally, a set of paper prompts has no ability to support script modification without an external mechanism to keep track of the use of the prompts in attempting a task, whereas the MAPS system has that ability.

Finally, the question needs to be asked: how is design for AT specific to AT or is it just an example of good design practices? It is similar in that all good design practice ensures that the fit between the user and the artifact satisfies the user(s). However, the major difference between good HCI design and good AT design is that the system needs to be a success right out of box—the prompter needs to be as “toaster-like” as possible; for example, care had to be taken to block double clicking and the like. Beyond system design are several content requirements, such as the correct prompting voice and the specifics of metadesign, as mentioned above. The aspects for which the requirement and technique are virtually the same are the need for participatory design, underbuilding for metadesign environments, provisions for script reuse, providing video help, and the use of existing computer skills and metaphors. In the end, the dissimilarities involve emphasis rather than being qualitative.

## 8. Future Work

There are many intriguing possible paths to tasks that answer questions that the development and evaluation of MAPS brought up and that could use the MAPS system as a platform to further explore task support for persons with cognitive disabilities and their support communities. Each of the following sections was derived from specific breakdowns in script creation and use or specific requests made in the participant observation portion of the MAPS realistic testing process. The proposal to automate the script-critiquing process came from analysis of the process of creating new scripts, which entailed filling out a script template form and having it examined by AT experts for errors and omissions. Similarly, the suggested work on selecting and retrieving script templates and reusable sub-scripts as well as images and recordings for prompts emerged from caregivers' frustrations in accessing and choosing components of scripts. Observing and documenting the caregivers creating support tools for script scaffolding, retracting, and extension led to a discussion with the caregivers, documented in section 6.6.2, about how this process could be computationally supported. Some work was done in this research to explore the relationship between the image presented on the handheld and the accuracy of following the prompt instructions. Some suggested extensions to the "image experiment" are discussed.

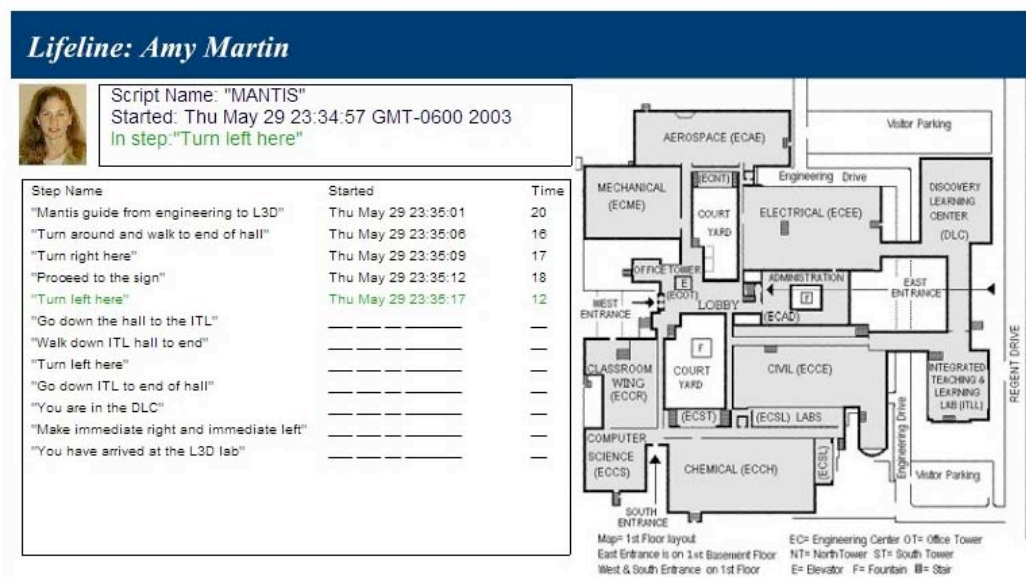


Figure 48 - Lifeline Caregivers Interface

## 8.1. Lifeline

During the development phase of MAPS, another CLever project, Lifeline [Gorman, 2005], was being designed and implemented. Lifeline's goal was to act as an information broker and conduit to caregivers and to information about the environment. As originally envisioned, MAPS scripts would have error-trapping information encoded into the script, which would be run on a prompter that was also running a Lifeline applet; as the script was run, information about encoded expectations of performance and context were to be relayed to the Lifeline server, and, depending on the situation, Lifeline could activate actions appropriate to the situation (see Figure 18 for a diagram of the architecture and information flow). A demo version of this architecture, with a web-based display (see Figure 48) for caregivers' use, updated in real time as the client stepped through the script, and the use of time as the triggering test on a particular script step was developed. In this case, if the client paused the script at a particular step for too long (with the appropriate length of step encoded by the caregiver; see Figure 49) the display would update in red letters and an SMS cell phone message was sent to selected cell phones containing the "error" data, script name, step title, and time of event. This demonstration system showed much promise in that it not only informed the caregiver via a web-based interface (thus accessible from any browser with the right password), but it also *actively* reached out to the caregiver if needed. This system, called *active distributed task support*, was the subject of several papers [Carmien et al., 2005a; Carmien et al., 2005b]. The second section of future work presented here describes in some detail the script-design-environment's annotation system and the design rationale and architecture of an active distributed task support system.



Figure 49 - MAPS Script-Design-Environment Interface for MAPS/Lifeline Prototype

## 8.2. Extending MAPS

### 8.2.1. Script Critiquing

Inspired by computational critiquing systems [Fischer et al., 1998] based on Schön's framework of reflection-in-action [Schön, 1983], several additions to the MAPS script-design-environment could be added to opportunistically capture and use information already existing in the script structure.. These additions specifically help the caregiver by pointing out possible poor script design through providing an avenue for system backtalk to the caregiver [Fischer et al., 1990]. One possible heuristic involves capturing the number of verbs in a cue and reflecting back to the caregiver that that prompt may be too complex for the user. To do this, MAPS needs to divide the client population into ranges of linguistic functionality, and the MAPS script-design-environment must be able to count verbs in a given prompt by using a list of common English verbs as a basis. When creating a prompt by placing an image and a recorded voice, because the file name for the recorded voice consists of the actual words in the recording, it is a trivial matter to count the number of verbs in the proposed script. Given the user's functionality category and the number of verbs in the proposed prompt, MAPS could flag a possible problem when the verb count exceeds the "acceptable" verb count for that category of user. Then, as the script is created, MAPS could analyze the elements of the script and provide a dialog popup box to expose a possible problem [Fischer et al., 1998], explain why it is a potential problem, and perhaps provide alternatives that the caregiver could take to create a more useful script. As a simple example, a prompt with six verbs for a user with an IQ of 50 would probably not be a good match.

Several other easy catches may be based on the structure of the prompts and the script. For instance, MAPS could count the number of steps in a script, and if falls either below or above a present granularity limit for persons with cognitive disabilities of a given category, the script-design-environment could present a critiquing window to the caregiver. Another possible error for the script-design-environment to flag might be a noun denoting a common household object, such as a specific measuring cup, in the later part of a script without having introduced it in an earlier prompt (and



perhaps checking for verbs such as “get” or “put”). These examples are instances of a design philosophy of getting “low hanging fruit” (see section 5.3.2).

### **8.2.2. MAPS Script Templates on Server – How to Use, How to Select**

The MAPS network server contains more than 170 script templates. In the current version of MAPS, there is no easy way to connect and disconnect from the network database and the local computer database (see Figure 23). Also, to use the templates, the caregiver must connect to the network database, load the script into the script-design-environment, disconnect from the network and connect to the local database, and save the script locally. With a wizard-like interface designed to do this automatically, the start of using the templates would contain an easy way to select a template script based on both the goal of the script (e.g., library patron) as well as on some functional characteristics of the client type (see section 8.2.1, discussing user functional type). Additionally, a facility is needed for submitting successful scripts back into the script template repository server, as well as making them anonymous before submission by replacing all step images with a generic placeholder. Currently, the only way to do this is for a super user in a database administrator role to do it by hand.

### **8.2.3. Refinement of Image and Voice File Storage and Access**

Currently, MAPS uses the file system to store the binary files of images and voice recordings, but with the goal of reuse and ease of retrieval, any reasonable growth of the files would make this system unmanageably large. A search system for images and prompts will need to be implemented, perhaps incorporating the ability to share the sounds and images from other caregivers. This is a difficult problem because it implies development of a meta-descriptor for each file, audio or image, that allows searching and sorting, and a user interface to support caregivers doing this.

Another request that several caregivers made of the MAPS script-design-environment is to provide an easy way to access “generic” prompt elements. Over the course of four dyads, general prompts instructing the client in MAPS-prompter control use and the rudiments of script usage often had to be repeated (i.e., “the button in the middle of the prompter allows you to replay the last words you heard”). The caregivers requested a control that would change the image and wav file choice boxes

to point at the generic prompt binaries, and a second click of the control would return the script-design-environment to the directories that were originally being used.

#### **8.2.4. Re-usable Script Segments**

From the beginning scenarios, MAPS was portrayed as having the ability to keep and reuse segments of scripts. For example, suppose that a dyad (caregiver/person with cognitive disabilities) was using MAPS to support bus trips. The segment of the script that guides the client to get from his/her house to the nearest bus stop would be the same across every bus travel script, as would be the segment related to travel from the bus stop to home at the end of the script. While it is possible to use existing scripts, deleting the steps that are not pertinent and replacing them with the ones that fit the new task, MAPS does not support inserting or appending scripts into scripts being edited. To do this would require some additions to the script-design-environment code and several small changes to the underlying database schema. Additionally, it would involve providing search capabilities across existing script segments and tying each segment to a single user, or making the segment usable across the spectrum of users.

#### **8.2.5. MAPS and Dynamic Scaffolding**

An additional structural functional addition concerns the notion of scaffolding and learning. As the MAPS users (with cognitive disabilities) use a particular script, they may memorize segments of the script and be able to decide to execute the memorized sequences on their own, without external prompting. This acquisition of mnemonic and executive functionality can be accommodated by implementing the notion of retracting (and extending) scaffolding by the MAPS script-design-environment [Davies & Miyake, 2004]. Because this is driven by both evaluation of *this* task performance as well as this one in the context of all performances of this task, the design needs to have a way to extract both relative and absolute performance information. Further, the information has to be processed in a fashion that supports non-computer-savvy caregivers to change the script appropriately. This functionality was not incorporated into the MAPS system as of this study. Appendix 13.14 illustrates how a caregiver in the realistic trials created a script to accommodate later collapsing of script scaffolding. The rightmost column in the table in appendix 13.14 indicates whether the steps

below it could be collapsed into the summative step above. In this case, the caregiver had to know that this might be a good script to collapse (61 steps to fold laundry), and to prepare to have the external triggers for the newly acquired internal scripts in place in the script (e.g., “Now let’s fold T-shirts”). When she had determined, by observation of the client doing the script, that all of this was in place, she collapsed the subsidiary steps into the external script cue of folding T-shirts.

The script-design-environment reads the logs of the particular scripts—logs that are created by the MAPS-prompter at script run time (see appendix 13.9)—recording the timing and sequence of the prompts as run by that user. The logs can be obtained at synch-time (when the next script is loaded into the MAPS-prompter) and analyzed. By using the analyzed logs as a basis, the MAPS script-design-environment can then actively offer the caregiver the option of collapsing the sequence of prompts, in the fashion of a critic, at the next script editing session. Triggering the collapsing of the prompts of the script is a script log indicating that the user had rapidly stepped through a sub-sequence of prompts in a script, indicating that the sub-sequence had been memorized and correctly identifying that it should be “triggered” at this point in the larger task. Collapsing of a sub-sequence is accomplished by producing a prompt that points to the sub-sequence and replacing that sub-sequence with the new prompt (e.g., replacing “open the refrigerator, take out the bread, close the refrigerator door, remove two slices from the bag” with “get two slices of bread out of the refrigerator”). This would trigger the new internal scripts (see section 7.2) with an overarching external script prompt. Similarly, expanding scaffolding could be triggered by the analyzed script-logs, indicating confusion about a specific prompt in a script. This confusion could be indicated by the client “rocking” over a section of the script (e.g., step-5, step-4, step-5, step-4, . . . , which might indicate an attempt by the client to clarify an instruction), by the client taking overly long to do a step, from a request for help on a MAPS-prompter equipped with a cell phone or wireless connectivity to a Lifeline server, or from other patterns of use that might be opportunistically gathered by using the state trail<sup>21</sup> of the script usage captured in the log. That prompt could be expanded by replacing the single prompt with a sub-sequence of prompts that break the task

---

<sup>21</sup> The state trail of a MAPS script is the recorded timing of viewing the prompts that constitute the script. This includes backtracking and requests for help. It is called a state trail to facilitate the use of software engineering / mathematical tools used to analyze and display “state machine” information.

into smaller chunks with many more prompts. Collapsing a sequence could also be done as a result of an explicit request, implementing a strategy of providing an *adaptation* function. In contrast, expanding a script could, if it were carefully designed to be an automatic action on the part of the MAPS system, implement an *adaptive* strategy.

### 8.3. MAPS as a Platform for Active Distributed Support

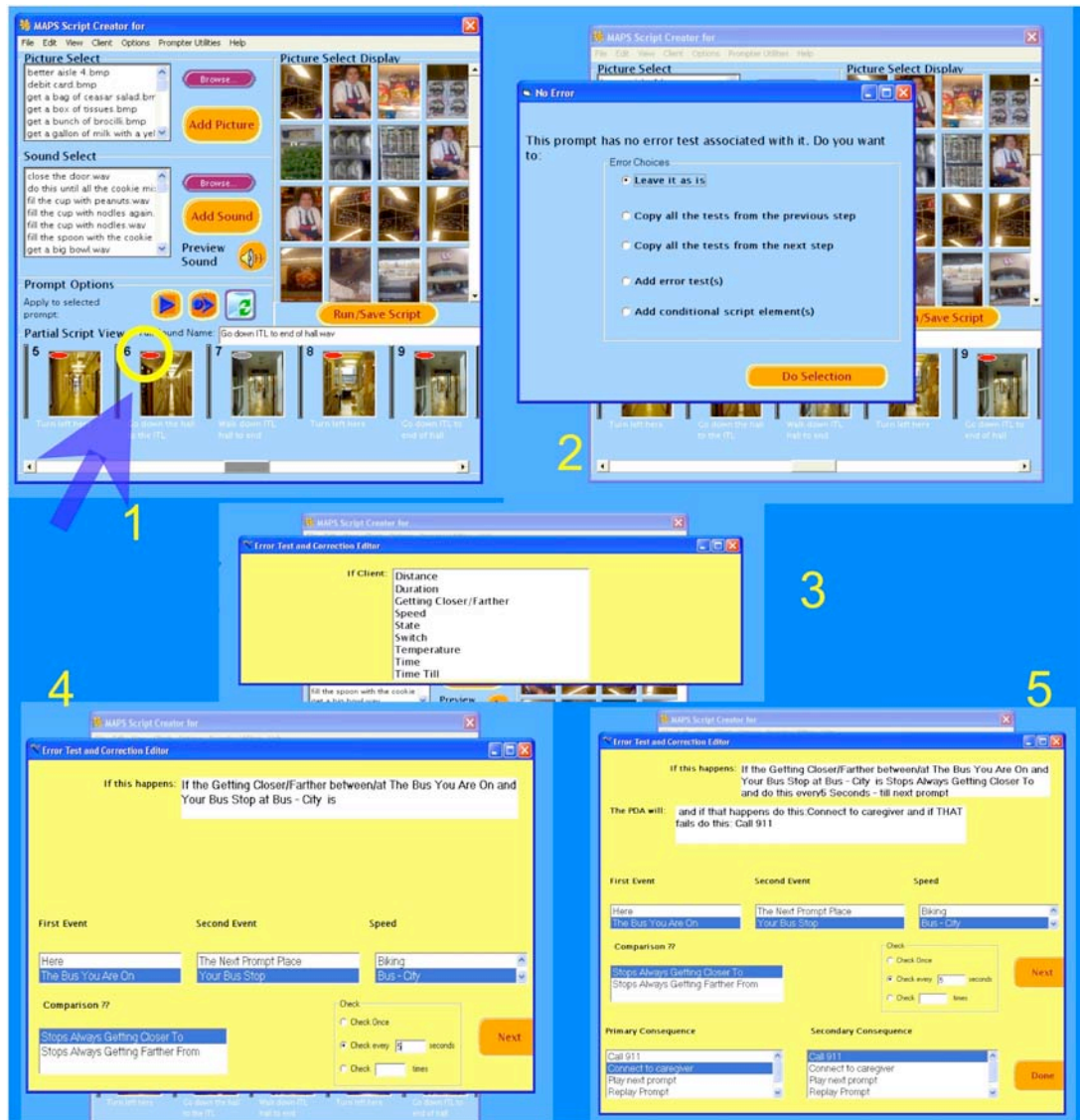
MAPS static prompting implicitly ignores the truth of plans that Lucy Suchman explicated so well in *Plans and Situated Actions* [Suchman, 1987]. The shocking and obvious truth is that formal, detailed plans almost never correspond exactly to the path of task completion; the environment changes, people change, and the task itself may change. To create a system that can robustly support task completion by a person with cognitive disabilities, the system must (1) be context aware and (2) be prepared to do *something* about changed conditions. MAPS was originally designed keeping in mind a number of possible extensions to the basic MAPS system that, although not part of this research, will be necessary to realistically support tasks in an ever-changing world. This more dynamic task support has been called *active distributed support* [Carmien et al., 2005b]. To implement this vision, changes needed to be made in the “production” MAPS script-design-environment and prompter. To anticipate this, initial prototypes of the functions would be needed, which were added to the ‘beta’ version of the script design environment. These prototypical functions fall into two categories: script structure and script-design-environment functions.

#### 8.3.1. Script Structure

In the area of script structure, scripts used in transportation and employment will need to have the ability to *actively* fork and loop, in contrast to the primitive forking and looping support provided in the current MAPS system (see section 5.3.3). In transportation scripts, the user needs to be dynamically told to wait for the bus, and to be told *at the right time* when to board the bus; similarly, in employment scripts, the client needs to be prompted through performing loops of a task and to change tasks when the environment makes it necessary to do so. This improvement requires making the MAPS-prompter context aware [Dey et al., 2001] [Dourish, 2004], the addition of a Lifeline server to the architecture, and encoding the appropriate annotations into the MAPS scripts.

### 8.3.2. Script-Design-Environment Functions

Starting from the last requirement and working up to the higher levels of the architecture, the following text describes the prototype system and proposed extensions. The MAPS script-design-environment has been extended to allow caregivers to annotate script prompts with error-tracking/conditional-prompting information. This was done for the working prototype shown in Figure 48 and Figure 49 and also in more detail, but not functional with Lifeline, in the current set of advanced functions in the MAPS script-design-environment (see Figure 50). The illustrations show, in the upper left screen shot (1), the ovals allowing insertion and querying of script annotations that are not apparent unless the advanced option is selected (see section 5.4.2.1 and Figure 29). The next screen shot (the upper right, (2)) shows the oval being clicked, the dialog box asking, in a wizard-like fashion, what the user wants to do. The next dialog box (the middle screen shot, (3)) is the result of having selected “add insert error test,” the choices for the kind of error test. Below that are the selected parameters for the selected error test; and finally at the bottom two screen shots ((4) and (5)) are test parameters and the consequences that the caregiver chooses for this script annotation, should it be triggered. The database structure to both support the conditional test and allow the easy addition to the lists of types and conditions of tests [Norman, 1981] by the tool designer is listed in appendix 13.8. By adding rows to the various tables in the database, the population of the various choices are changed, thus allowing the customization of the error GUI without re-coding and compilation. This is the current database that the caregivers used in the realistic tests (but with the additional annotating functions turned off). The Lifeline server accepts script and conditional test information, and, on the basis of the script annotation parameters and sensor information, intervenes on the MAPS-prompter and/or to the caregiver to do the appropriate action based on user, task, and environment.

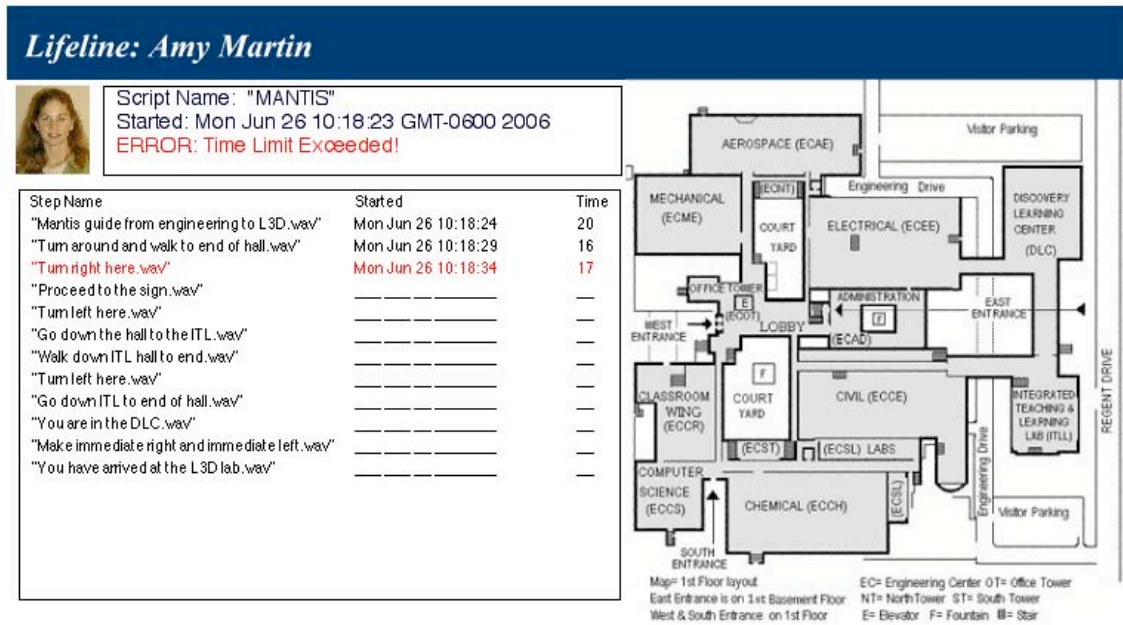


**Figure 50 - MAPS Error Trapping and Conditional Prompting Annotation Interface**

MAPS static task support could be made into dynamic and context-sensitive active task support [Carmien et al., 2005b] with the addition of dynamic prompt delivery, or dynamic script support. Another structural addition to scripting could be the dynamic pushing of a prompt dependent on a non-error state, such as suggesting an umbrella on a rainy day, or pushing a reminder of the overarching task as motivation [Banich, 2002] (“You are doing great, Anne, on your trip to Auntie Mary’s house!”), increasing the probability of successfully accomplishing the task. An active fork in a script could be needed in a script for employment where the task may depend on the contents of a queue;

similarly, active looping may be required in waiting for a bus at a bus stop with the script requiring a prompt to “ignore this bus, please keep waiting,” and repeating this until the correct bus arrives. Based on this architecture, the Lifeline server could, by using dedicated sensors embedded in the environment as well as opportunistically gathering context-meaningful information, such as real-time locations of city buses, push contextually meaningful prompts to either bring the user back to the script’s path or to use the changing environment to help the user adapt to changing circumstances in the task (e.g., suggesting that she take a coat in cold weather, or guiding the user to resource locations when the expected location is depleted). The domain of context and context awareness is rich with possibilities as well as depth of details, from ontologies [Dey et al., 2001] [Heckmann, 2006] to applications [Mihailidis & Fernie, 2002].

The other side of active distributed support is the provision for error trapping and error correction [Norman, 1983; Reason, 1990]. By using the advanced functions in the script-design-environment, the caregiver could craft (see Figure 50) error tests that would be passed on the Lifeline system, and, depending on the parameters of the tests and the state of the world as detected by the Lifeline system and its polling of sensors, an appropriate error-correcting response could be evoked. This response may range from repeating the last prompt on the prompter (“this is your bus, please get on it”) [Sullivan, 2005], to pushing another prompt to the prompter (“stay on the bus and someone will come to get you”), to contacting a caregiver with a contextualized message (“Cassy has ridden the bus past her bus stop, it is 11:23 p.m., and she was last at 23rd and Pine”), to connecting the cell phone that is part of the prompter (as the newer IPAQs are) to a caregiver, or any combination of all the above [Sullivan & Fischer, 2003]. When a caregiver has to be summoned or involved, the caregiver’s script configuration for the annotations would contain escalating information to ensure that something helpful would happen (e.g., try the mother; if no response, then the case worker; if no response, then 911).



**Figure 51 - Lifeline Caregiver Interface with Error Message**

A prototype of primitive error-trapping and coordination between events in the MAPS-prompter, the Lifeline server, and caregivers was implemented in 2004, providing an existence proof of both the functions of the proposed system and the usefulness of even a primitive system to prospective caregivers. The prototype consisted of a Lifeline applet running on the prompter that was also running the MAPS prompting application, a Lifeline server running on a server reachable from the Internet, and a web server running on the same machine as the Lifeline server that allowed caregivers to access the status of a ongoing script (see Figure 48). Upon startup, the Lifeline applet read the MAPS applications performance log and communicated with the Lifeline server. When the prompter loaded a script, the Lifeline server populated the website (a Macromedia Flash application) with the steps of the script, and as each step was invoked, passed to the server along with the time stamp and specifics of that step is the expected time that that step should take. If the next step is not invoked within that time limit, the Lifeline server puts an error message on the web interface and sends a message to the caregiver's cell phone (which it had read out of an initialization file when the server started up) (see Figure 51, left). The prompter user also had one of the hardware buttons on the IPAQ mapped to a panic message being sent to the Lifeline server (see Figure 51, left), which caused a different SMS



message and a corresponding message on the caregiver's web interface. The system has been demonstrated in the L<sup>3</sup>D laboratory and at places as disparate as Vienna and Atlanta.

#### **8.4. Work Supporting Prompting: An Extension of the Image Experiment**

The image experiment, as discussed in section 6.5, produced an experimental result supporting a preference of photographs rather than icons for prompt images. It is possible, however, that a stronger statement could be made that would be supported by experimental data. "Best practices" documentation [Assistive Technology, 2005] suggests that, due to the "universe of one" effect, particularly regarding the conjunction of cognitive disability and sensory impairments, a given person with cognitive disabilities may have sufficiently strong "image preferences"<sup>22</sup> to make a difference between a successful script and one that is problematic. Several best practices rules state that for each person with cognitive disabilities there are definite preferences to maximize matching accuracy. Much of the practice of matching an AT device and setup to users with cognitive disabilities is dependent on the personal skill of the professional, in contrast to matching persons with sensory or motor disabilities, which is much more reliant on objective tests for a successful match. Further work could be done in this area, more deeply exploring the variation of image types or styles and accuracy and speed of matching. The experiment should be redone with a larger sample set and many more participants, with a goal of: (1) confirming that this best practices insight is in fact true, and (2) developing a simple test, not unlike the MED-AUS taxonomy that supports general AT selection [Rehabilitation Research Design & Disability (R<sub>2</sub>D<sub>2</sub>) Center, 2006], which may allow a caregiver to administer to a person with cognitive disabilities to maximize the possibility of using the right object representation in a script.

A similar experiment might be designed and an attempt made to operationalize the choice of verbiage for efficacious prompts for specific individuals with cognitive disabilities. This language test could be used by a critiquing system in a later version of MAPS.

---

<sup>22</sup> In this case, the "image preference" does not imply what image is consciously pleasing to the user, but is closer to the sense of an eyeglass optical prescription.

## 9. Conclusions

This study began with the assertion that (1) some persons with cognitive disabilities, along with caregivers, could develop more independence in their world by using MAPS; (2) those AT systems that suffer from abandonment caused by difficult (re-)configuration can benefit from a dual-user approach to AT design; and (3) it is critical to the process of understanding high-functioning AT adoption to have an ecologically valid approach to studying the initial use of these AT systems, and ethnographic approaches are well suited for this. This study was limited due to the need to first develop the tools that would be used in the study. The few dyads that were studied did, by their experience with MAPS, demonstrate that by using MAPS—and by inference systems similar to MAPS—persons with cognitive disabilities did develop more independence. Independence in these cases is instanced by the first client doing laundry by herself, another client doing solo shopping, another folding clothing out of dryer by himself, and the last client being offered a regular employment contract in a much shorter time than the client’s job coach anticipated.

What are systems *similar to* MAPS? The key parts of MAPS that are critical to adoption are: (1) support for successful initial experience (i.e., dual-user interfaces); (2) user interfaces for persons with cognitive disabilities that provide the minimum degree of functionality but still provide supports for the task at hand; and (3) a design process that involves all the stakeholders. The need for dual-user interfaces follows from the case of a device failing due to difficulty in reconfiguration, and the role of the configuration falling to the caregiver (see section 2.1.4), therefore, attention should be paid to the caregiver’s application for configuring the AT device [Carmien & Kintsch, 2006]. Therefore, there is a need for a *dual-user-interface* approach.

In the end, the most relevant discoveries and inventions arising from this study are a usable innovative script-design-environment and an approach to studying and evaluating AT systems. The importance of the script-design-environment is illustrated by noting the needed complexity of a system that has in its GUI 40 controls versus the prompter’s 5. This 8:1 ratio illustrates the relative importance of “getting it right” for the script-design-environment, and why the ratio of time for developing the two systems was roughly about 70:1.

The innovative script-design-environment is based on the concept of the dual-user interface and developed in collaboration among caregivers, AT professionals, and tool designers—the primary stakeholders in this domain. Supporting this goal were the design guidelines of building on existing models of computer applications (see section 5.2.1); providing help with short videos of the system in use; viewing the script in various perspectives; and storing and retrieving scripts, which allowed for script reuse.

Why ecologically valid field studies? Because the very thing that makes controlled experiments in the laboratory scientifically valid (i.e., the removal of all confounding environmental influences) makes generalization to the real world problematic. MAPS is a component in a socio-technical environment, so the evaluation must be in a socio-technical environment [Mumford, 2000a]. The approach taken is to do tests in realistic environments, studied by using ethnographic methods. Only after that is it reasonable to perform more quantitative tests on the parts of the system.

The details of how this ethnographic approach was carried out are discussed in section 6.1.2, but a larger question needs to be considered here. What did participant observation offer that could not be obtained by more rigorous kinds of evaluation? The ethnographic approach captured the context and opportunity to observe changes in use over time. Also, the process of initial observations results in the client and caregiver reaching a comfort level with the observer so that when the first script is designed and used, both elements of the dyad are comfortable with the observer present. One of the problems created by the observer becoming so familiar with the persons and environment of the testing is that when breakdowns occur during script usage, it is difficult to refrain from stepping out of the role of observer and offering help too soon, thus undercutting the unique example of a breakdown. The other problem encountered with the ethnographic approach is that by the time the fourth dyad was studied, the freshness and detailed observations that characterized the first two dyads was not present. Both problems were purposely and proactively reacted against as the study progressed.

### **9.1. Leveling the Playing Field**

This research set out to discover and create a system to support independence for persons with cognitive disabilities and their caregivers. In the process of analyzing and proposing solutions to this

problem, innovative perspectives in using HCI lenses to study them were developed. From this deeper understanding, a system was created and refined. MAPS was then taken into representative settings, and the process of adoption by the dyads was studied. From these ethnographic observations, patterns and breakdowns were analyzed. Then a discussion of possible future work that was based on existent work was presented. Finally, this work was finished with some concrete suggestions for AT design and how the modification of the “generic” HCI concepts might provide feedback into the canon.

On a personal level, working with the young adults with cognitive disabilities left this researcher with a deeper understanding of the varieties of ability we all have and what it means to be human. By applying intelligence augmentation techniques and collaboratively working toward a design that empowered all the stakeholders, the MAPS system fostered independence for clients, in the best fashion—an independence that includes socialization and interdependence in the lives of both the persons with cognitive disabilities and their caregivers. This work demonstrated, to the researcher, how a properly conducted design process can result in not just artifacts, but real and positive changes in the lives of participants—persons with cognitive disabilities, caregivers, and technologists alike.

## 10. Glossary

Phrase	Definition	Reference External / Internal <sup>23</sup>	Section
Abandonment	A substantial portion of all assistive technology is abandoned after initial purchase and use—as high as 70% in some cases. A large component of the cause for such abandonment is difficulty in configuring and adapting (reconfiguring) software.	External [Phillips & Zhao, 1993]  Internal [Kintsch & dePaula, 2002]	2.2
Activity:	Two definitions. Activity theory: “a form of doing directed to an object”[Kuutti, 1996]. Common usage: An activity is simply intentional behavior bound by time and space	External [Engeström, 2001]	3.2
Activities of Daily Living (ADL):	Activities of daily living: Routine activities carried out for personal hygiene and health and for operating a household. (e.g., bathing with sponge, bath, or shower; dressing; toilet use; Transferring (in and out of bed or chair); urine and bowel continence; eating)	External [Scherer, 1996]	1.1
Affordance:	The attributes of an object that express the ability to perceive the possible actions performed in a specific type of environment or on a particular object. It is the perceived and actual properties of an object that determine how the object is used. In other words, affordances provide hints or clues about the object’s function.	External [Norman, 1993]	7.2
Artifact	Any object made or used by humans		
Assistive technology (AT):	Any item, piece of equipment, or product system, whether acquired commercially off the shelf, modified, or customized, that is used to increase, maintain, or improve functional capabilities of individuals with disabilities.	External [1988]  Internal [Kintsch & dePaula, 2002]	2.1.2

---

<sup>23</sup> The external reference is the canonical paper, the internal is a paper on the topic from a CLever and/or L<sup>3</sup>D perspective

Phrase	Definition	Reference External / Internal	Section
Augmentative and Alternative Communication devices	Alternative and Augmentative Communication (AAC) involves alternate methods of communicating needs, feelings, ideas, and perceptions through the use of electronic and nonelectronic devices that provide a means for expressive and receptive communication for persons with limited or no speech.	External [Beukelman & Mirenda, 1998]	2.1.2
Boundary object	Artifacts, documents and perhaps even vocabulary that can help people from different communities build a shared understanding. Boundary objects will be interpreted differently by the different communities, and it is an acknowledgment and discussion of these differences that enables a shared understanding to be formed.	External [Nardi, 1996b]	3.4
Caregiver	A person providing some form of assistance to another person who is dependent on that help due either to medical reasons or to a simple inability to conduct their routine ADLs. (e.g., a parent, teacher, or professional)	External [Snell, 1987]  Internal [Kintsch & dePaula, 2002]	2.1.4
“Collapse point” steps	A step or prompt in a script that subsumes a set of following steps. A collapse point can stand for the following steps when they become internal scripts		6.6.2
Cognitive disabilities; Cognitive impairments	Any disability that affects mental processes, including, but certainly not limited to, mental retardation, attention deficit disorder, dyslexias, aphasias, language delay, dementias, and other psychiatric and behavioral disorders	External [American Psychiatric Association, 2000]	2.1.3
Communities of practice	Groups of practitioners who work as a community in a certain domain.	External - [Lave, 1991]	3.4
Context-aware computing	Acquiring and utilizing information about the context of a device to provide services that are appropriate to the particular people, place, time, events.	External -[Dey et al., 2001]	3.3.1

Phrase	Definition	Reference External / Internal	Section
Deficiencies in executive function	A category of cognitive functions that includes abilities such as: anticipating future needs and planning accordingly, setting priorities, regulating impulses and drives, self-awareness and self-correction.	External [Norman & Shallice, 2000]	2.3
Deficiencies in memory	Inability to recall or store information in one's memory, either to completely remember or recall only part of the stored information.	External [Reason, 1990]	1.2
Distributed cognition	Knowledge about the world is contained in both internal (mental) structures and elements in the environment, containing externally stored information, and derived by combining the two. Distributed cognition is an approach that views the cognitive act as a result of a system that is comprised of an actor and artifacts in the actor's environment.	External [Hollan et al., 2001]  Internal [Fischer, 2003]	3.3
External scripts	An external artifact or aid that guides a person through doing a task that he or she could not otherwise do by using already available chunks of behavior <see internal scripts>	Internal [Carmien et al., 2006]	7.2
High-functioning AT	Assistive technology that is designed to support missing cognitive abilities, typically complex to configure and computer based; examples include augmentative and alternative communication devices and task support systems		1.4
Independence	In the context of persons with cognitive disabilities, independence refers to being able to live and function with as minimal support from caregivers and others as possible. Independence needs to be appropriately balanced with inclusion and socialization.	Internal [Carmien et al., 2005a]	1.1
Internal scripts	Scripts that guide individuals in the way they process tasks they are faced with, and these internal scripts are standing in a complex relationship to the external scripts provided by an artifact or by other persons.	Internal [Carmien et al., 2006]	7.2
Metadesign	Tools that designers use to do design. This additional level of design is metadesign. System-building techniques that support metadesign include "end-user modification" and "end-user programming."	Internal [Giaccardi & Fischer, 2005]	3.4

Phrase	Definition	Reference External / Internal	Section
Orthotic	An orthotic device or orthosis is a device that is applied externally to the limb or body. The purpose can be to provide support or extension of a function. In this case, a cognitive orthotic supplements (not replaces) cognitive function.	External [LoPresti et al., 2004]	1.6
Panic button	A device, or function in a device that allows a user to summon help, usually involving direct personal intervention.		6.2
Plan	An abstract representation, whether explicit or implicit, of a set of steps that when performed will lead to a desired goal, A plan has at minimum two elements: the steps that are intended to be executed and the goal that they will accomplish.	External [Schank & Abelson, 1977]  Internal [Carmien et al., 2006]	2.1.1
Prompt	A set of external, physical artifacts that assists in guiding a person through a task. An example is a card with a picture of the step in a task to do, or a set comprising an image and a verbal instruction. In this context a ringing of a timer does not constitute, of itself, a prompt: here prompts have at least some minimal content.	External [Epstein et al., 2001]  Internal [Carmien, 2005]	2.1.1
Prompter	A computationally based system that presents images and sounds that comprise a prompt, a series of prompts constitute a script, which is an instruction that guides a person to complete a task.	External [Davies et al., 2002]	5.4.1
Satisficing	Satisficing is an alternative to optimization for cases where there are multiple and competitive objectives in which one gives up the idea of obtaining a “best” solution. In this approach, one sets lower bounds for the various objectives that, if attained, will be “good enough” and then seeks a solution that will exceed these bounds.	External [Newell & Simon, 1972]	0
Script	A representation of a task broken down into discrete steps	External [Schank & Abelson, 1977]  Internal [Carmien, 2004a]	1.2
Socio-technical environments	Systems that have both technical and human/social aspects that are tightly bound and interconnected. Socio-technical design is an approach to complex organizational work design that recognizes the interaction between people and technology in workplaces.	External [Mumford, 1987]  Internal [Carmien et al., 2005a]	1.1



Phrase	Definition	Reference External / Internal	Section
Symmetry of ignorance (or asymmetry of knowledge)	A way of describing situations in which several participants or roles in an endeavor each individually have parts of the knowledge needed to accomplish the task, but none has enough to accomplish the task independently	External [Rittel, 1984]  Internal [Fischer et al., 2002]	3.4
Task	A group of behaviors that accomplish a goal, typically of short duration (i.e., making a pizza is a task, raising a child is not).		
TfLea (tool for learning)	A tool for learning is “something that I use at some point that leaves me changed so I can do, by myself, what I could not do before.” An example of this is reading.	Internal [Carmien & Fischer, 2005]	7.2.2.1
TfLiv (tool for living)	A tool for living is “something that I need to use to do what everyone else can do ‘normally.’ ” An example of this is eyeglasses.	Internal [Carmien & Fischer, 2005]	7.2.2.2
Underbuilt	To underbuild a tool is to make it with enough functionality to accomplish a task but not so much functionality built-in to force the task to be accomplished in a specific way.	External [Brand, 1995]	3.5
WAIS-R	An IQ test widely used with persons with cognitive disabilities. The WAIS-R gives a global IQ and also two separate IQs for the two scales: verbal and performance. There are 6 verbal subscales and 5 performance subscales.	External [Wechsler, 1997]	6.6.1
Wicked Problems	Wicked problems have incomplete, contradictory, and changing requirements. The problem is not understood until after formulation of a solution. Stakeholders have radically different worldviews and different frames for understanding the problem. Constraints and resources to solve the problem change over time. The problem is never solved but solutions to wicked problems are typically better, worse, or good enough (satisficing).	External [Rittel, 1984]	3.4

## 11. Abbreviations

Abbreviations	Term	First Mention
AAMR	American Association on Mental Retardation	2.1.3
ACM	Association for Computing Machinery	2.1.2
ADA	Americans with Disabilities Act of 1990	2.1.2
ADL	Activities of daily living	1.1
AI	Artificial intelligence	1.1
AT	Assistive technology	1.1
ATM	Automatic teller machine	1.1
BVSD	Boulder Valley School District	1.1
CLever	Cognitive Levers	1.1
dpi	Dots per inch	5.2.2
GUI	Graphical user interface	5.2.1
HCI	Human-computer interaction (or interface)	1.1
IADL	Instrumental activities of daily living	1.1
L <sup>3</sup> D	The Center for Lifelong Learning and Design	1.1
MAPS	Memory Aiding Prompting System	1.1
NCTM	National Council of Teachers of Mathematics	7.2.2.4
ODBC	Open database connectivity	5.4.2
PEAT	Planning and Execution Assistant and Training	4.3
PC	Personal Computer	1.1
PDA	Personal digital assistant	1.1

Abbreviations	Term	First Mention
RERC-ACT	Rehabilitation Engineering Research Center for the Advancement of Cognitive Technologies	1.1
RESNA	Rehabilitation Engineering and Assistive Technology Society of North America	2.1.2
SIGCAPH	(ACM) Special Interest Group on Computers and the Physically Handicapped	4
TfLiv	Tool for living	7.2.2.2
TfLea	Tool for learning	7.2.2.1
UI	User interface	5.4.2.1
VB	Visual Basic	5.4.1
WIN-CE	Windows – Compact Edition	5.4.1
XML	eXtensible Markup Language	4.3

## 12. References

- AbleLink (2003) AbleLink Technologies website, available at [www.ablelinktech.com](http://www.ablelinktech.com).
- ADA (1990) American with Disabilities Act of 1990, available at [www.usdoj.gov/crt/ada/adahom1.htm](http://www.usdoj.gov/crt/ada/adahom1.htm).
- Aist, E. H. (1973) *The Effect of Two Patterns of Visual Prompting on Learner Achievement in Industrial Arts*, Department of Psychology, Arizona State University.
- Allen, R. B. (1997) "Mental Models and User Models." In M. G. Helander, T. K. Landauer, & P. V. Prabhu (Eds.), *Handbook of Human-Computer Interaction, Volume 1*, Elsevier Science B.V., Amsterdam, pp. 49-63.
- The American Association on Mental Retardation (2002) AAMR Definition of Mental Retardation, available at [http://www.aamr.org/Policies/faq\\_mental\\_retardation.shtml](http://www.aamr.org/Policies/faq_mental_retardation.shtml).
- American Psychiatric Association (2000) *Diagnostic and statistical manual of mental disorders: DSM-IV-TR*, 4th, text revision. ed., American Psychiatric Association, Washington, DC.
- Assistive Technology (2005) Evaluware. <http://www.assistivetech.com/p-evaluware.htm>
- Baesman, B., & Baesman, N. (2000) *The Visions System*, available at <http://www.thevisionssystem.com>.
- Banich, M. T. (2002) "The Role of Attention in Cognitive Disabilities (poster)," *Research Frontiers in Cognitive Disability and Technology (II)*, 2002 Coleman Institute Conference on Cognitive Disability and Technology, Boulder, Colorado.
- Barry, D. (2006) Executive Function – Definition, Description, Available at <http://www.minddisorders.com/Del-Fi/Executive-function.html>
- Barthes, R. (1972) *Mythologies*, Hill and Wang, New York.
- Bateson, G. (1972) *Steps to an Ecology of Mind*, Chandler Publishing Company. San Francisco, CA
- Berg, B. L. (2004) *Qualitative Research Methods, Fifth Edition.*, Allyn and Bacon. Boston, MA,
- Bernard, H. R. (2000) *Social Research Methods*, Sage Publications, inc., Thousand Oaks, CA.
- Bernard, H. R. (2002) *Research Methods in Anthropology*, Altamira Press, Walnut Creek, CA.
- Beukelman, D., & Mirenda, P. (1998) *Augmentative and Alternative Communication*, 2nd ed., Paule H. Brookes Publishing, Baltimore, MD.
- Bodine, C. (2005) Assistive Technology Partners, available at <http://www.uchsc.edu/atp>.

- Boulder Valley School District (2006) BVSD Special Education Transition Employment, available at [http://content.bvsd.org/specialed/bv\\_spedtransition.shtml](http://content.bvsd.org/specialed/bv_spedtransition.shtml).
- Boyd-Graber, J., Boyd-Graber, J., Nikolova, S., Moffatt, K., Kin, K., Lee, J., Mackey, L., Tremaine, M., & Klawe, M. (2006) "Participatory Design with Proxies: Developing a Desktop-PDA System to Support People with Aphasia," *CHI'06 Conference on Human Factors in Computing Systems*, Montreal, Quebec, pp. 151-160.
- Braddock, D. (2006) *2006 Cognitive Disability in US*, University of Colorado, Coleman Institute.
- Brand, S. (1995) *How Buildings Learn: What Happens After They're Built*, Penguin Books, New York.
- BVSD (2004) Boulder Valley School District website, available at [www.bvsd.k12.co.us](http://www.bvsd.k12.co.us).
- Carmien, S., & Hudak, G. (2003) "MAPS: PDA Scaffolding for Independence for Persons with Cognitive Impairments," 26th International Conference on Technology and Rehabilitation, Atlanta GA.
- Carmien, S., & Fischer, G. (2003-2006) "(HRC-0402.01) MAPS: PDA Scaffolding for Independence for Persons with Cognitive Impairment, a part of the overall project "Building Cognitive Levers to help people help themselves," Human Research committee, University of Colorado.
- Carmien, S. (2004a) "Doctoral Consortium: MAPS: Creating Socio-technical Environments in Support of Distributed Cognition for People with Cognitive Impairments and Their Caregivers," *CHI '04 Extended Abstracts on Human Factors in Computing Systems*, Vienna, Austria, pp. 1051-1052.
- Carmien, S. (2004b) "Task Support for People with Cognitive Impairments and Their Caregivers," *American Journal of Occupational Therapy*, 14(3), pp. 1-4.
- Carmien, S. (2005) "End User Programming and Context Responsiveness in Handheld Prompting Systems for Persons with Cognitive Disabilities and Caregivers," *Proceedings of CHI'05 Conference on Human Factors in Computing Systems*, Portland, OR, pp. 1252-1255.
- Carmien, S., Dawe, M., Fischer, G., Gorman, A., Kintsch, A., & Sullivan, J. F. (2005a) "Socio-Technical Environments Supporting People with Cognitive Disabilities Using Public Transportation," *Transactions on Human-Computer Interaction (ToCHI)*, 12(2), pp. 233-262.
- Carmien, S., DePaula, R., Gorman, A., & Kintsch, A. (2005b) "Increasing Workplace Independence for People with Cognitive Disabilities by Leveraging Distributed Cognition among Caregivers and Clients," *Computer Supported Cooperative Work (CSCW) - The Journal of Collaborative Computing*, 13, pp. 443-470.
- Carmien, S., & Fischer, G. (2005) "Tools for Living and Tools for Learning," (CD-ROM), *11th International Conference on Human-Computer Interaction*, Las Vegas, NV.
- Carmien, S., Fischer, F., Fischer, G., & Kollar, I. (2006) "The Interplay of Internal and External Scripts — A Distributed Cognition Perspective." In F. Fischer, H. Mandl, J. Haake, & I. Kollar (Eds.), *Scripting Computer-Supported Communication of Knowledge – Cognitive, Computational, and Educational Perspectives*, (in press).

- Carmien, S., & Kintsch, A. (2006) "Dual User Interface Design as Key to Adoption for Computationally Complex Assistive Technology," *29th International Conference on Technology and Rehabilitation (RESNA)*, Atlanta GA.
- Carmien, S., & Wohldman, E. (2006) "Mapping Images to Objects by Cognitively Disabled Young Adults," (*submitted*) *American Journal on Mental Retardation*.
- Carroll, J. M. (2003) *HCI Models, Theories and Frameworks, Toward a Multidisciplinary Science*, Morgan Kaufmann Publishers. San Fransisco, CA
- CLever (2005) CLever: Cognitive Levers -- Helping People Help Themselves, available at <http://13d.cs.colorado.edu/clever/>.
- Cole, E. (1997) "Cognitive Prosthetics: An Overview to a Method of Treatment," *NeuroRehabilitation*, 12(1), pp. 31-51.
- Cole, E., & Dehdashti, P. (1998a) "Patient-Centered Design as a Research Strategy for Cognitive Prosthetics: Lessons Learned from Working with Patients and Clinicians for 2 Decades," *The Third Annual ACM Conference on Assistive Technologies*, Marina Del Rey, CA, pp. 11-18.
- Cole, E., & Dehdashti, P. (1998b) "Computer Based Cognitive Prosthetics: Assistive Technology for the Treatment of Cognitive Disabilities," *The Third Annual ACM Conference on Assistive Technologies*, Marina Del Rey, CA, pp. 11-18.
- Cole, E. (2006) "Patient-Centered Design as a Research Strategy for Cognitive Prosthetics: Lessons Learned from Working with Patients and Clinicians for 2 Decades," *CHI'06 Workshop on Designing Technology for People with Cognitive Impairments*, Montreal, Quebec.
- Coleman (2004) Coleman Institute for Cognitive Disabilities website, available at [www.cu.edu/ColemanInstitute](http://www.cu.edu/ColemanInstitute).
- Csikszentmihalyi, M. (1996) *Creativity — Flow and the Psychology of Discovery and Invention*, HarperCollins Publishers, New York.
- Davies, D. K., & Stock, S. E. (1996) "PictureCoach and PocketCoach: An Integrated Multi-Media Training System for Teaching and Maintaining Vocational Skills for Adults with Developmental Disabilities," *Closing the Gap*, Minneapolis, MN.
- Davies, D., Stock, S., & Wehmeyer, M. L. (2002) "Enhancing Independent Task Performance for Individuals with Mental Retardation through Use of a Handheld Self-Directed Visual and Audio Prompting System," *Education and Training in Mental Retardation and Developmental Disabilities*, 37(2), pp. 209-218.
- Davies, D. K. (2004) AbleLink Technologies website, available at [www.ablelinktech.com](http://www.ablelinktech.com).
- Davies, D. K., Stock, S. E., Brown, R. B., & King, L. R. (2005) *AbleLink Instructional Media Standard (AIMS) - Conceptual Overview*, AbleLink Technologies, Inc. Colorado Springs CO

- Davies, E. A., & Miyake, N. (2004) "Special Issue: Scaffolding," *The Journal of the Learning Sciences*, 13(3), pp. 265-451.
- dePaula, R. (2004) *The Construction of Usefulness: How Users and Context Create Meaning with a Social Networking System*, Ph.D. Dissertation, University of Colorado at Boulder.
- Dey, A. K., Abowd, G. D., & Salber, D. (2001) "A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications," *Human-Computer Interaction*, 16(2-4), pp. 97-166.
- Dixon, L. (1981) "A Functional Analysis of Photo-Object Matching Skills of Severely Retarded Adolescents," *Journal of Applied Behavior Analysis*, 14(4), pp. 465-478.
- Dourish, P. (2004) "What We Talk About When We Talk About Context," *Personal Ubiquitous Computing*, 8(1), pp. 19-30.
- Ehn, P. (1989) *Work-Oriented Design of Computer Artifacts*, Second ed., Arbetslivscentrum, Stockholm.
- Emerson, R. M., Fretz, R. I., & Shaw, L. L. (1995) *Writing Ethnographic Fieldnotes*, University of Chicago Press.
- Engelbart, D. C. (1988) "A Conceptual Framework for the Augmentation of Man's Intellect." In I. Greif (Ed.), *Computer-Supported Cooperative Work: A Book of Readings*, Morgan Kaufmann Publishers, San Mateo, CA, pp. 35-66.
- Engeström, Y. (2001) "Expansive Learning at Work: Toward an Activity Theoretical Reconceptualization," *Journal of Education and Work*, 14(1), pp. 133-156.
- Epstein, J. N., Willis, M. G., Conners, C. K., & Johnson, D. E. (2001) "Use of a Technological Prompting Device to Aid a Student with Attention Deficit Hyperactivity Disorder to Initiate and Complete Daily Tasks: An Exploratory Study," *JSET E Journal*, Available at <http://jset.unlv.edu/>
- Farber, E. T. (1962) "Faber College Motto," National Lampoon\
- Fischer, G., Lemke, A. C., Mastaglio, T., & Morch, A. (1990) "Using Critics to Empower Users." In J. Carrasco, & J. Whiteside (Eds.), *Proceedings of CHI'90 Conference on Human Factors in Computing Systems*, ACM, New York, pp. 337-347.
- Fischer, G., & Nakakoji, K. (1992) "Beyond the Macho Approach of Artificial Intelligence: Empower Human Designers—Do Not Replace Them," *Knowledge-Based Systems Journal, Special Issue on AI in Design*, 5(1), pp. 15-30.
- Fischer, G. (1994) "Domain-Oriented Design Environments," *Automated Software Engineering*, 1(2), pp. 177-203.
- Fischer, G. (1998) "Making Learning a Part of Life—Beyond the 'Gift-Wrapping' Approach of Technology." In P. Alheit, & E. Kammler (Eds.), *Lifelong Learning and Its Impact on Social and Regional Development*, Donat Verlag, Bremen, Germany, pp. 435-462.

- Fischer, G., Nakakoji, K., Ostwald, J., Stahl, G., & Sumner, T. (1998) "Embedding Critics in Design Environments." In M. T. Maybury, & W. Wahlster (Eds.), *Readings in Intelligent User Interfaces*, Morgan Kaufmann, San Francisco, pp. 537-559.
- Fischer, G. (1999a) "Domain-Oriented Design Environments—Supporting Individual and Social Creativity." In J. S. Gero, & M. L. Maher (Eds.), *Computational Models of Creative Design IV*, Key Centre of Design Computing and Cognition, Sydney, Australia, pp. 83-111.
- Fischer, G. (1999b) "Social Creativity, Symmetry of Ignorance and Meta-Design," *Proceedings of Creativity & Cognition 1999*, New York, pp. 116-123.
- Fischer, G. (2000) "Social Creativity, Symmetry of Ignorance and Meta-Design," *Knowledge-Based Systems Journal (Special Issue on Creativity & Cognition)*, Elsevier Science B.V., Oxford, UK, 13(7-8), pp. 527-537.
- Fischer, G., & Scharff, E. (2000) "Meta-Design—Design for Designers," *3rd International Conference on Designing Interactive Systems (DIS 2000)*, New York, pp. 396-405.
- Fischer, G. (2001) "User Modeling in Human-Computer Interaction," *User Modeling and User-Adapted Interaction (UMUAI)*, 11(1), pp. 65-86.
- Fischer, G., Ehn, P., Engeström, Y., & Virkkunen, J. (2002) "Symmetry of Ignorance and Informed Participation," *Proceedings of the Participatory Design Conference (PDC'02)*, Malmö University, Sweden, pp. 426-428.
- Fischer, G., & Ostwald, J. (2002) "Seeding, Evolutionary Growth, and Reseeding: Enriching Participatory Design with Informed Participation," *Proceedings of the Participatory Design Conference (PDC'02)*, Malmö University, Sweden, pp. 135-143.
- Fischer, G. (2003) "Distributed Cognition: A Conceptual Framework for Design-for-All." In C. Stephanidis (Ed.), *Proceedings of HCI International 2003 (Crete, Greece), Volume Vol. 4*, Lawrence Erlbaum Associates, Mahwah, NJ, pp. 78-82.
- Fischer, G. (2006) "Distributed Intelligence: Extending the Power of the Unaided, Individual Human Mind," *Proceedings of Advanced Visual Interfaces (AVI) Conference*, Venice, May 23-26, pp. 7-14.
- Fischer, G., & Giaccardi, E. (2006) "Meta-Design: A Framework for the Future of End User Development." In H. Lieberman, F. Paternò, & V. Wulf (Eds.), *End User Development — Empowering People to Flexibly Employ Advanced Information and Communication Technology*, Kluwer Academic Publishers, Dordrecht, The Netherlands, pp. 427-458.
- Frankford Hospitals (2004) Frankford Hospitals - Home Health - Glossary, available at <http://www.frankfordhospitals.org/e3front.dll?durki=7140#C>.
- Galvin, J. C., & Donnell, C. M. (2002) "Educating the Consumer and Caretaker on Assistive Technology." In M. J. Scherer (Ed.), *Assistive Technology: Matching Device and Consumer for Successful Rehabilitation*, American Psychological Association, Washington, DC, pp. 153-167.



- Giaccardi, E. (2004) *Principles of Metadesign: Processes and Levels of Co-Creation in the New Design Space*, Ph.D. Dissertation, CAiiA-STAR, School of Computing, Plymouth, UK.
- Giaccardi, E., & Fischer, G. (2005) "Creativity and Evolution: A Metadesign Perspective," *Proceedings of the European Academy of Design (EAD-6) Conference*, Bremen, Germany, March, available at [http://ead06.hfk-bremen.de/conference/ead06\\_cdrom.zip](http://ead06.hfk-bremen.de/conference/ead06_cdrom.zip).
- Gilberts, G., Agran, M., Huges, C., & Wehmeyer, M. (2001) "The Effects of Peer Delivered Self-Monitoring Strategies on the Participation of Students with Severe Disabilities in General Education Classrooms," *Journal of the Association for Persons with Severe Handicaps*, 26(1), pp. 25-36.
- Gorman, A. (2005) Lifeline Website, available at <http://l3d.cs.colorado.edu/clever/projects/lifeline.html>.
- Grudin, J. (1987) "Social Evaluation of the User Interface: Who Does the Work and Who Gets the Benefit?" In H. Bullinger, & B. Shackel (Eds.), *Proceedings of INTERACT'87, 2nd IFIP Conference on Human-Computer Interaction (Stuttgart, FRG)*, North-Holland, Amsterdam, pp. 805-811.
- Guzdial, M. (1994) "Software-Realized Scaffolding to Facilitate Programming for Science Learning," *Interactive Learning Environments*, v4 n1 p1-44
- Heckmann, D. D. (2006) UbisWorld - Research on User Modeling and Ubiquitous Computing, available at <http://www.ubisworld.org/>.
- Hollan, J., Hutchins, E., & Kirsch, D. (2001) "Distributed Cognition: Toward a New Foundation for Human-Computer Interaction Research." In J. M. Carroll (Ed.), *Human-Computer Interaction in the New Millennium*, ACM Press, New York, pp. 75-94.
- Hutchins, E. (1994) *Cognition in the Wild*, The MIT Press, Cambridge, MA.
- Imagine! (2004) Imagine! website, available at [www.imaginecolorado.org](http://www.imaginecolorado.org).
- Isaac (1998) Isaac Project, available at <http://www.certec.lth.se/english/Isaac/>.
- Jönsson, B., & Svensk, A. (1995) "Isaac—A Personal Digital Assistant for the Differently Abled." In *Proceedings of the 2nd TIDE Congress*, Paris, France, pp. 356-361. Available at: [www.certec.lth.se/doc/isaaca](http://www.certec.lth.se/doc/isaaca).
- Katz, S., Ford, A., Moskowitz, R., Jackson, B., & Jaffe, M. (1963) "Studies Of Illness in The Aged. The Index Of ADL: A Standardized Measure of Biological and Psychosocial Function.," *Journal of the American Medical Association*, 21(185), pp. 914-919.
- Keating, T. (2006) Picture Planner, available at [http://www.eugeneresearch.org/picture\\_planner/index.html](http://www.eugeneresearch.org/picture_planner/index.html).
- King, T. (1999) "Assistive Technology—Essential Human Factors." Allyn & Bacon, Boston.

- King, T. (2001) "Ten Nifty Ways to Make Sure Your Clients Fail with AT and AAC! (A Human Factors Perspective on Clinical Success—or Not)," 19th Annual Conference: Computer Technology in Special Education and Rehabilitation.
- Kintsch, A. (2000) "Special Software for Special Students," CD distributed by the Boulder Valley School District.
- Kintsch, A., & dePaula, R. (2002) "A Framework for the Adoption of Assistive Technology," *SWAAAC 2002: Supporting Learning Through Assistive Technology*, Winter Park, CO, pp. E3 1-10.
- Kobsa, A. (2001) "Generic User Modeling Systems," *User Modeling and User-Adapted Interaction*, 11, pp. 49-63.
- Krantz, P., MacDuff, M., & McClannahan, L. (1993) "Programming Participation in Family Activities for Children with Autism: Parents' Use of Photographic Activity Schedules," *Journal of Applied Behavior Analysis*, 26(1), pp. 137-138.
- Kuutti, K. (1996) "Activity Theory as a Framework for Human-Computer Interaction Research." In B. Nardi (Ed.), *Context and Consciousness*, MIT Press, Cambridge, MA, pp. 17-44.
- L<sup>3</sup>D (2005) Center for LifeLong Learning and Design, University of Colorado, Boulder, available at <http://l3d.cs.colorado.edu/>.
- Lancioni, G., Van den Hof, E., Furniss, F., O'Reilly, M., & Cunha, B. (1999) "Evaluation of a Computer-aided System Providing Pictorial Task Instructions and Prompts to People with Severe Intellectual Disability," *Journal of Intellectual Disability Research*, 43(1), pp. 61-66.
- Lancioni, G., O'Reilly, M., Seedhouse, P., Furniss, F., & Cunha, B. (2000) "Promoting Independent Task Performance by Persons with Severe Developmental Disabilities through a New Computer-aided System," *Behavior Modification*, 24(5), pp. 700-718.
- LaPlante, M. E., Hendershot, G. E., & Moss, A. J. (1997) "The Prevalence of Need for Assistive Technology Devices and Home Accessibility Features," *Technology and Disability*, 6, pp. 17-28.
- Lave, J. (1991) "Situated Learning in Communities of Practice." In L. B. Resnick, J. M. Levine, & S. D. Teasley (Eds.), *Perspectives on Socially-Shared Cognition*, American Psychological Association, Washington, DC, pp. 63-82.
- Lawton MP, & Brody, EM. (1969) "Assessment of Older People: Self-Maintaining and Instrumental Activities of Daily Living.," *Gerontologist*, 9(3), pp. 179-186.
- LeCompte, M., & Schensul, J. (1999) *Analyzing & Interpreting Ethnographic Data*, Altamira Press, Walnut Creek, CA.
- Levinson, R. (2003) PEAT: The Planning and Execution Assistant and Training System., available at [www.brainaid.com](http://www.brainaid.com).

- Lewis, C., & Rieman, J. (1993) *Task-Centered User Interface Design: A Practical Introduction*, available at <ftp://ftp.cs.colorado.edu/pub/cs/distribs/clewis/HCI-Design-Book>.
- Lofland, J., & Lofland, I. H. (1995) *Analyzing Social Settings: A Guide to Qualitative Observation*, Wadsworth, Belmont, CA.
- LoPresti, E. (2003) Definition of Cognitive Disabilities, RESNA SIG-20, available at <http://www.resna.org/sigs/sig20/definition.htm>.
- LoPresti, E. F., Mihailidis, A., & Kirsch, N. (2004) "Assistive Technology for Cognitive Rehabilitation: State of the Art," *Neuropsychological Rehabilitation*, 14(1-2), pp. 5-39.
- LRE for LIFE Project (2001) Steps for Building Instructional Program Packets: Selected Activity Analyses (Unabbreviated Version), University of Tennessee, College of Education, available at <http://web.utk.edu/~lre4life/ftp/TADSman.PDF>.
- Luckasson, R., & Reeve, A. (2001) "Naming, Defining, and Classifying in Mental Retardation," *Mental Retardation*, 39(1), pp. 47-52.
- Lynch, W. (1995) "You Must Remember This: Assistive Devices for Memory Impairment," *Journal of Head Trauma Rehabilitation*, 10(1), pp. 94-97.
- Martin, B., and McCormack, L. (1999) "Issues Surrounding Assistive Technology Use and Abandonment in an Emerging Technological Culture.," *Proceedings of Association for the Advancement of Assistive Technology in Europe (AAATE) Conference*. Düsseldorf, Germany
- McCallion, P. J., Matthew (1997) "Exploring The Impact Of Culture And Acculturation On Older Families Caregiving For Persons with Developmental Disabilities," *Family Relations*, 46(4), p. 347.
- McCue, M., Chase, S., Dowdy, C., Pramuka, M., Petrick, J., Aitkne, S., & Fabry, P. (1994) "Functional Assessment of Individuals with Cognitive Disabilities: A Desk Reference for Rehabilitation," Center for Applied Neuropsychology, Pittsburgh, PA.
- McLuhan, M. (1964) *Understanding Media: The Extensions of Man*, The MIT Press, Cambridge, MA.
- Mihailidis, A., & Fernie, G. (2002) "Context-Aware Assistive Devices for Older Adults with Dementia," *Gerontechnology*, 2(2), pp. 173-188.
- Mihailidis, A., Barbenel, J. C., & Fernie, G. (2004) "The Efficacy of an Intelligent Cognitive Orthosis to Facilitate Handwashing by Persons with Moderate-to-Severe Dementia.," *Neuropsychological Rehabilitation*, 14(1-2), pp. 135-171.
- Mumford, E. (1987) "Sociotechnical Systems Design: Evolving Theory and Practice." In G. Bjerknes, P. Ehn, & M. Kyng (Eds.), *Computers and Democracy*, Avebury, Brookfield, VT, pp. 59-77.
- Mumford, E. (2000a) "Socio-technical Design: An Unfulfilled Promise or a Future Opportunity." In R. Baskerville, Stage, J., and DeGross, J.I (Ed.), *Organizational and Social Perspectives on Information Technology*, Kluwer Academic Publications, Boston, pp. 33-46.

- Mumford, E. (2000b), "A Socio-technical Approach to Systems Design," *Requirements Engineering*, 5(2), pp. 59-77.
- Nardi, B. (1996a) "Activity Theory and Human-Computer Interaction." In B. Nardi (Ed.), *Context and Consciousness: Activity Theory and Human-Computer Interaction*, The MIT Press, Cambridge, MA, pp. 7-16.
- Nardi, B. (1996b) *Context and Consciousness and Consciousness: Activity Theory and Human-Computer Interaction*, The MIT Press, Cambridge, MA.
- Nardi, B. A. (1997) "The Use of Ethnographic Methods in Design and Evaluation." In M. G. Helander, T. K. Landauer, & P. V. Prabhu (Eds.), *Handbook of Human-Computer Interaction, Volume 1*, Elsevier Science B.V., Amsterdam, pp. 361-366.
- Nardi, B. A., & O'Day, V. L. (1999) *Information Ecologies: Using Technology with Heart*, First ed., The MIT Press, Cambridge, MA.
- National Council of Teachers of Mathematics (1989) *Curriculum and Evaluation Standards for School Mathematics*, National Council of Teachers of Mathematics.
- National Educational Association (2004) NEA MB: Long-Term Care Insurance - Glossary, available at <http://www.neamb.com/insurance/ltgpge.jsp#C>.
- Newell, A., & Simon, H. A. (1972) *Human Problem Solving*, Prentice-Hall, Englewood Cliffs, NJ.
- Newell, A. F., & Gregor, P. (1997) "Human Computer Interfaces for People with Disabilities." In M. G. Helander, T. K. Landauer, & P. V. Prabhu (Eds.), *Handbook of Human-Computer Interaction, Volume 1*, Elsevier Science B.V., Amsterdam, pp. 813-824.
- Newell, A. F. (2006) "A User Centered Approach to Supporting People with Cognitive Dysfunction," *ACM CHI 2006 Workshop on Designing Technology for People with Cognitive Impairments*, Montreal, Quebec, available at [http://www.cs.ubc.ca/~joanna/CHI2006Workshop\\_CognitiveTechnologies/positionPapers/22\\_Newell\\_CHIworkshop\\_keynote.pdf](http://www.cs.ubc.ca/~joanna/CHI2006Workshop_CognitiveTechnologies/positionPapers/22_Newell_CHIworkshop_keynote.pdf).
- Nickerson, R. S. (1993) "On the Distribution of Cognition: Some Reflections." In G. Salomon (Ed.), *Distributed Cognitions Psychological and Educational Considerations*, Cambridge University Press, Cambridge, UK, pp. 229-260.
- Norman, D. A. (1981) "Categorization of Action Slips," *Psychology Review*(88), pp. 1-15.
- Norman, D. A. (1983) "Design Rules Based on Analyses of Human Error," *Communications of the ACM*, 26(4), pp. 254-258.
- Norman, D. A. (1990) *The Design of Everyday Things*, Currency Doubleday, New York.
- Norman, D. A. (1993) *Things That Make Us Smart*, Addison-Wesley Publishing Company, Reading, MA.

- Norman, D. A., & Shallice, T. (2000) "Attention to Action: Willed and Automatic Control of Behavior." In M. Gazzaniga (Ed.), *Cognitive Neuroscience: A Reader*, Blackwell, Malden, MA, pp. 376-390.
- The Oregon Research Institute (1988) "Teaching People with Developmental Disabilities,"
- Ostwald, J., et al. (2003) DynaGloss, Part of the Dynasite System, available at [http://seed.cs.colorado.edu/dynagloss.MakeGlossaryPage.fcgi\\$URLinc=6](http://seed.cs.colorado.edu/dynagloss.MakeGlossaryPage.fcgi$URLinc=6).
- Overseas Development Administration, S. D. D. (1995) *Guidance Note On How To Do Stakeholder Analysis Of Aid Projects And Programmes*, Available at <http://www.euforic.org/gb/stake1.htm>.
- Packer, L. E. (2004) Overview of Executive Dysfunction, available at [http://www.tourettesyndrome.net/ef\\_overview.htm](http://www.tourettesyndrome.net/ef_overview.htm)
- Pea, R. (1993) "Practices of Distributed Intelligence and Designs for Education." In G. Salomon (Ed.), *Distributed Cognitions: Psychological and Educational Considerations*, Cambridge University Press, Cambridge, UK, pp. 47-87.
- Pea, R. D. (2004) "The Social and Technological Dimensions of Scaffolding and Related Theoretical Concepts for Learning, Education, and Human Activity," *The Journal of the Learning Sciences*, 13(3), pp. 423-451.
- Perkins, D. N. (1993) "Person-plus: A Distributed View of Thinking and Learning." In G. Solomon (Ed.), *Distributed Cognitions Psychological and Educational Considerations*, Cambridge University Press, Cambridge.UK, pp. 88-110.
- Perlis, A. J. (1982) "Epigrams on Programming." In *SIGPLAN Notices*, pp. 7-13.
- Perry, M. (2003) "Distributed Cognition." In J. M. Carroll (Ed.), *HCI Models, Theories and Frameworks, Toward a Multidisciplinary Science*, pp. 193-223. Morgan Kaufmann Publishers an Fransisco, CA
- Pfleeger, S. L. (2001) *Software Engineering - Theory and Practice*, Prentice Hall, Upper Saddle River, NJ.
- Phillips, B., & Zhao, H. (1993) "Predictors of Assistive Technology Abandonment," *Assistive Technology*, 5(1).
- Piuchno, R. A. (2003) "Enmeshed Lives: Adult Children with Developmental Disabilities and Their Aging Mothers.," *Psychology and Aging*, 18(4), pp. 851-857.
- Plato (2003) *Phaedrus*, Oxford University Press. Oxford, England
- Postman, N. (1985) *Amusing Ourselves to Death—Public Discourse in the Age of Show Business*, Penguin Books, New York.
- Preece, J., Rodgers, Y., Helen, S., Benyon, D., Hollannd, S., & Cary, T. (1994) *Human-Computer Interaction*, Addison-Wesley, Reading, MA.

- Reason, J. (1990) *Human Error*, Cambridge University Press, Cambridge, UK.
- Reed, R. W. (1989) *An Investigation of Two Prompting/Fading Procedures to Teach Independent Dire Evacuation Behaviors to Individuals with Severe/Profound Mental Retardation*, University of New Orleans.
- Rehabilitation Research Design & Disability (R<sub>2</sub>D<sub>2</sub>) Center (2006) Assistive Technology Outcomes Measurement System Project (ATOMS Project), available at <http://www.uwm.edu/CHS/r2d2/atoms/>.
- Reimer-Reiss, M. (2000) "Assistive Technology Discontinuance," *California State University, Northridge Center on Disabilities' 18th Annual International Technology and Persons with Disabilities Conference*, Los Angeles, CA
- RESNA (2004) RESNA Publications, available at <http://www.resna.org/ProfResources/Publications/Publications.php>.
- Rittel, H. (1984) "Second-Generation Design Methods." In N. Cross (Ed.), *Developments in Design Methodology*, John Wiley & Sons, New York, pp. 317-327.
- Rittel, H., & Webber, M. M. (1984) "Planning Problems Are Wicked Problems." In N. Cross (Ed.), *Developments in Design Methodology*, John Wiley & Sons, New York, pp. 135-144.
- Robert III, H. M., Evans, W. J., Honemann, D. H., & Balch, T. J. (2000) *Robert's Rules of Order, 10th Edition*, Perseus Publishing, New York, NY
- Rogers, E. M. (1995) *Diffusion of Innovations*, Fourth ed., The Free Press of Glencoe, New York.
- Rosson, M. B., & Carrol, J. M. (2002) *Usability Engineering*, Academic Press, San Francisco.
- Russel, J. N., Hendershot, G. E., LeClerer, F., Jean, H., & Adler, M. (1997) *Trends and Differential Use of Assistive Technology Devices: United States: 1994: Advanced Data from the Centers for Disease Control and Prevention - National Center for Health Statistics*.
- Salomon, G. (Ed.) (1993) *Distributed Cognitions: Psychological and Educational Considerations*, Cambridge University Press, Cambridge, UK.
- Saskatchewan Learning - Special Education Unit (2003) Task Analysis, available at <http://www.sasked.gov.sk.ca/k/pecs/se/docs/meeting/s6analysis.html>.
- Schank, R. C., & Abelson, R. P. (1977) *Scripts, Plans, Goals, and Understanding*, Lawrence Erlbaum Associates, Inc., Hillsdale, NJ.
- Scherer, M. J. (1996) *Living in the State of Stuck: How Technology Impacts the Lives of People with Disabilities*, Second ed., Brookline Books, Cambridge, MA

- Scherer, M. J., & Galvin, J. C. (1996) "An Outcomes Perspective of Quality Pathways to the Most Appropriate Technology." In M. J. Scherer, & J. C. Galvin (Eds.), *Evaluating, Selecting and Using Appropriate Assistive Technology*, Aspen Publishers, Inc, Gaithersburg, MD., pp. 1-26.\
- Schuler, D., & Namioka, A. (Eds.) (1993) *Participatory Design: Principles and Practices*, Lawrence Erlbaum Associates, Hillsdale, NJ.
- Schön, D. A. (1983) *The Reflective Practitioner: How Professionals Think in Action*, Basic Books, New York.
- SIGCAPH (2004) Assets SIGCAPH Biannual Conference, available at <http://www.acm.org/sigcaph/conferences/>.
- Simon, H. A. (1982) *Models of Bounded Rationality, Volume 1: Economic Analysis and Public Policy*, The MIT Press, Cambridge, MA.
- Simon, H. A. (1984) "The Structure of Ill-structured Problems." In N. Cross (Ed.), *Developments in Design Methodology*, John Wiley & Sons, New York, pp. 145-166.
- Snell, M. E. (1987) *Systematic Instruction of Persons with Severe Handicaps*, Merrill Publishing Company, Columbus, OH.
- Sparrow, S., & Cicchetti, D. (1985) "Diagnostic Uses of the Vineland Adaptive Behavior Scales," *Journal of Pediatric Psychology*, 10 (2), pp. 215-225.
- Stephenson, J., & Linfoot, K. (1996) "Pictures as Communication Symbols for Students with Severe Intellectual Disability," *AAC Augmentative and Alternative Communication* 12 (December 1996), pp. 224-255.
- Suchman, L. (1987) *Plans and Situated Actions: The Problem of Human-Machine Communication*, Cambridge University Press, Cambridge, UK.
- Sullivan, J., & Fischer, G. (2003) "Mobile Architectures and Prototypes to Assist Persons with Cognitive Disabilities Using Public Transportation," *26th International Conference on Technology and Rehabilitation*, Atlanta GA (CD archive).
- Sullivan, J. F. (2005) *Mobility for All*, available at <http://l3d.cs.colorado.edu/clever/projects/mobility.html>.
- Svensk, A. (2001) *Design for Cognitive Assistance*, Lund University. Lund, Sweden
- Terveen, L. G. (1995) "An Overview of Human-Computer Collaboration," *Knowledge-Based Systems Journal, Special Issue on Human-Computer Collaboration*, 8(2-3), pp. 67-81.

Trungpa, C. (1988 (reprint)) *The Myth of Freedom and the Way of Meditation*, Shambhala, Boston.

US Technology-related Assistance for Individuals with Disabilities Act (1988), Pub. L. 100-407, Section 3.1, "The US technology-related assistance for individuals with disabilities act of 1988,"

Wechsler, D. (1997) "Wechsler Adult Intelligence Scale (WAIS)," The Psychological Corporation, San Antonio, Texas

Yates, F. A. (1966) *The Art of Memory*, University of Chicago Press, Chicago.

Ye, Y. (2001) *Supporting Component-Based Software Development with Active Component Repository Systems*, Ph.D. Dissertation, University of Colorado at Boulder.

Ye, Y., & Fischer, G. (2002) "Supporting Reuse by Delivering Task-Relevant and Personalized Information," *Proceedings of 2002 International Conference on Software Engineering (ICSE'02)*, Orlando, FL, pp. 513-523.



### 13. Appendices

This section contains supporting documents that are either too large to be inserted in the text in-place or have a level of detail that would interrupt the flow of the discourse. Each appendix is prefaced with a short introduction to the material as well as an explanation of the details of the document. All of these documents are the original ones used in the study, lightly edited for readability.

<i>APPENDIX</i>	<i>PURPOSE</i>
<i>Appendix A – Initial Interview with Experts: Topics</i>	<i>To gain an understanding of domain</i>
<i>Appendix B – Usability Test</i>	<i>To support the design and evaluation of MAPS script design environment</i>
<i>Appendix C – Usability Evaluation Sheet</i>	<i>A form to record the specifics of use problems and the details of the think-aloud process in usability studies of the script development environment</i>
<i>Appendix D – Caregivers’ Expectations</i>	<i>Outline of realistic study</i>
<i>Appendix E – Task Analysis Checklist</i>	<i>Guide for script design learning session</i>
<i>Appendix F – Script Template Form</i>	<i>The template for designing scripts</i>
<i>Appendix G – Field Trials Exit Interview</i>	<i>The exit interview topics for realistic use evaluation</i>
<i>Appendix H – MAPS Database Schema</i>	<i>The database scheme that supports MAPS</i>
<i>Appendix I – MAPS-Prompter Log</i>	<i>The log created on the prompter after each script use</i>
<i>Appendix J – Basic Use of Prompter Test – Glider</i>	<i>The details of the initial prompter use test</i>
<i>Appendix K – Glider Test Evaluation Form</i>	<i>A template for evaluation of the initial prompter test</i>
<i>Appendix L – Script Sample – Linear Script</i>	<i>A typical script</i>
<i>Appendix M – Script Sample Using Forking and Looping</i>	<i>A typical multi-scripting script (i.e. multiple scripts that support forking and looping)</i>
<i>Appendix N – Script Designed for Collapsing</i>	<i>An example of scaffolding collapsing in a MAPS script</i>
<i>Appendix O – Ethnographic Analysis Matrix</i>	<i>An intermediate form of the analysis the ethnographic coding</i>
<i>Appendix P – Caregivers Crib Sheet Example</i>	<i>A document that caregivers made to support certain MAPS tasks</i>

### 13.1. Appendix A – Initial Interview with Experts: Topics

*This document was used as the starting point for the initial research into the domain of AT, persons with cognitive disabilities, and support communities with local experts at the University of Colorado. Their expertise spanned dyslexia and communication disabilities, to assistive technology developers, to special education assistive technology professionals. Each interview was semi-structured and began by using the following as initial discussion points. (See section 6.1.2.1.)*

I would like to cover the following areas in an interview with you:

- What is a cognitive disability?
- How does it differ from a sensory disability?
- How are they similar -- can a sensory disability lead to a cognitive disability?
- How does an assistive technology designed for different disabilities (the two categories above) differ?
  - Discuss the difference between learning tools and living tools.
- What do you see is most needed in this area (i.e. tools to compensate for cognitive disabilities)?
  - A brief discussion of:
    - Autism
    - Downs
    - Dyslexia
    - Cerebral Palsy
- What would be important research directions for this project ?

## 13.2. Appendix B – Usability Test

*This document was the set of instructions for the tester to use during the usability testing of the MAPS design environment (referred to here as the editor) (see appendix 13.4 and section 6.3). The tester also had an Excel spreadsheet form to record the specifics of use problems and the details of the think-aloud process (see Appendix 13.3).*

9/21/04

### Usability testing of MAPS Caregivers Script-editor

Procedure:

- 1) Get permission forms signed
- 2) Instruct the user
  - a. about MAPS
    - i. explain the MAPS/LifeLine concept
    - ii. explain the MAPS editor concept
    - iii. explain the basic function of the MAPS editor: (don't spend more than 3 minutes on these- very brief)
      1. basic metaphor (mention PowerPoint)
      2. how mouse can interact (i.e. clicking may insert image)
      3. menus – point them out (but don't open them)
      4. dual image preview/ script preview & save window
    - iv. have them look at two of the help videos (not the one of the add a prompt)
  - b. About think aloud method: The basic idea of thinking aloud is very simple. You ask your users to perform a test task, but you also ask them to talk to you while they work on it. Ask them to tell you what they are thinking: what they are trying to do, questions that arise as they work, things they read. The basic instructions can be very simple: *"Tell me what you are thinking about as you work."* People can respond easily to this, especially if you suggest a few categories of thoughts as examples: things they find confusing, decisions they are making, and the like. There are some other points you should add. Tell the user that you are not interested in their secret thoughts but only in what they are thinking about their task. Make clear that it is the system, not the user, that is being tested, so that if they have trouble it's the system's problem, not theirs. Also say: *Please don't stop your flow to tell me your thoughts, you can just talk as you work, it doesn't have to be complete sentences.*
- 3) Actual experiment
  - a. task portion
    - i. instructions to experimenters:

1. time tasks
2. note problems
3. offer help in non directed way:

So say things that encourage them to talk, but that do not direct what they should say. Good choices are "Tell me what you are thinking" or "Keep talking". Bad choices would be "What do you think those prompts about frammis mean?" or "Why did you do that?"

4. Offer wizard of OZ type support if a feature is needed that is not yet implemented (or we did not see a need for in the initial design). i.e. . This is the WIZARD OF OZ method: you get a person to emulate unimplemented functions and generate the feedback users should see. What if a task involves user input that is their free choice, like a name to use for a file? You can't anticipate what the users will type and put it on your mockup screens. But you can let them make their choice and then say something like, "You called your file 'eggplant'; in this mockup we used 'broccoli'. Let's pretend you chose 'broccoli' and carry on." – This may be necessary for image or verbal prompt file creation, storage or retrieval

ii. Tasks:

1. Open an existing script and make yourself familiar with it (you could preview it in the little window I showed you) (this involves loading a script from the database & running it in the previewer) (have them use the Chinese cookies)
2. Load an existing script (the Glider one) change it by adding a prompt and by changing an image and save it as Glider1.
3. Create a script (choose either shopping script or bus trip script) and save it. Make the script only 7 steps long.
4. use the user options:
  - a. change the sounds preference to 'double click to insert'
  - b. change the autosave options to 'autosave off'
5. Load the McDonald template script, change 3 of the prompts by adding the right image and sound, and save it as MyMcDonald. (I will give you some more instructions about the 'right' image and sound as you do this step.

b. narrative portion

- i. Biggest possible failure point for this system and their needs
- ii. Ask them about:
  1. menu system
  2. help system
  3. MAPS concept (i.e. database & PC → handheld concepts)

4) Caregivers questions

- a. Their sense of how this could work with their situation
  - i. What would work very well
  - ii. What would not work well at all
- b. Script ideas

- i. Their idea for what would be a killer script for them: - *Please describe (in as much detail as you can) the one script that might make the most positive change in your life? AND -Please describe (in as much detail*

*as you can) the one script that might make the most difference in the life of the person you are the caregiver for?*

- ii. Three other good ideas of good scripts:
  1. names/ functions of the scripts
  2. a little scenario level description of how they could work

### 13.3. Appendix C – Usability Evaluation Sheet

*This document is the form to record the specifics of use problems and the details of the think-aloud process for the MAPS design environment evaluation (see appendix B and section 6.3).*

Page 1 of 2                      MAPS - caregiver GUI                      Date: \_\_\_\_\_  
 Location: \_\_\_\_\_  
 Investigator: \_\_\_\_\_                      Participant ID: \_\_\_\_\_

Skill	Not an issue	Moderate issue	Kept user from being successful	Comments
Create a script				
Modify a script				
Load/Use Template				
Load a script				
image controls				
sound controls				
ppt view				
script preview box				
image preview/select box				
Menu items				
Number of requests for help :				

Page 2 of 2                      MAPS - caregiver GUI  
 Participant ID: \_\_\_\_\_ Date: \_\_\_\_\_

**General Observations:** \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

**Specific control issues:** \_\_\_\_\_  
 \_\_\_\_\_

**Specific model issues:** \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

### 13.4. Appendix D – Caregivers’ Expectations

*This document outlined the expected hours the realistic assessment would take for caregiver and client. It was handed to the dyad at the first, “get to know you” session. The number of hours was a realistic estimation of the time needed from the dyad but most of the dyads ended up spreading the tasks over many more days than the two weeks predicted. (See section 6.6.)*

11/29/05

#### Caregiver’s handout for Naturalistic MAPS system testing

I have invited you and your child to participate in testing and evaluating the MAPS prompting system. In order to do this I will initially spend some time observing your day-to-day life, in particular how you together accomplish daily tasks. Next I will introduce the MAPS system, provide training for you and your child and finally observe all of you using the system to accomplish real tasks. The table below summarizes my initial expectation of the time needed:

Task	Who	Time expected
1) Observe daily life and routines	C, P	Two days of 4 to 6 hours ‘hanging out’
2) Intro to prompting	C	One hour session
a) Prompting		
b) Task\ segmentation		
3) Intro to the caregivers editor	C	One 2 hour session
a) Caregivers usability tests		
4) Discuss what tasks would be appropriate	C, P	One 2 hour session
5) Choose a task	C, P	One hour session
6) Intro to the handheld prompter		
a) Glider test	P	½ hour
b) Cookie test	P	One Hour session
7) Make task 1 (semi-laboratory)	C	From 2 to 4 hours depending
8) Use prompter to do task 1 (repeatedly)	P	Less than one hour (but repeated)
9) Make task 2 (semi-laboratory)	C	2 hours
10) Use prompter to do task 2	P	Less than one hour (but repeated)
11) Make task 3 (realistic)	C	4 – 6 ours spread out over several days
12) Use prompter to do task 3	P	Between one to 2 hours

(C= caregivers, P= person with cognitive disabilities)

I would prefer to do all of the testing in one two week period, but your vacation and other plans may prevent this, so I could break the tasks up and do them when it would best suit your schedule.

I realize participating in the MAPS testing represents both a large commitment of your time and that having me in your home “observing” your day to day life may be burdensome and I promise to minimize intrusion into your life.

Thanks,

Stefan Carmien



### 13.5. Appendix E – Task Analysis Checklist

*This document was the outline of topics to be covered in the one-hour task segmentation tutorial that was given to each caregiver before they started to create scripts. (See section 6.6.)*

11/15/05

#### Agenda for task segmentation session

What I want to learn:

- 1) How do typical tasks get done before the introduction of MAPS
  - a) What is the model of what they are doing to support tasks currently:
    - i) How they trigger parts of internal scripts
    - ii) What internal scripts exist currently
    - iii) How they grow a new task
    - iv) How they correct & learn
- 2) I want to have an idea of what tasks they (person with cognitive disabilities & caregiver) think might be useful & doable

Material to cover:

- 1) Basics of task segmentation:
  - a) Video
  - b) Examples from PDF
- 2) Make a list of script ideas
- 3) Choose one for script one
  - a) Do rough break down and discuss this in light of video
  - b) Talk about how we will photo idea one
  - c) Talk about how to record idea one
- 4) Set date to train in the software
- 5) Set date to start making script 1 (photos)

### 13.6. Appendix F – Script Template Form

*This form was used by the caregivers to do preliminary design of a script. It has columns for the words of the verbal prompt, for the file name of the image that will be used with that verbal prompt to make a step of the script, and finally a column for notes about that step. These were distributed to the caregivers as blanks, with empty columns, with only the step numbers on the form. In this partially filled out example, the caregiver has filled out the verbiage of the strip. After filling out the name of the file of the image and recording the prompts, the caregiver can then go to the script- design- environment and assemble the script. (See section 6.6.)*

5/3/06

## Script for Hamburger helper

Step #	Verbal Prompt	Image Name	Notes
1	Hi [REDACTED] today you are making Potatoes and garlic hamburger helper		
2	You will need the box of hamburger helper		
3	And hamburger		
4	Get the frying pan		
5	Get the 2 cup measure		
6	And the 4 cup measure		
7	Put two and three fourths cups water in the 4 cup measure		Added 4 to the prompt
8	Put the cup measure in microwave for five minutes		
9	Put hamburger in pan on stove and turn on medium high		
10	While browning the hamburger, open the box and put sauce packet and potato packet on counter		
11	Get two cup measure and pour three fourths cup milk into it and leave on counter		
12	After hamburger is browned		
13	Pour in hot water		
14	Pour in milk		
15	Open packet and pour in sauce packet		

16	Open packet and pour in potatoes		
17	Stir mixture together		
18	Cover the pan and lower heat to low		
19	Set timer for sixteen minutes		
20	When timer goes off – uncover pan and turn off stove		
21	Let the food in pan sit for 2 minutes while you get the plates ready		
22	Eat when you are ready		
23	Good job, [REDACTED] you made dinner!		

### 13.7. Appendix G – Field Trials Exit Interview

*This document outlines the semi-structured interview done with the caregivers (and in some cases the clients) at the end of the realistic field trials. (See section 6.7.)*

#### MAPS Exit interview (semi structured)

Template (replace actual script names for Script X):

I thought we could review each of the steps of this journey-

- 1) The intro part
- 2) The prompting class
- 3) Learning MAPS editor
- 4) Making Script 1
- 5) Doing Script 2
- 6) Making Script 2
- 7) Doing Script 2
- 8) Making Script 3
- 9) Doing Script 3

For you:

- What worked
- What did not work
- What was great about the MAPS editor
- What was awful about the MAPS editor
- Other scripting ideas we could have explored
- Will you use this on your own
  - If so, how?
  - If not, why?
- What would a reasonable set of expectations for my ongoing support of MAPS for 7X & 6X?

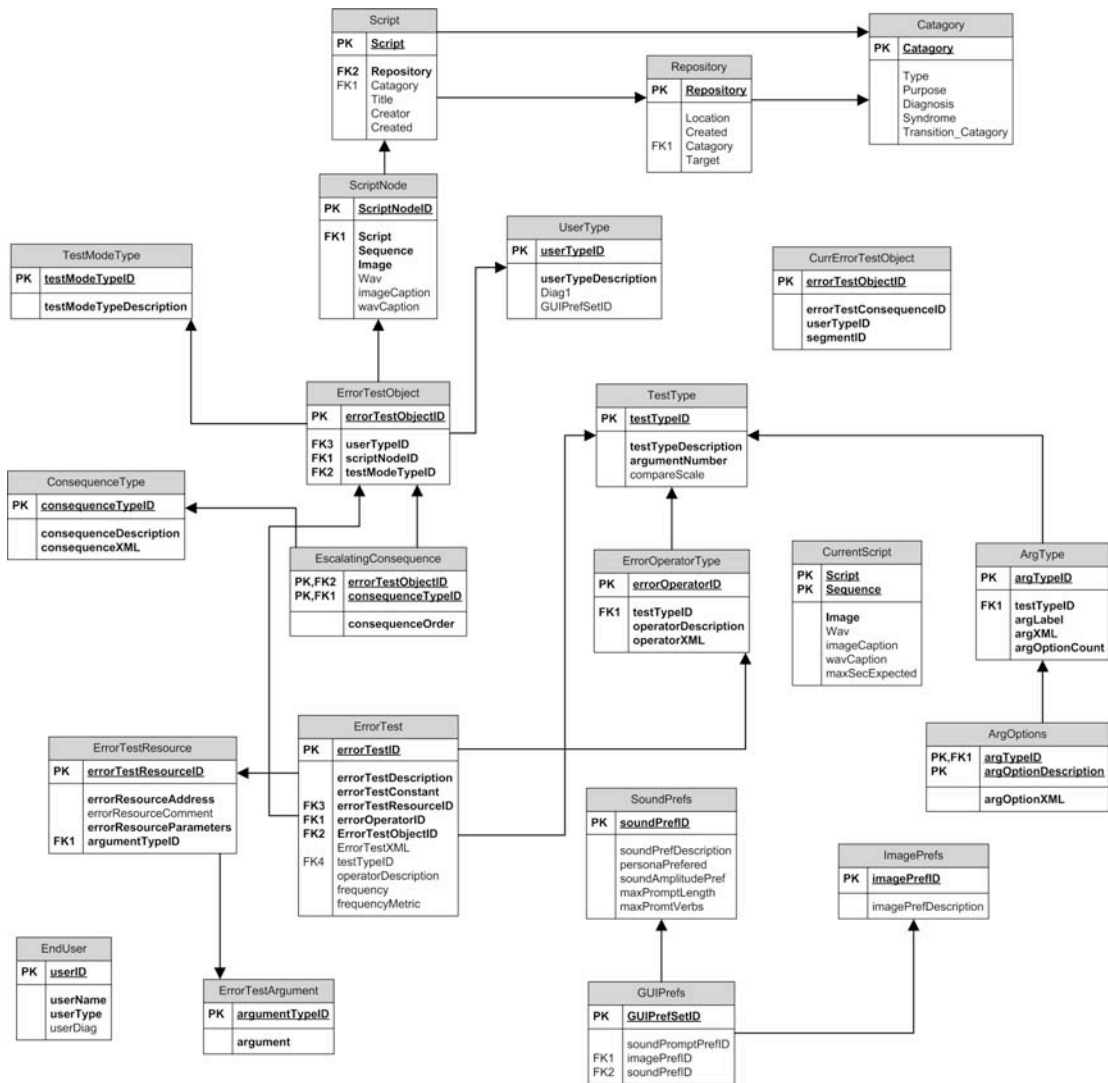
For 6X:

- How she felt about the prompting

- How she felt about the prompter
- Using it:
  - o To do housekeeping task
  - o To cook
  - o To shop
- Other ideas she might have for MAPS
- What she hated about MAPS
- What she liked about MAPS
- What was she using it for in the days when she took it to school
- Can I help her get started using it for a MP3 player?

### 13.8. Appendix H – MAPS Database Schema

The following E-R diagram of the MAPS database shows the tables, fields, and relationships between the elements of the database. The upper right tables (ScriptNode, Script, Repository, Category) are the current set used by the script design environment to create and store scripts. The prompter uses the currentScript table to display and “play” scripts. The rest of the tables support the error trapping and error correction functions in the advanced user mode of the script design environment and prompter. (See section 5.4.3.)



### 13.9. Appendix I – MAPS-Prompter Log

*This document is the log file that the MAPS prompter produces when a script is played. The first line of text contains the date stamp of when the script started to run, followed by a list of the steps in this script. After this, line by line is a listing of each prompt step as it is played on the prompter, with a date, then the time, then the script prompt sequence number (i.e., 0 is the first, 2 is the third, etc.), followed by the file name of the verbal prompt (which should be descriptive of that step in the script). (See section 5.4.1.)*

logText=11/8/05 03:29:32 PM,"■■■■ shopping", "Hi ■■■■ lets go shopping .wav", "Look for the bags of salad .wav", "Get a bag of Caesar salad .wav", "Next put a bunch of broccoli in a plastic bag.wav", "Go to isle 15 please .wav", "put a jar off passta sauce in the basket.wav", "go to isle 12.wav", "Pick out a bottle of light syrup.wav", "Go to isle six .wav", "Get a box of tissues .wav", "Walk over to isle two .wav", "Get your favorite flavor of light ice cream.wav", "Turn in by the ice cream.wav", "Put a box of waffles in your baske .wav", "get a gallon of milk with a yellow top.wav", "Go to the bakery sign .wav", "You need a bag of hamburger buns.wav", "Go to the check out lane .wav", "Give the chekcer the safeway card.wav", "Give the checker the money .wav", "Thanks ■■■■ you did a great job shopping.wav"

11/8/05 03:29:36 PM,0,"Hi ■■■■ lets go shopping .wav"  
 11/8/05 03:30:46 PM,1,"Look for the bags of salad .wav"  
 11/8/05 03:31:02 PM,2,"Get a bag of Caesar salad .wav"  
 11/8/05 03:31:48 PM,3,"Next put a bunch of broccoli in a plastic bag.wav"  
 11/8/05 03:31:59 PM,2,"Get a bag of Caesar salad .wav"  
 11/8/05 03:33:08 PM,3,"Next put a bunch of broccoli in a plastic bag.wav"  
 11/8/05 03:34:43 PM,4,"Go to isle 15 please .wav"  
 11/8/05 03:35:04 PM,5,"put a jar off passta sauce in the basket.wav"  
 11/8/05 03:35:40 PM,6,"go to isle 12.wav"  
 11/8/05 03:36:00 PM,7,"Pick out a bottle of light syrup.wav"  
 11/8/05 03:36:33 PM,8,"Go to isle six .wav"  
 11/8/05 03:37:02 PM,9,"Get a box of tissues .wav"  
 11/8/05 03:37:37 PM,10,"Walk over to isle two .wav"  
 11/8/05 03:37:57 PM,11,"Get your favorite flavor of light ice cream.wav"  
 11/8/05 03:38:28 PM,12,"Turn in by the ice cream.wav"  
 11/8/05 03:39:56 PM,13,"Put a box of waffles in your baske .wav"

11/8/05 03:41:46 PM,14,"get a gallon of milk with a yellow top.wav"  
11/8/05 03:42:16 PM,15,"Go to the bakery sign .wav"  
11/8/05 03:42:41 PM,16,"You need a bag of hamburger buns.wav"  
11/8/05 03:42:42 PM,17,"Go to the check out lane .wav"  
11/8/05 03:42:45 PM,16,"You need a bag of hamburger buns.wav"  
11/8/05 03:43:56 PM,17,"Go to the check out lane .wav"  
11/8/05 03:45:21 PM,18,"Give the chekcer the safeway card.wav"  
11/8/05 03:45:29 PM,19,"Give the checker the money .wav"  
11/8/05 03:45:30 PM,20,"Thanks [REDACTED] you did a great job shopping.wav"



### 13.10. Appendix J – Basic Use of Prompter Test – Glider

*This is the template for the script used in the preliminary test of multimedia prompting by young adults with cognitive disabilities. The eight-step script guided the user through assembling the plastic glider. Below this is the text read to the participants in the experiment. This script constituted the only instructions given to the participants. (See section 6.2.)*

5/2/03

#### Instructions Version Of Glider Assembly Script

	Prompt	Image
1	Take the glider parts out of the bag.	Clear plastic bag with parts
2	Make sure you have all five parts	Parts laid you on table top next to bag
3	Slide the big wing thru the big slot in the plane until it is in the center. Be careful when you slide it in.	Just this w/ hand
4	Slide the small wing thru the small slot in the back of the plane	Just this w/ hand
5	Put the tail wing on top of the back of the wing in the small slot	Just this w/ hand
6	Put the pilot on the top of the plane	Just this w/ hand
7	Hold the plane with your fingers and throw it	Just this w/ hand
8	Have fun the glider is yours to keep.	Glider in mid-air.

#### MAPS Prompter basic controls script (read to participants)

Hi, my name is Anja. Do you want to build an airplane with me? I have a cool new tool that will show you how you can make an airplane you can keep.

This is PDA and it shows directions on how to do things.

Like here it's telling me what I need to buy at the grocery store.

I press the screen to make it go to the next page.

If I want to go back and check the page before I press here.

And if I have a problem, lets say I wasn't sure what kind of \_\_\_\_\_ I was supposed to buy I could press this button and someone would help me.

Do you want to try it out a little yourself?

0

Ok, now lets have it show me how to make an airplane and then afterwards you can make one.

>Back Tail< Hit panic button

>Pilot< Think I skipped something, go back and check

Ok, now it's your turn. If you have problem just hit this button and I'll help you.

### 13.11. Appendix K – Glider Test Evaluation Form

*The following forms were used to evaluate the participants in the preliminary test of multimedia prompting by young adults with cognitive disabilities. The evaluation was focused on specific abilities with the aim of developing a sense of ability-based determination of whether the subject could use the prompter. (See section 6.2.)*

Page 1 of 2

MAPS - general usability testing

Investigator: \_\_\_\_\_

Participant ID: \_\_\_\_ Date: \_\_\_\_\_

Caregiver ID: \_\_\_\_\_

Skill	Not an issue	Moderate issue	Kept student from being successful	Prompt #	Specific Documented Disability	Comments
Visual discrimination of prompt						
Visual comprehension of prompt						
Auditory perception of prompt						
Auditory comprehension of prompt						
Fine motor accessibility to MAPS screen						
Fine motor accessibility to Forward button						
Fine motor accessibility to Panic button						
Fine motor accessibility to Back button						
Comprehension of button navigation						
Successful completion of task						

Number of requests for help using Panic button:	_____
Number of requests for help without use of Panic button:	_____
Number of unrequested interventions:	_____

MAPS - general usability testing

Participant ID: \_\_\_\_\_ Date: \_\_\_\_\_

Caregiver ID: \_\_\_\_\_

General Observations: \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Documented Disability: \_\_\_\_\_

Observations of Controls: \_\_\_\_\_

Observations of MAPS "model" comprehension: \_\_\_\_\_








**13.12. Appendix L – Script Sample – Linear Script**








*There is an example of a straightforward linear (i.e., no forking or looping) script. In this case, it is the third script of the second dyad drawn from the pool of families in special education with the BVSD. The first column is the prompt sequence number, the second is the words in that step, and the rightmost column is the image that was displayed on the prompter. (See section 6.6.1.)*

5/3/06

Script for  
Hamburger helper

Step Number	Verbal Prompt	Notes
1	Hi [redacted] today you are making Potatoes and garlic hamburger helper	
2	You will need the box of hamburger helper	
3	And hamburger	
4	Get the frying pan	
5	Get the 2 cup measure	
6	And the 4 cup measure	

7	Put two and three fourths cups water in the 4 cup measure	Added 4 to the 
8	Put the cup measure in microwave for five minutes	prompt 
9	Put hamburger in pan on stove and turn on medium high	
10	While browning the hamburger, open the box and put sauce packet and potato packet on counter	
11	Get two cup measure and pour three fourths cup milk into it and leave on counter	
12	After hamburger is browned	
13	Pour in hot water	
14	Pour in milk	

15	Open packet and pour in sauce packet	
16	Open packet and pour in potatoes	
17	Stir mixture together	
18	Cover the pan and lower heat to low	
19	Set timer for sixteen minutes	
20	When timer goes off – uncover pan and turn off stove	
21	Let the food in pan sit for 2 minutes while you get the plates ready	
22	Eat when you are ready	
23	Good job, ***** you made dinner!	
24		
25		
26		
27		
28		

### 13.13. Appendix M – Script Sample Using Forking and Looping

*This is an example of a script that uses forking and looping. The script is composed of three sub-scripts. At step 30 of script 1, the user is given the opportunity to loop back over this script. At steps 32 and 33, the user is presented with forking possibilities. Similarly in script 2 or 3, step 11 presents further opportunities of forking. (See section 5.3.3 and 6.6.3.)*

#### ■ Script one

2/20/05

I need a color sheet that I can use for their odd names & color issues.

**Colorizing in Each Size** < We should have a 'demo' script to show ■ how to use the IP/AQ – not here >

Step number	Verbal prompt	Image name	Notes
1	"Let's colorize this rack"	Color Order	Picture of rack with sorted clothes exactly the opposite – rack with mixed up clothes? >
2	"Yellow"	Yellow.bmg	<Need to shot or (better) have a copy of the actual color scheme to copy to prompts -
3	"Orange"	Orange.bmg	
4	"Red"	Red.bmg	
5	"Burgandy"	Burgandy.bmg	
6	"Purple"	Purple.bmg	
7	"Pink"	Pink.bmg	
8	"Green"	Green.bmg	
9	"Aqua"	Aqua.bmg	
10	"Blue"	Blue.bmg	
11	"Beige"	Beige.bmg	
12	"Rust"	Rust.bmg	
13	"Brown"	Brown.bmg	
14	"Grey"	Grey.bmg	
15	"Black & White"	Black & White.bmg	
16	"Black"	Black.bmg	
17	"White"	White.bmg	



18	"Ask Supervisor which clothing to colorize"	Supervisors.bmp	Get a picture of all 3 supervisors together at register
19	"Get a rolling hanger rack"		Picture of rolling hanger rack
20	"Take rack to clothing area"		Picture of clothing area
21	"Start with 'Small' Size and work up"		Close up of small tag
22	"Start with yellow or first color in that size"		Picture of yellow clothing
23	"Take out any item not that size"		Picture of the wrong size tags
24	"Hang wrong size items on your rolling rack"		Picture of that fitted (or less) rolling rack
25	"Continue to put in color order"		Picture of rainbow of color
26	"Check that all clothing has the tag on the aisle side"		Picture of clothing with all having arm with tag on aisle
27	"This size is complete if it is in color order"		<This is murky to me, do we need this step?> Yes, keep it
28	"Look on the rolling rack for any clothing of next size" "Move to next size"		Picture of rolling rack (from before?) <how to deal with her being at large size- in this case there are no next size, yet we need steps 27-30 in all the other sizes>
29	"Take clothing from next size from rolling rack and put in color order"		<Generic prompt picture>
30	"If there are more sizes to colorize, press script re-start, else press screen"		
31	"This colorizing task is finished, good work!"		<Generic prompt picture>
32	"Here are some ideas for more to do before the supervisor comes by"		<Generic atagirl>
33	"Press script 2 for hangers or 3 for shelves task"		<Generic decision point icon>

**Hangar script # 2**

1	“Remove Empty Hangers”		Pict of empty hangers in a clothing section
2	“Get a rolling rack”		Pict of empty rolling rack (reuse previous one)
3	“Take out any empty hangers”		Pict of empty hangers in a clothing section
4	“Put hangers on rolling rack”		Picture of half filled rolling rack
5	“When rolling rack is full”		Picture of full rolling rack
6	“Roll rack to production in back of store”		Picture of double doors in back of store
7	“Turn left, take hangers from rolling rack and put on hanger racks”		Picture of how it should look when done
8	“Match sizes and types of hangers”		Picture of two different hangers
9	“This task is completed, good work”		Generic atagirl image here
10	“If it is Time to leave press on the screen”		<Generic loop point icon> could we not say do more hangar pulling?
11	“If the colorizing and hangar task has been done and it is still time to work, find a supervisor and ask what you can do now”		Picture of the supervisors

### Top Shelves #3

1	"Rag Top Shelves "		Picture of store shelves on top of clothing.
2	"Get a cart"		Picture of cart
3	"Choose one section of shelves"		<Generic picture? >
4	"Make sure all items on shelf are all the same – all wood, all holiday"		How to illustrate this <do we want to illustrate not matching or matching?>? put in picture of correct shelf
5	"Take any items that aren't the same of shelf, put in basket"		Picture of basket shelf with toys interspersed, ceramic items
6	"Make sure there are no empty spaces"		Picture of correct shelf – with ½ done sorting
7	"Spread items across shelf"		Show shelf with clumped items?
8	"Take items from basket and put on their correct shelf"		Pret of basket shelf with toys & ceramics mixed Hand putting toy on shelf with toys
9	"This task is completed, good work"		Generic atagirl image here
10	"If it is Time to leave press on the screen"		<Generic loop point icon> could we not say do more shelf straightening?
11	"If the colorizing and shelf task has been done and it is still time to work, find a supervisor and ask what you can do now"		Picture of the supervisors

#### Questions:

- 1) We still need the 'official' color sheet
- 2) I think it would be a good idea to do this before giving it to her

### 13.14. Appendix N – Script Designed for Collapsing

*This is an example of a script designed for collapsing as portions are internalized. This particular document is a snapshot of the script in the midst of the design process, without having worked out the exact verbiage and image for each step. The caregiver has identified each step with the appropriate notes for each internal script that she has inventoried of her client in the fourth column from the left. In the rightmost column, she has indicated whether that step could stand for the steps following when the client learned that set of steps. The step with “N” above the steps with “Y” stood for the ones directly below.. An example of this is step 2, “fold T-Shirts,” which would be sufficient to trigger the behavior in steps 3 through 8, which the client might make into an internal script that then would be part of the client’s inventory of internal scripts that could be invoked with a single prompt. (See sections 6.6.3, 5.1.4.2, and 5.3.3.)*

3/10/06

#### Laundry script

#	Verbal prompt	Image name	Notes	Collaps-able?
1			Unload your laundry basket making piles of "like" clothes on your bed (t-shirts in one pile, socks in another, etc) <what are 'like' clothes -this is unclear>	N
2			<First> Fold T-Shirts	N
3			If it's inside out, put it right side in	Y
4			Lay it down flat on the bed with the front of the T-shirt facing down <Down is heard to understand how about facing the bed>	Y
5			Fold the T-shirt in half, so that the sleeves are aligned <Aligned is a difficult word>	Y
6			Fold the bottom of the T-shirt up so that it aligns with the top <Aligned is a difficult word>	Y
7			Fold over the sleeves	Y
8			Place folded T-shirts in drawer labeled "T-shirts"	Y
9			<Next>Fold Pants	N
10			<If> It it's inside out, put it right side in	Y
11			Lay pants down on bed with back pockets facing down <Down is heard to understand how about facing the bed>	Y
12			Fold one leg over the other	Y
13			Fold in half	Y
14			Place folded pants in drawer labeled "pants"	Y
15			<Next>Fold Underwear	N
16			If it's inside out, put it right side in	Y
17			Fold the underwear in half long-ways	Y
18			Fold the underwear up short-ways	Y
19			Place folded underwear in drawer	Y

			labeled "underwear"	
20			<Next> Fold Socks	N
21			If inside out, put them right side in	Y
22			Put pairs of like socks together	Y
23			Place one sock over the other sock so they are aligned <Aligned is a difficult word>	Y
24			Fold the pair in half long-ways	Y
25			Place folded sock pairs in drawer labeled "socks"	Y
26			<Next> Fold Thermals	N
27			Thermal Pants: (follow instructions for "Fold Pants")	Y
28			Thermal Shirts: (follow instructions for "Fold T-Shirts")	Y
29			Place folded thermals in drawer labeled "thermals"	Y
30			<Next> Hang Clothes in Closet	N
31			Put any sweatshirts, nice collar shirts, or jackets on a hanger <Nice is difficult to define>	Y
32			Place in closet <Not personal enough – sorta robotic>	Y
33			<Next> Fold Towels and Sheets	N
34			If any towels or sheets in hamper, lay them out flat on bed or floor <Again needs more words & less robo-like>	Y
35			Fold in half	Y
36			Fold in half again	Y
37			Place them in the closet next to the bathroom	Y

### 13.15. Appendix O - Ethnographic Analysis Matrix

The following pages contain a summary of the coding of the realistic test field notes and transcriptions. The column at the left identifies which dyad the results come from. The order of the constituting parts of the dyad is not significant (i.e., 63 and 73 is not different than 73 and 63). In the top row are the codes that selected parts of the field notes/transcripts were assigned to. There can be multiple codes assigned to a given text fragment. After the code assignment, the count of code/dyad is in the intersection of the row and column. By comparing like events with differing antecedents/results, conjectures could be asserted. An example of this is evaluation of interventions by caregivers. All of these were coded with “caregiver and pwcd script interaction-intervention” (where “pwcd” stands for person with cognitive disabilities) and, after ruling out two observations as not relevant, by referring back to the source text, a breakdown could be done of the circumstances of the intervention, thus allowing some tentative conclusions to be drawn. Similarly, by grouping several codes together that refer to a similar event or experience and breaking down the total into specific exemplars, nuances of use / experiences can be examined. An example of this is in the discussion on navigation in section 6.7.5, the codes of “person with cognitive abilities learning prompter controls” and “person with cognitive disabilities not remembering use of prompter controls” can be used to discuss the issue of navigation or use of the controls of the MAPS prompter (after one of the observations in the later coding being rejected as not relevant) in section 6.7.2.

CaseIDs	adaptations of environment for pwcd	additional dist cog artifact - in task environment	additional dist cog artifact - on prompter	additional PC setup needed
Dyad 63 and 73	0	1	7	0
Dyad 64 and 74	1	0	0	0
dyad 71 and 61	6	13	2	5
Dyad 72 and 62	0	0	2	0
TOTAL	7	14	11	5

CaseIDs	additional dist cog - for scripting	affordances of prompter	AT needs	caregiver occupation
Dyad 63 and 73	0	2	0	0
Dyad 64 and 74	0	0	1	0
dyad 71 and 61	1	1	1	0
Dyad 72 and 62	1	3	0	1
TOTAL	2	6	2	1

CaseIDs	Caregiver education	Caregiver MAPS failure	caregiver MAPS success story	caregiver needs of AT for pwcd
Dyad 63 and 73	0	0	7	1
Dyad 64 and 74	4	3	11	0
dyad 71 and 61	1	1	1	2
Dyad 72 and 62	1	9	11	2
TOTAL	6	13	30	5

CaseIDs	caregiver PC literacy	caregiver re. pwcd relation to MAPS and style	script learning how to do it Guided Use	script mismatch with task
Dyad 63 and 73	2	3	2	5
Dyad 64 and 74	3	2	0	2
dyad 71 and 61	3	0	2	20
Dyad 72 and 62	1	2	0	10
TOTAL	9	7	4	37

CaseIDs	caregiver day-to-day help given to PWCD	caregiver and MAPS editor use	caregiver and pwcd script interaction- intervention	script ideas
Dyad 63 and 73	0	1	3	8
Dyad 64 and 74	5	9	8	7
dyad 71 and 61	0	2	32	3
Dyad 72 and 62	0	2	14	12
TOTAL	5	14	57	30

CaseIDs	tics and preservations	critical IADL success script and task	diagnosis	image of prompts
Dyad 63 and 73	0	0	3	8
Dyad 64 and 74	0	0	1	6
dyad 71 and 61	0	1	3	2
Dyad 72 and 62	0	0	3	2
TOTAL	0	1	10	18

CaseIDs	caregiver script design	Caregiver script design lessons	caregiver script editing quirks	collapse script
Dyad 63 and 73	10	3	6	1
Dyad 64 and 74	13	6	7	3
dyad 71 and 61	6	0	10	9
Dyad 72 and 62	18	9	0	3
TOTAL	47	18	23	16
CaseIDs	caregivers expectations for pwcd employment	Caregivers helping pwcd with IADL	Caregivers PWCD safety issues	prompter additional functions - misc
Dyad 63 and 73	6	1	0	3
Dyad 64 and 74	0	0	0	1
dyad 71 and 61	3	2	2	0
Dyad 72 and 62	0	0	0	1
TOTAL	9	3	2	5
CaseIDs	caregiver script editor feature need	Caregivers daily schedule	caregivers expectations for MAPS	caregivers expectations for pwcd
Dyad 63 and 73	3	0	10	14
Dyad 64 and 74	2	0	5	6
dyad 71 and 61	10	3	8	9
Dyad 72 and 62	14	0	2	1
TOTAL	29	3	25	30
CaseIDs	intelligibility or volume of prompts	MAPS beta bugs	professional caregivers	prompter additional functions - MP3
Dyad 63 and 73	3	7	3	7
Dyad 64 and 74	5	16	2	1
dyad 71 and 61	7	12	1	0
Dyad 72 and 62	6	10	4	0
TOTAL	21	45	10	8
CaseIDs	pwcd communication	pwcd employment	fragile WINCE OS	generic prompts
Dyad 63 and 73	0	18	6	0
Dyad 64 and 74	3	14	4	0
dyad 71 and 61	18	5	1	1
Dyad 72 and 62	11	5	0	0
TOTAL	32	42	11	1



CaseIDs	double clicking	dynamics of dyad	evaluation of caregiver scripter interface	expanding scripts
Dyad 63 and 73	2	0	0	0
Dyad 64 and 74	10	0	0	2
dyad 71 and 61	18	1	0	5
Dyad 72 and 62	4	0	1	3
TOTAL	34	1	1	10

CaseIDs	holding IPAQ while doing task	person with cognitive not remembering use of prompter controls	playing with MAPS prompter	problems with IPAQ OS or Hardware
Dyad 63 and 73	7	1	0	0
Dyad 64 and 74	3	2	0	1
dyad 71 and 61	1	4	0	1
Dyad 72 and 62	2	4	0	0
TOTAL	13	11	0	2

CaseIDs	MAPS log	order of prompts	Parental power issues	person with cognitive abilities learning prompter controls
Dyad 63 and 73	0	1	3	14
Dyad 64 and 74	0	0	2	3
dyad 71 and 61	1	1	5	0
Dyad 72 and 62	0	0	11	3
TOTAL	1	2	21	

CaseIDs	pwcd error doing prompt	pwcd personality and behavior	PWCD prompter navigation problems	pwcd scheduled events
Dyad 63 and 73	0	16	2	0
Dyad 64 and 74	2	38	0	1
dyad 71 and 61	8	20	0	5
Dyad 72 and 62	5	31	1	1
TOTAL	15	105	3	7

CaseIDs	pwcd transition	pwcd using MAPS as TfLea	re-writing scripts	pwcd and family roles
Dyad 63 and 73	1	3	5	0
Dyad 64 and 74	0	6	2	12
dyad 71 and 61	3	2	2	1
Dyad 72 and 62	0	3	3	12
TOTAL	4	14	12	25
CaseIDs	Prompter backup and restore - batt issues	Prompting class	PWCD abilities	pwcd ADL support
Dyad 63 and 73	10	1	0	0
Dyad 64 and 74	0	0	2	1
dyad 71 and 61	0	1	0	1
Dyad 72 and 62	0	2	2	1
TOTAL	10	4	4	3
CaseIDs	pwcd asks for help with script	pwcd co-creating scripts	pwcd needs additional function in MAPS	PWCD not stopping at end of cued step
Dyad 63 and 73	0	3	0	1
Dyad 64 and 74	2	0	0	3
dyad 71 and 61	4	0	0	19
Dyad 72 and 62	2	0	1	10
TOTAL	8	3	1	33
CaseIDs	pwcd existing high technological devices	PWCD internal scripts	PWCD job coaching	pwcd learning style
Dyad 63 and 73	0	0	5	8
Dyad 64 and 74	0	6	0	7
dyad 71 and 61	0	12	7	0
Dyad 72 and 62	6	3	2	3
TOTAL	6	21	14	18
CaseIDs	pwcd opinion of a script	pwcd skills	pwcd skipped step - not double clicking	pwcd soft skills and prompting
Dyad 63 and 73	6	1	0	12
Dyad 64 and 74	4	9	3	1
dyad 71 and 61	1	1	0	0
Dyad 72 and 62	13	2	2	0
TOTAL	24	13	5	13

CaseIDs	pwcd schooling	pwcd script and or prompter use	pwcd script interleaving task or timing issue	PWCD siblings
Dyad 63 and 73	2	6	2	0
Dyad 64 and 74	2	2	0	2
dyad 71 and 61	1	0	7	1
Dyad 72 and 62	12	0	7	4
TOTAL	17	8	16	7

CaseIDs	Script - how long to make	Script collapse or expand	script forking and looping	tacit knowledge not encoded in script
Dyad 63 and 73	1	0	6	3
Dyad 64 and 74	5	0	0	0
dyad 71 and 61	2	0	0	3
Dyad 72 and 62	2	0	0	1
TOTAL	10	0	6	7

CaseIDs	recording prompt process confusing	relating to prompter as if caregivers was present	representation of script prompts	robotic voice on prompts
Dyad 63 and 73	4	1	2	6
Dyad 64 and 74	5	4	0	1
dyad 71 and 61	0	0	1	0
Dyad 72 and 62	1	0	6	0
TOTAL	10	5	9	7

CaseIDs	task choice	task segmentation	too low skill set of pwcd - cant use	too high skill set of pwcd - don't need
Dyad 63 and 73	2	8	2	3
Dyad 64 and 74	0	0	0	0
dyad 71 and 61	0	1	0	1
Dyad 72 and 62	0	9	0	0
TOTAL	2	18	2	4

CaseIDs	script missing step	script templates	script verbiage	scripts too small granularity
Dyad 63 and 73	2	1	6	0
Dyad 64 and 74	0	2	1	0
dyad 71 and 61	0	2	0	0
Dyad 72 and 62	6	3	1	3
TOTAL	8	8	8	3

CaseIDs	uneven abilities - day to day	video help	voice of prompts	wanted to keep system afterward
Dyad 63 and 73	3	4	3	0
Dyad 64 and 74	1	1	1	0
dyad 71 and 61	4	0	1	0
Dyad 72 and 62	5	2	4	0
TOTAL	13	7	9	0

### 13.16. Appendix P - Caregivers Crib Sheet Example

This image is an example of the sorts of secondary distributed cognition artifacts that the caregiver of the dyads produced as an aid in designing and using scripts. Every one of the dyads made some sort of intermediate artifact. This example is a crib sheet to aid caregiver 74 in using the MAPS prompter with her PC. Similarly, Figure 35 is a secondary artifact for the creation of scripts. At the end of the study, all of the crib sheets were put together in a document that was given to the caregivers to use in continuing to make scripts on their own, outside of the research.

