AUGMENTING ENVIRONMENTS WITH MULTIMODAL INTERACTION

David R. McGee

B.S., Washington State University, 1995

A dissertation presented to the faculty of the
OGI School of Science and Engineering
at Oregon Health & Science University
in partial fulfillment of the
requirements for the degree
Doctor of Philosophy
in
Computer Science and Engineering

June 2003

The dissertation "Augmenting Environments with Multimodal Interaction," by David R. McGee, has been examined and approved by the following Examination Committee:

Dr. Philip R. Cohen, Thesis advisor
Professor, OGI School of Science and Engineering

Dr. James Hook
Assoc. Professor and Chair, OGI School of Science and Engineering

Dr. Hiroshi Ishii
Professor, Massachusetts Institute of Technology

Dr. Sharon Oviatt
Professor, OGI School of Science and Engineering

Dr. Misha Pavel
Professor, OGI School of Science and Engineering

Dedication

I dedicate this dissertation to my family.

To my parents and my children: two generations that encouraged and sustained me throughout the trial;

Foremost, to my wife—who more than any other deserves what she gets—I'm home, Mary!

Acknowledgements

I wish first to thank the men and women of the Fifth and Eleventh Marine Corps Regiments, the First Marine Expeditionary Force, the Forty First Separate Infantry Brigade (Oregon Army National Guard) and all of the men and women of the Army, Army National Guard, and Marine Corps who aided me in my field work and participated in Rasa's usability experiments.

To all of my colleagues in the Command Post of the Future Program who treated me as the investigator I was striving to become, thank you. Steve Fisher (USMC, retired) encouraged and cajoled his Marine Corps colleagues and me, until the two parties, I hope, learned from each other. Thank you Steve. Col. Jack Thorpe (USAF, retired) I consider a valued mentor. Among many other things, I owe him the debt that all students of life owe.

Thank you, Dr. Donald Hanley, a true and steadfast friend, and a brilliant editor. I wish to also thank Chuck LoPresti, whose statistical prowess was welcome.

To my colleagues in the Center for Human-Computer Communication and the entire Computer Science and Engineering Department at the OGI School of Science & Engineering for their assistance, friendship, and tutelage. To the other developers of QuickSet from which Rasa is a descendant, I give my thanks. Jay Pittman, Ira Smith, Josh Clow, Michael Johnston, Matt Wesson and Lizhong Wu, we worked as friends and colleagues together to create something unique. You are all so talented. What a pleasure; I will miss those days. To my office partner and fellow student Sanjeev Kumar, I wish the greatest success as he approaches his terminus.

To my managers and other mentors at Battelle's Pacific Northwest Laboratory thank you for your support over the years. A special thanks to Ray Bair, Jim Thomas, Earl Heister, Kris Cook and Lee Ann Dudney. And to the rest of my colleagues at PNNL who encouraged me to step away from the laboratory for a few years and return, I hope, a much improved man.

Finally, I am honored to be guided by a brilliant advisor, Professor Philip Cohen, whom I admire, respect, and feel fortunate to call my colleague and my friend.

This work was supported in part by the Command Post of the Future Program, Information Systems Office of DARPA under contract number N66001-99-D-8503, and also in part by ONR grants: N00014-95-1-1164, N00014-99-1-0377, and N00014-99-1-0380.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Abstract

Augmenting Environments with Multimodal Interaction

David R. McGee, B.S.

Ph.D., OGI School of Science and Engineering,
Oregon Health and Science University
June 25, 2003

Thesis Advisor: Dr. Philip R. Cohen

In air-traffic control centers, military command posts, and hospital emergency rooms, life-and-death decisions must be made quickly in the face of uncertain information. In these high-stress environments, professionals put a premium on safety, timeliness, team cohesiveness, and mutual awareness. Consequently, these professionals discard computational tools in favor of those that are more robust, malleable, physical, and high in resolution. Indeed, we claim that in order for computational tools to be effective and accepted, they must support these properties. Within this thesis, we argue that it is now possible, using the language of a task, to augment real world artifacts, creating digital tools as robust, malleable, and portable as paper and other physical artifacts.

In support of this thesis, we developed Rasa, a tangible augmented reality environment that digitally enhances the existing paper-based command-and control-capability in a military command post. By observing and understanding the users' speech, pen, and touch-based multimodal language, Rasa computationally augments the physical objects on a command post map, linking these items to their digital representations—for example, linking a paper map to the world and Post-it™ notes to military units.

Herein, we (1) argue that the properties of physical tools cannot be ignored when designing computational replacements, and demonstrate the effect of doing so in a military command post; (2) identify constraints for the design of fail-safe, mission-critical

systems for decision support; (3) present Rasa, a system for augmenting physical objects, transforming them into tangible interfaces; (4) review the related work in augmented reality and paper-based and invisible interfaces; (5) present the findings of an experimental evaluation of Rasa in the field; (6) discuss the results of this evaluation, the limitations of Rasa, its relevance, and the overall impact of Rasa on the design of intelligent systems; (7) expound upon potential avenues of future work, including a vision for Rasa that includes the computational augmentation of arbitrary physical objects

Foreword

Portions of this work have appeared elsewhere. Herein, each has been significantly revised and extended from their original forms. Moreover, we retained copyright of these materials for inclusion in theses publications such as this one. Thus, all material here falls under unrestricted or fair use terms of U.S. Copyright Law. Our research on confirmations in multimodal dialogue, portions appearing in Section 4.7.1, was first published at the International Joint Conference on Computational Linguistics (COLING/ACL 1998), © Université de Montréal, (McGee, Cohen, & Oviatt, 1998). The findings in our field work leading to Rasa's initial design and constraints, portions appearing in Chapter 2 and 3, was first reported at the Conference on Designing Augmented Reality Systems in April 2000, © ACM, Inc. (McGee, Cohen, & Wu, 2000). We initially described Rasa's architecture at the Intelligent User Interfaces Conference in January of 2001, © ACM, Inc. (McGee & Cohen, 2001), and portions of that work appear here in Chapter 4. Our arguments on how human language use extends the context of physical objects appeared in an essay in a special issue on "Context-aware Computing" in the Human-Computer Interaction Journal in December of 2001, © Lawrence Erlbaum Associates, Inc. (McGee. Pavel, & Cohen, 2001), and portions appear here in Chapter 3. The addition of computer vision to Rasa's sense-making of military command-post map tools was presented at Perceptual User Interfaces, November 2001, © ACM, Inc. (McGee, Pavel, Adami, Wang, & Cohen, 2001). Finally, an empirical study of Rasa's use by members of the Oregon Army National Guard was first described at the Conference on Human Factors in Computing Systems (CHI 2002) April of 2002, © ACM, Inc. (McGee, Cohen, Wesson, & Horman, 2002). Portions of that work are supplied in Chapter 5 and 6.

"As a minimum, it seems to me, we should insist on all major computer installations being designed to 'fail softly' by falling back to a degraded state of operation rather than collapsing catastrophically. In the case of chemical plants, nuclear power stations, or medical intensive care units, we should insist that the control function is so designed that it can if necessary be taken over by a human operator in the event of a computer breakdown." (Hawkes, 1971)

"After a vote against management, Vivendi Universal announced earlier this year that its electronic shareholder-voting system, which it had adopted to tabulate votes efficiently and securely, had been broken into by hackers. Because the new system eliminated the old paper ballots, recounting the votes—or even independently verifying that the attack had occurred—was impossible." (Mann, 2002, pg. 82)

Chapter 1 Introduction

Systems designed to support computation in environments like military command posts, hospital intensive care units, or traffic control centers often fail because they do not take into account the dominant role of physical tools. These tools are adequate for their purposes because they possess at least two critical properties of tools in environments where professionals are constantly making life-and-death decisions. They encourage collaboration and are extremely robust. This thesis examines the following question: can we lower the barrier to acceptance of technology in these highly conservative workplaces if we design computational aids that retain the properties of physical artifacts, thereby adopting and augmenting the non-computerized tools and procedures of a task? As the quotes at the top of the page seek to demonstrate, such conservatism is not just needed for certain tasks, but required.

Through implementing and evaluating a new system called Rasa, we demonstrate that systems can be designed that enhance the users' experience with their chosen physical tools. By combining three methods—*augmented reality*, *tangible interfaces*, and *multimodal interaction*—we can design systems that retain traditional physical qualities while delivering the benefits associated with advanced digital computation. The goals of this thesis are 1) to describe how to design systems for highly conservative work places, 2) to provide a series of constraints on those designs based on our ethnographic findings, 3) to deliver some fundamental software components that can be used to develop such systems in the future, and 4) to support all of these with an empirical comparison of one set of tools to its augmented tool set exemplified by Rasa.

In the remainder of this chapter, we will define the terms relevant to this thesis, present the background research in these areas, discuss the motivating challenges, and summarize the argument put forth.

1.1 Background

In general, this thesis expands on Weiser and Wellner's (Weiser, 1993; Wellner, 1993) visions of the potential for ubiquitous and pervasive computing technologies by merging the foundations of three relatively new human-interface methods:

- *augmented reality*—superimposing digital information onto or alongside physical objects in the real world, thereby augmenting human users' perception of those objects or the users' environment (Azuma, 1997; Mackay, Velay, Carter, Ma, & Pagani, 1993),
- *multimodal interaction*—relying on one or more natural human input modalities such as spoken and written language, touch, or hand gestures (Bolt, 1980; Cohen et al., 1999; Maybury, 1993; Oviatt et al., 2001), and
- *tangible interfaces*—adopting physical objects as a means of manipulating digital information (Fitzmaurice, Ishii, & Buxton, 1995; Ishii & Ullmer, 1997; Rekimoto & Saitoh, 1999; Ullmer, 2001).

These techniques provide the means to enhance *existing* tools, such that they can profit from information technology without radical modification of them or their use. Discussion of such tools is in Chapter 2. This section introduces each of these computer interaction methodologies to the reader. More information on each technique is provided in the ensuing chapters of the thesis.

1.2 Thesis statement and research claims

This thesis contends that

> It is possible to augment or extend real world artifacts, such as paper maps and Post-it notes, creating digital tools that are as robust, malleable, and portable as these physical ones.

More precisely, we claim that it is possible to design computationally augmented artifacts:

1. That are resistant to power and digital communication failures.
2. That are as easy to use or easier to use than the natural, physical tools.
3. That are as malleable as physical artifacts.
4. That do not significantly increase the cost of capturing the information represented by the physical objects in a digital format.
5. That are as transportable as existing tools.
6. That are as high in resolution as the current artifacts.

In support of these claims, we will be presenting (1) a system design (Chapter 4) that attempts to meet these challenges and (2) evidence of the design's effectiveness from an empirical study which compares the use our implementation of the design, in Rasa, to existing physical tools (Chapter 5). Such claims represent a broad approach to designing fundamentally new computational tools based on successful physical tools. Moreover, such designs represent a method for capturing the benefits of both high-technology and low-technology approaches to solving problems.

1.3 Contributions

This thesis contributes to the state-of-the-art in tangible, multimodal, and intelligent interface design. Based on our observations in military tactical operations centers and related ethnographies in similarly high-risk, stressful, and safety-critical environments, we present a set of design constraints and a methodology that supports the development of complex physical and computational hybrid tools. This thesis delivers Rasa, the first tangible computing system to augment physical tools via multimodal language.

The manuscript contains a description of the system, architecture, and a set of fusion rules and constraints in support of a specific domain application: the common operational picture tool in a military command post setting. Finally, we provide the first evaluation of a tangible interface in support of an existing work process.

1.4 Thesis overview

Chapter 2 of this thesis describes especially conservative work environments and characterizes the tools used in them. Based on this assessment of these environments and their tools, we develop a set of design criterion for the introduction of computing tools into these environments in Chapter 3. In Chapter 4 we provide a description of Rasa, the tool we designed that embodies these constraints for one common tool within military command centers. We performed an empirical evaluation of Rasa, which we present in Chapter 5. We discuss the results of the experiment in Chapter 6, along with the identification of Rasa's limitations and requisite future work to overcome them. In Chapter 7, we present our vision and conclusions.

"A glimpse at the ways in which [computers] are used within their respective settings reveals how the various tools and artifacts that were hitherto believed to be a reflection of an outmoded and bygone age—paper notes, jottings, sketches, scribbles and the like—constitute critical resources in a broad range of organizational domains. Why such unsophisticated tools and artifacts remain an integral feature of workplace activities, despite the deployment of new technologies, cannot be treated as a curious example of 'cultural lag', but rather as an embodiment of a complex, and largely unexplained, form of human practice which features in the accomplishment of mundane actions and activities." *(Heath & Luff, 2000)*

Chapter 2 Decision environments: tasks, tools, and teams

Computing systems that attempt to automate safety-critical decision environments, such as those found in hospitals, traffic control centers, and military command posts, often do not account for the way that physical artifacts and human language constitute a medium for collaborative activity. Consequently, these approaches fail to win the favor of naturally conservative end-users. In this chapter, we examine ethnographies of these three environments to understand their common characteristics and the artifacts used in them. Based on these comparisons, we argue that when automating these types of environments, designers must begin to deliver tools that support co-present collaboration and are extremely reliable. Otherwise, our solutions will continue to fall into the chasm that separates early- and late-adopters of technology (Moore, 1991).

2.1 Overview

Despite the breakthroughs that computing has offered in the past half-century, people still spend most of their time working and playing in the real world: meeting each other, sketching out diagrams together, writing and editing as a team, collaborating on designs, etc., all without the aid of computers. Due to the tangible nature of everyday physical tools, people frequently discard computational substitutes. Put simply, people often find physical tools more appropriate than computational ones.

For example, flight controllers across the globe still insist upon using pens and paper flight strips to negotiate each aircraft's route through their airspace (Mackay, 1999; Mackay, Fayard, Frobert, & Médini, 1998; Mackay et al., 1993). They use shared symbols to annotate the strips. providing not only a record of change, but also a tool for collaboration; they lay out the flight strips on boards as an indicator of relative height and distance. Similarly, military officers use a paper map, pens, and Post-it™ notes to track the position and disposition of units in the field and to plan future conduct (McGee et al., 2000). Officers draw symbols on Post-it notes that represent people and equipment, and then array the Post-its on a shared map board that is visible to all who enter the command center. Health care professionals create bundles of highly specific information on paper, consisting of preformatted charts, but also whatever happens to be at hand (Gorman et al., 2000; Heath & Luff, 2000), such as the back of a gauze pad (Figure 2.1).

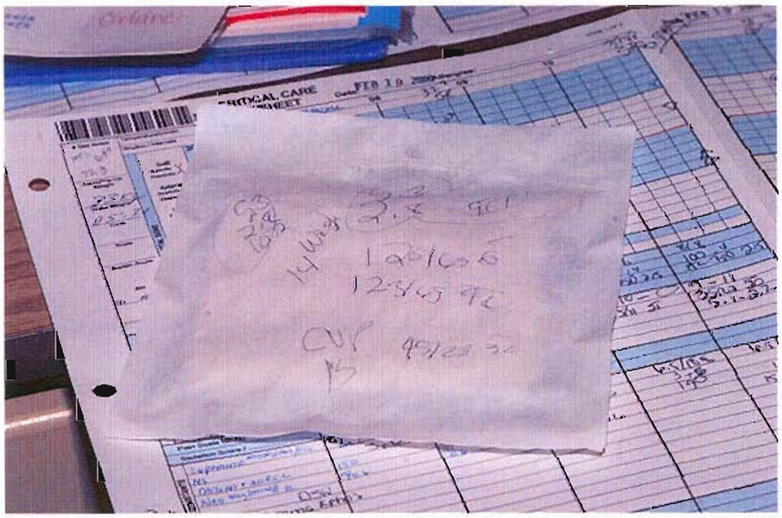


Figure 2.1 An annotation written on the back of a gauze pad envelope, from (Gorman et al., 2000).

These three domains represent a class of work environments in which computing has largely failed. Professionals in these domains remain technologically conservative, because choosing new tools that are not an improvement over existing ones can result in tragic consequences. Our view is that new approaches to human computer interaction that *blend* the distinctive qualities of these environments and their existing tools with computation will bridge the chasm between these necessarily pragmatic work processes and the technological innovation they deserve.

In the next two sections, we will further characterize the three challenging environments introduced earlier, in terms of the tasks people perform in them, the people who perform these tasks, and the tools they use to execute these tasks. We will examine the properties of decision environments where the cost of adopting safety-critical systems that simply do not assist or, worse yet, have the potential to fail, can be tragic. We will describe how designs to automate these tasks fail to account for a considerable part of their purpose; namely, as a means of robust problem-solving within teams—oftentimes collaborating without expressly interacting (i.e., by observing others, overhearing, etc.). This failure is what ultimately led us to our set of constraints on the design of computing systems for these environments (McGee et al., 2000). These constraints were the basis of our design for Rasa (McGee & Cohen, 2001), a decision-making tool for command and control that takes a new approach. Rather than replacing the physical tools that preceded it, Rasa adds computing to them. This approach, which encourages designers to evolve tools when faced with overly conservative users, is one way to bridge the aforementioned chasm that separates late-adopters of technology from potentially beneficial applications.

2.2 Late-adopters

The following three sections describe the three environments mentioned above: health care, air traffic control, and military command and control. For air traffic control and health care, we summarize findings from the ethnography of our colleagues working with experts in those domains. We present our own ethnography of military command and control in Section 2.2.3. Section 2.2.4, examines these similarities in detail.

2.2.1 Critical health care

This summary of activities and artifacts in health care is drawn primarily from the work of Gorman et al. (Gorman et al., 2000) in intensive care units.

The intensive care units (ICUs) of hospitals exhibit a "high level of patient complexity and acuity; a considerable and unpredictable flow of patients in and out of the unit; complex medical equipment of every description to support a variety of patient care and other tasks; a constant stream of diverse hospital personnel and visitors from outside the unit; a remarkably high level of ambient noise; and a professional team approach that is highly focused on patient care in a setting of constant change, interruption and uncertainty." (ibid) Moreover, ICUs are "characterized by high uncertainty, low predictability, frequent interruptions, and potentially grave outcomes; where time and attention are highly constrained; and where interdisciplinary team-work is essential." (ibid)

Due in large part to these conditions, expert clinicians rely on *bundles*, "organized, highly selective collections of information, to help solve problems and maintain situation awareness." (ibid) According to other medical researchers (Woods, Cook, & Billings, 1995) maintaining situation awareness is critical for related tasks such as anesthesiology as well.

People construct these bundles from digital and paper-based information sources. Paper sources ensure that some parts of the bundle are discardable, foldable, durable, and portable. Consequently, they can survive frequent revision and heavy use. Entries on these sources are often made in pencil, allowing clinicians to erase and add new information, so that a current record is always at hand and unique, though often redundant with other records. These collections help clinicians organize and prioritize the performance of tasks. They provide a flexible means of recording information, and they are malleable, allowing clinicians to create unique annotations on partially completed bundles. Lab slips taped on a door indicate requests to collect specimens, making bundles contextual as well. Additionally, some of these paper-based bundles are quite ad hoc, literally "back of the envelope" creations (again, see Figure 2.1) where clinicians make quick notes in their own shorthand. These ad hoc bundles tend to be tightly integrated with specific tasks, and therefore useful when electronic documentation is distant.

2.2.2 Traffic control

The following assessment of air-traffic control is drawn from the ethnography and design concepts of Mackay and her colleagues (Mackay, 1999; Mackay et al., 1998) and researchers at the CSCW Research Centre at Lancaster University (Bentley et al., 1992; Harper, Hughes, & Shapiro, 1991; Hughes, Randall, & Shapiro, 1992). Heath and Luff (Heath & Luff, 2000) have identified similar characteristics for traffic control in underground train systems.

"Air-traffic control is a complex, collaborative activity, with well-established and successful work practices. The work is highly situated, requiring rapid responses to constantly changing conditions. The work is also risky: a controller holds the fates of thousands of people in the course of an hour. Mistakes that result in crashes are simply not acceptable." (Mackay et al., 1998)

Controllers use paper flight strips, which contain an ongoing record of the scheduled flight route through each individual flight sector, to organize the traffic, plan strategies, and record and monitor key decisions. The controllers personalize their view of the traffic, often collaboratively, by physically laying out the strips in two dimensions on a strip board to represent the flights in both the temporal and spatial dimensions. Oftentimes, controllers place the strips of aircraft that are in conflict next to one another, or 'cock out' strips (set them at an angle) to draw attention to them. This physical layout of objects helps to organize the controllers' perception of the real world, and augment the controllers' observation of events related to the flight strips in his or her periphery.

The paper flight strips act as a physical representation of the otherwise-invisible aircraft. "Controllers often take strips in their hands as a concrete reminder to deal with that strip next." (ibid. pg 560) The controllers equate strips with aircraft. Physically handling the strips gives controllers a sense of ownership and responsibility for the aircraft and the lives at stake, especially during cooperative work, where physically handling a strip in the presence of others connotes ownership of the task. Furthermore, the strips help controllers communicate with one another without interruption of the task. Controllers annotate strips using a shared symbology that their co-workers can easily understand. Strips let controllers easily accommodate on-going changes in the task. The controllers write or draw annotations, corrections, and amendments directly on the strip, including circling important flights, unusual destinations, arrows pointing at routes, or symbols designating crossing patterns. "Moreover, the actual performance of some manual activities, such as writing on the strips, manipulating them in the racks,... serve to keep the controller, and other members of the team, 'geared into' the work." (Bentley et al., 1992, pp. 127)

This behavior allows controllers to create a "rich mental image of the traffic... reducing the controller's mental load, allowing him or her to retain only the important details, since the rest of the information is always instantly accessible in front of them. The physical strips can be viewed as a concrete component of the controller's mental representation, helping him or her handle more information and successfully deal with interruptions." (Mackay, 1999, pp. 322-323)

Perhaps more importantly, strips are robust, reliable, and do not break down. Indeed, at various air traffic control centers, strips, battery-powered radios, and cell phones may be the essential tools for safely landing aircraft when "modern" computing systems fail. These failures have occurred after earthquakes, when buffer overflows cause systems to crash, and when uninterruptible power supply tests accidentally cause the main power to fail, for example. "Since controllers are responsible for people's lives, they want a system that works even when all other systems fail. (Mackay et al., 1998)"

The advantages of adding computing tools to these tasks are numerous: an increase in the amount information sharing among controllers via active databases, automated conflict detection and resource allocation, etc. Governments have been attempting to replace paper flight strips with wholly automated air traffic control systems since 1987. Such paperless air traffic control systems have met with critical failure, such as the UK's National Air Traffic System (NATS), developed by IBM, which after ten years of development has finally became fully-operational January 2002. Despite these failures, the number of related automation efforts has recently increased due to the overwhelming increase in air traffic itself.

2.2.3 Command and control

We base our assessment of work practices and artifacts in the domain of military command and control on field studies we conducted at the Marine Corps bases at Twentynine Palms and Camp Pendleton, California, and with the Oregon Army National Guard in Portland, Oregon. During these studies, we observed commanders and their subordinates engaging in field training exercises. We videotaped these observations, and from each videotape, we transcribed spoken and gestural interactions that occurred within the command post. The photograph in Figure 2.2 was taken during an especially frenetic period in a division command post we observed during an exercise at Ft. Leavenworth, Kansas.

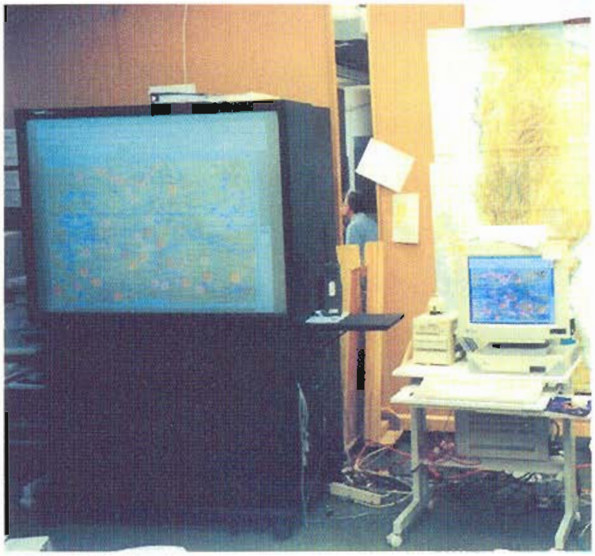


Figure 2.2 Military command and control systems in action. Photo courtesy of Bill Scherlis.

On the left is a rear-projected SMART Board™ and PC and on the right is a Sun workstation. Nineteen other systems are in the immediate foreground. On each is the latest command and control (C^2) systems, high-dollar investments created to aid the commander and his staff by providing situational awareness. However, no one is using these computers (or any of the other 19). Rather, the commander and his staff have quite purposefully turned their backs on computer-based tools and graphical user interfaces, preferring instead an 8-foot by 6-foot paper map, arrayed with 3M Post-it notes (Figure 2.3).

Figure 2.3 What commanders prefer. Photo courtesy of Bill Scherlis.

The numerous responsibilities of officers include tracking the movement of friend, foe, and neutral parties, planning for future combat, evaluating both enemy and friendly situations, etc. To engage in these activities, they construct a kit of useful items from everyday objects, which always includes a high-fidelity paper map of the terrain, pens, and objects that commanders use to represent the various forces. These kits are a shared resource amongst a team of collaborators, for the officers do not work alone. They are constantly conferring, arguing, and assessing analyses of the situation and courses of action. Indeed, most of this activity takes place right at the paper map.

The large map depicted in Figure 2.3 has the same two-fold purpose as the C2 systems in Figure 2.2 above: (1) to depict the terrain, its occupants (military units consisting of soldiers and machinery), their position, and capabilities; and (2) to overlay that information with a graphical rendition of the daily plan for the force. Commanders or their map plotters sketch a symbol representing a unit's functional composition and size in ink on each Post-it (examples are depicted in Figure 2.4 and Figure 2.5). As units' positions arrive over the radio, the plotters move the Post-its representing these units to their respective positions on the map.

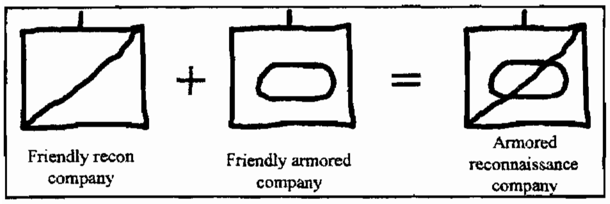


Figure 2.4 Composition of unit symbols

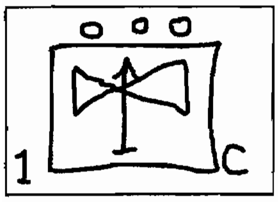


Figure 2.5 Friendly attack helicopter, First platoon of Charlie Company.

The users establish associations between the Post-it notes and their real-world counterparts with a standardized, compositional language, used since Napoleon's time, which is capable of denoting thousands of types of units. The unit symbol is one among thousands derivable from a composable *language* for these pictograms that officers learn during their training and use daily. The location of the Post-it on the map represents the unit's position in the real world.

The military symbology language taught to all soldiers consists of shapes to indicate friend (rectangle) or foe (diamond), a set of lines, dots, or "X's" to indicate unit size (squad to army), and a large variety of symbols to indicate unit function (e.g., mechanized, air defense, etc.) denoted by combinations of meaningful diagrammatics. For example, an armored reconnaissance unit's symbol is a combination of the marks used for armor and for reconnaissance, as shown in Figure 2.4. In addition, the unit's reporting structure (e.g., First platoon, Charlie company) is often indicated by an abbreviation (e.g., 1 and C written to each side of the symbol as in Figure 2.5). By virtue of this language's compositional nature, the symbol language can denote thousands of units as well as their position in the unit hierarchy. Because the language is compositional, soldiers are able to understand and generate complex concepts in terms of their parts. Moreover, soldiers use the language to communicate the situation to others by arraying such symbols on written notes or other devices (e.g. pushpins) on the map. Somewhere between several dozen and several hundreds of these Post-its may be arrayed on a typical command post map.

Officers keep the map up-to-date by constant communication both up and down the units' organizational hierarchy. In response to radio reports, it is the job of users in each command post to keep all of this information as accurate and as complete as possible, so that their superiors can make critical decisions efficiently and quickly. The currently deployed computer systems were intended to reduce the analog communication flow by providing situational awareness digitally at all levels. However, because the computational interface lacks certain properties that are essential for decision-making in this environment, the officers often continue to choose paper tools in favor of their computational alternatives. They do so because paper has extremely high resolution, is malleable, cheap, lightweight, and can be rolled up and taken anywhere. Indeed, command posts will often roll up their maps with attached Post-its, move to a new location, unroll, and within minutes continue operations as before. Additionally, officers handle the Post-it notes as they collaborate. As they debate the reliability of sensor reports and human observation to determine the actual position of units in the field, they jab at locations on the map where conflicts may arise. They also pick up Post-it notes and hold them in their hand while they debate a course of action.

In addition to the information represented by the Post-it Notes on the map, auxiliary information is available on nearby charts. At any time, anyone who looks at the map should have a clear picture of the current state of affairs.

However, despite major efforts to digitize command and control for ground forces, command posts are still very much a paper, acetate, and grease pencil affair. Command posts must be absolutely robust to all kinds of failure (e.g., hardware, software, communications, and power). Because they are subjected to oppressive environmental and operational conditions, these types of failures are common. During our recent observations, communications were intermittent, power generators failed, and the desert conditions proved fatal to hardened desktop computers. Humans are another strained resource in this environment—the workers themselves are heavily task loaded. Overall, these conditions lead to a lack of tolerance for any human interface that is confusing, unforgiving, or difficult to operate.

The effort spent on this work practice has doubled as command posts have begun digitizing this task. Not only does the officer, or a computer specialist assigned to her, update the unit's position on her graphical user interface (GUI), she continues to update an acetate map overlay hidden behind the projector's screen using the Post-it techniques described above. The officers synchronize the paper and digital copies of the current situation in order to mitigate the risk of losing command capability should the computing system fail.

2.2.4 Task properties

Air traffic control, critical health care, and command and control share a number of similar characteristics. (1) Users are building situational awareness: primarily formulating plans and creating mental representations for real-world processes, activities, and objects using uncertain information. (2) These professionals cannot achieve these objectives by working alone; instead, they rely upon the expert judgments of their colleagues, reducing uncertainty in their own judgments by collaborating with and observing each other. (3) Collaboration is at the heart of these tasks and is attained primarily using shared language. (4) Such activities are distinct from traditional "collaboration" in that these users rely not only on direct communication but also on situational cues. These tasks are embedded in an environment in which the "end-user" is never just an individual at work, but in which "secondary" interactions, such as overhearing and directly or peripherally observing this work, are just as important, sometimes even more so. (5) Despite severe constraints in time, attention, and expertise, experts must make decisions quickly to protect human life, often leading to satisficing (making a satisfactory, but perhaps not optimal, decision). (6) Each of these environments must operate almost continuously, requiring several shifts. Consequently, particular individuals may or may not be on hand when a crisis arises. The rapid transfer of prioritized and situation-specific knowledge is essential when shift changes occur or when new plans are developed. (7) Safe operation is a paramount concern in these tasks. In each, human experts make decisions that reduce uncertainty, and safeguard and protect lives. These are human-centered, collaborative, decision tasks that are both mission- and safety-critical. Because of the potential loss of life due to mistakes or failures, experts have adopted robust procedures and tools that diminish these risks as much as possible.

2.3 Artifact properties

The physical artifacts adopted for mission and safety-critical decision support tasks (e.g., flight strips, situational maps, and physician's bundles) share common properties also. Gorman et al. (Gorman et al., 2000) in their ethnography of clinical records use identify five important features of "bundles": tangibility, informality, redundancy, annotation, and active creation. We concur with their classification, and revise and extend it here.

2.3.1 Tangibility

When technologists consider supplanting physical tools with computational replacements, we often ignore the existing tools' tangibility. Computing tools often do not adequately mimic these aspects, such as their portability, malleability, and ability to act as placeholders for real-world objects. We have described several instances in Section 2.2 above, yet they are worth summarizing here.

Note-taking tools that are not sufficiently malleable (supportive of rapid change) can pull a physician's attention away from his or her patients, colleagues, and tasks. This malleability is also essential in air traffic control centers. Command and control tools that replace maps are not sufficiently malleable, or portable. Moreover, they do not supply information at a resolution that is adequate for a number of tasks. By restricting how officers can interact with one another, they also do not effectively support side-by-side collaboration near the map.

People can easily make physical artifacts take on different meanings in different situations. For example, people use Post-its to stand in for the objects dominating these workplaces, relationships among them, or tasks, and concerns. In addition, passing these physical objects from one person to another is an efficient way of delegating responsibility for these concerns. Users employ this method effectively in all of these settings.

2.3.2 Informality

Informal objects are flexible in the variety and types of information that they carry with them. Moreover, the structure of this information is not prescribed by the tool, but made flexible by language, symbols, diagrams, and similar ad hoc methods of human communication. Because these tools do not prescribe the structure of information, users need not struggle to understand the structure as it is formally imposed by traditional information systems, and the metaphorical "interaction" techniques such tools must adopt. Consequently, the cognitive overhead of physical tools that support informality are greatly reduced in comparison with typical computing tools. Finally, physical tools, such as paper, support information processing that is temporary by themselves being disposable. As such, paper's informality is put to very effective use. Individuals and groups combine information provisionally, avoiding the cognitive overhead of tools that require formal specification of categories and relationships.

2.3.3 Persistence

Real-world artifacts are by their nature persistent and robust to the types of failures we commonly associate with machines. This aspect of persistence is critical for these environments, considering the safety-critical nature of the tools. People cannot wait for tools in these environments to reboot, let alone recover from failure. Physical artifacts have their own set of robustness issues. Paper, for example, can be misplaced, burned, lost, smudged, torn, etc. Fortunately, these failures do not generally overlap with those of computing systems.

2.3.4 Annotation

Physical artifacts support rapid annotation. They can be marked or written on in well-understood ways, and thereby updated physically and for the purposes of sharing the state of the task or the representative object. These annotations are from a shared language that can be specific to each domain, perhaps even specific to each team and environment.

2.3.5 Active creation and interaction

Physical artifacts are publicly available, occupying space in the world. Because of this, they can be laid out in the real world according to task-specific needs. In doing so they serve as cues and memory aids, arranged by people according to criteria such as criticality, scheduling, or for domain-specific reasons (e.g., flight strips arranged by location in air space, or in a time sequence depending on needs). Furthermore, as aids for a team's situational awareness, the *creation, references to,* and *manipulation of* these objects must be an activity that others remain aware of even when not directly engaged in it themselves (i.e., in the periphery of the physical environment). When this is possible, the interaction with each artifact and the artifacts themselves provide situational awareness. Moreover, interacting in this manner ensures that the persons who must ultimately make the decisions, can remain vigilant even when the activity has diminished (Parasuraman & Mouloua, 1996).

2.4 Influence of task properties on artifact properties

We examine why physical artifacts, as opposed to computational tools, are specifically chosen in these environments. First, the kinds of co-located, situational, collaborative activities that are the norm in these environments are difficult, if not impossible, to support with current computing systems. By their nature, these tasks often involve people interacting dynamically with shared tools, whether pointing at flight strips that represent flight paths in jeopardy or grabbing a Post-it note and arguing about "its" potential damaging effects on friendly unit positions. See Figure 2.6 for one example. Today's computers tend to reduce or eliminate the amount of human-human communication surrounding them. However, this is just the kind of behavior that collaborators, such as the officers in command posts, rely on to assess a situation. Second, the ability to communicate efficiently and effectively, using shared languages, aids in the rapid evolution of problem solving. Forms of this communication, such as written symbology or shorthand, allow us to quickly perform activities such as naming and referring. However, this ability is missing from most computational interfaces. Moreover, humans interact *multimodally* (with speech, gesture, and written symbols) to enhance the bandwidth of their communication. Consequently, people choose artifacts that increase multimodal communication channels rather than detract from them. Third, because of the reliance on collaboration to arrive at rapid decisions, and the number of personnel required to execute these complex tasks, users frequently interrupt one another, leading to a reliance on tools that serve as persistent memory aids, such as written notes, and robust placeholders for task state. Finally, the tools that they choose are common and fail-safe. They must be quickly replaceable, interchangeable with what can naturally be found at hand, and reliable. Flight strips are made with a pen, a piece of paper, and a pair of scissors; maps are often simply sketched when a full-relief map is unavailable, inappropriate, or unnecessary. Therefore, these artifacts remain in use even as computing attempts to supplant them. They are reliable, support collaboration, and use symbols and language that serve to aid memory and cognition, whereas computing systems halt work upon failure, fail to make work visible to collaborators, and restrict the flow of human spoken, gestural, and symbolic communication.

Figure 2.6 Officers engaged in several discussions of an evolving situation at a map.

2.5 Constraints on the design of systems to support these tasks

From these insights, we identified five key constraints for designing systems that retain the benefits of the artifacts identified. These constraints are given in Table 2.5. We use the term augmentation to describe how users annotate physical artifacts in order to extend an object's meaning: for example, commanders augment Post-it notes so that they represent particular units in the field, and controllers augment flight strips so that they represent a different flight plan than the one scheduled.

Table 2.5. Design constraints

Minimality Constraint	Changes to the work practice must be minimal. The system should work with user's current tools, language, and conventions.
Human Performance Constraint	Multiple end-users must be able to augment the artifacts themselves using the shared language that is native to the environment.
Malleability Constraint	Because users gain information about the real world object over time, the meaning of an augmentation should be changeable; at a minimum, it should be incrementally so.
Human Understanding Constraint	The users must be able to perceive and understand their own augmentations unaided by technology. Moreover, multiple users should be able to do likewise, even if they are neither spatially nor temporally co-present. Users must also understand what the augmentation entails about the corresponding objects in the real world.
Robustness Constraint	The work must be able to continue without interruption should the system fail.

Together these constraints ensure that any design to automate environments accounts for the properties that the existing tools provide. The minimality constraint addresses the need for computational artifacts to be based on the existing common tools, procedures, and language. Consequently, users can continue to create and interact actively with the current tangible artifacts, regardless of failures or inconsistencies in the associated computing systems. The human performance constraint ensures that the common shared language becomes the means by which users interact with the computational component of the system. The malleability constraint guarantees that users can extend the meaning of artifacts over time. The human understanding constraint requires that all users understand how each artifact is augmented, i.e., what it represents in the real world.

Two derived constraints immediately follow from these: First, a corollary of the minimality, human performance, and human understanding constraints is that in order to function in the given environment, human-machine interfaces, especially those interfaces necessary to augment an object or change the meaning of an object, must be based on the current work style. The minimality constraint reminds us that the system should adopt the tools common to each environment; therefore, any computational aid should attempt to perturb the task as little as possible, and thus should be rendered as invisibly as possible. Naturally, since language is such an integral component of the tasks outlined above, proposed interfaces should leverage this tool of human behavior. One solution to this dilemma is to add sufficient sensing mechanisms to the environment so that users can rely, as much as feasible, on any existing multimodal language to complete their tasks

A second consequence, based on the minimality and human understanding constraints, is that the system *must* rely on the language of the work practice to establish the proper representational relationships between the augmented objects and the digital world. Those denotational relationships should be analogous to those being created between the physical artifacts and the real world entities that they represent. It is the job of the system's semantic interpreter to ensure that these relationships are consistent.

The corollaries ensure two things: (1) that both human understanding and the ability to extend objects are possible even without computational aid, and (2) that the system uses an interface that is essentially invisible, relying on the natural spoken and written language of the task rather than just contemporary approaches like graphical user interfaces. In essence, these constraints serve to ensure that the properties we identified earlier are retained.

2.6 Technology chasms

Physical tools are so familiar that mimicking them, e.g., building metaphors of physical tools and environments has been and still is *the* major theme in human interface design. In his classic "The Design of Everyday Things" (Norman, 1988), Norman, examines the distance between (1) a person's intended actions and the actions actually required to affect the state of the world, and (2) actual changes brought about in the world and the readiness with which these changes are perceived. These distances Norman calls the gulfs of execution and evaluation respectively. People perceive the functions of common objects from their physical attributes and social standards relating to their use, leading to what Norman calls a *natural mapping*. An object's ability to express these functional mappings is commonly referred to as the object's *affordances* (Gibson, 1979). People simply perceive affordances for everyday objects. Chairs afford sitting and trees afford shade, for example.

We claim that contemporary computational interfaces fail to take advantage of the affordances and natural mappings of physical objects; consequently, they create a technological chasm (Moore, 1991), which prevents more conservative workers, like those mentioned earlier, from using computational tools. According to Moore, this chasm represents the span of acceptance of technology products from early adopters to the much larger group of so-called "pragmatists and conservatives." The pragmatists want evolution rather than revolution, the opposite of the early adopters. They want technology that enhances and integrates into existing work practices and systems, whereas early adopters are willing to expend energy on learning, understanding, and advocating technology that is only potentially fruitful. Moore's conservative users want *continuous* rather than *discontinuous innovations*, which force the customer to change his or her behavior, infrastructure (e.g., current hardware and software), processes, etc. For whatever reason, pragmatists and conservatives are unwilling to make these adjustments.

We contend that supporting natural mappings and the affordances of *physical objects* can eliminate these gulfs as they are typically introduced by desktop computing interfaces. Moreover, we claim that this represents a way to bridge the chasm between early and late-adopters.

2.7 Summary

In general, users in the work environments discussed in this paper have resisted attempts to computerize their tasks. We suggest a number of reasons for this. The spoken and written language of these and many other tasks makes the users' collaboration possible—teamwork that is critical to their success and safety. Yet, designs often fail to account for the effects of the technology on the existing human-human collaboration that naturally derives from the use of *language and shared artifacts* in a situated context. Moreover, physical tools are more efficient, reliable, convenient, and cheap—and they get the job done.

However, computing systems, most especially personal computers, do not exhibit these properties of physical artifacts. As a result, users in our target environments have chosen one of two options: (1) retain the "redundant" physical artifact-based work practice and use it in tandem with the computing system, thus doubling the users' effort or worse, or (2) dispose of the computing automation system altogether and continue to rely on the physical artifacts. We argue that users should have more options than these: options that blend useful physical tools with computation.

We present one embodiment of this proposition in the form of Rasa (Chapter 4), a system that augments a commander's map and paper tools to capture and understand the spoken and written language used naturally in the commander's collaborative task.

In this chapter, we argued that a set of constraints can be derived from the needs and requirements expressed by users of certain safety- and mission-critical environments. Such environments rely heavily on physical tools to provide support for teamwork, situated communication, and robust operation. We presented summaries of ethnographies for three such environments that have consistently rejected replacing these paper-based processes with traditional computing solutions. We argue that systems supporting these constraints can help their users cross the chasm of technology adoption, even as they rightly retain their necessarily conservative stance. Indeed, such systems could help propel these user populations beyond the current generation of computer interface technology toward next generation more natural, multimodal interfaces.

"A major role of new technology should be to make tasks simpler. A task can be restructured through technology, or technology might provide aids to reduce the mental load. Technological aids can show the alternative courses of action; help evaluate implications; and portray outcomes in a more complete, more easily interpretable manner. These aids can make the mappings more visible or, better, make the mappings more natural. Four major technological approaches can be followed:

- Keep the task much the same, but provide mental aids.
- Use technology to make visible what would otherwise be invisible, thus improving feedback and the ability to keep control.
- Automate, but keep the task much the same.
- Change the nature of the task."

(Norman, 1988, pp. 191-192)

Chapter 3 Augmenting environments with language

3.1 Introduction

In describing the human-machine interface and its design, Krueger argued that: "The computer should adapt to the human, rather than the human adapting to the computer." (Krueger, 1991) Likewise, we seek to augment the tools in a work practice, such that in the beginning we significantly alter neither the tools nor their use. In this chapter provide a general description of how we propose to "augment" physical tools.

3.2 Approach

In order to communicate efficiently about objects in the real world, people ascribe meaning to objects at hand, using suppositions and assumptions: "We're the Xs and the opposing team is the Os." Or, "Let's assume that this rock is the building across the street." Similarly, people develop physical models of objects in the real world: some simple (e.g. a sketch), some intricate (e.g., maps). Researchers have focused on developing tools that augment reality using these physical models as *tangible interfaces* to computing systems.

In our approach, we also strive to develop techniques that can be used to augment natural environments with computation. However, rather than design and engineer tools that are statistically assigned their meaning in advance of their use, the approach discussed in detail below relies on the language of naming and referring to assign meaning to physical objects dynamically.

3.2.1 Augmenting environments

Augmented reality typically refers to extending our human senses, usually that of sight, to merge virtual overlays of information with reality (Feiner, Macintyre, & Seligmann, 1993; Mackay et al., 1993). Figure 3.1 provides a conventional example.

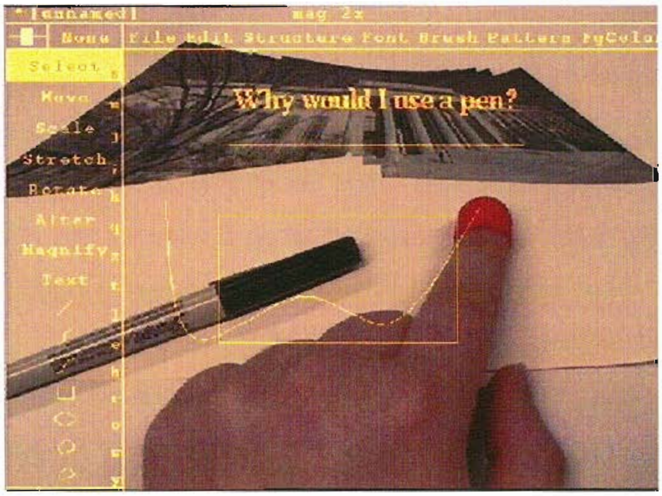


Figure 3.1 Photo of AR prototype. Courtesy of MIT Wearables project. Used by permission.

Due to the augmented reality, the user's visual display has been extended and is now a drawing surface. Similarly, his finger has been extended to become a pen on that surface. The researcher who designed the system predetermined both of these augmentations. Indeed, in most augmented reality systems, the designers choose what aspects of reality are augmented, i.e., which artifacts are extended and what new meanings they acquire.

Figure 3.1 poses a reasonable question. "Why would I use a pen," when I can use my finger to draw in my new augmented reality? There are several reasons why I may choose a pen over my finger. A pen is a more natural and precise writing instrument than a finger. It leaves behind a trail, a permanent, persistent, public, physical record. Like many physical tools, the pen has experienced millennia of evolution. Rather than replace it with a computational convenience (or contrivance), why not construct a way for these highly evolved physical artifacts to evolve into computational ones, just as the finger above has been made into a writing instrument?

The larger question here, however, is not whether a pen is a more appropriate tool, but whether the user should have the ability to choose augmentation strategies. Should my finger be a pen forever, constantly leaving digital ink in its wake? In the next section, we will argue that people choose augmentation strategies all of the time, and that denying them this ability in computationally augmented realities limits these systems' usefulness and flexibility.

3.2.2 Natural augmentation

People already augment reality. With language, we have the ability to transform arbitrary physical objects into something entirely different: a combination of the original object and an associated one. This aspect of language—the ability to name and refer, thereby creating linguistic placeholders for later use— can be made the basis for the design of interfaces that are aware of broader aspects of the context of use.

For example, suppose we want to give directions to the other side of town, and we know that a prominent water tower and its physical relation to streets, buildings, etc. will aid us. To do this, we often use objects at hand and create an association, using spoken or written language, to combine the two. For instance, we can point at the coffee cup and say, "LET'S SUPPOSE THAT THIS IS THE WATER TOWER ON THE OTHER SIDE OF TOWN." Likewise, we can use a pencil to represent Main Street. Alternatively, we can draw a symbol of a water tower on a piece of paper, next to a pair of lines labeled "Main Street." In each case, spoken, written, diagrammatic, and/or iconic language alters the meaning of the physical objects or symbols such that they become stand-ins for other objects in a different context (i.e., the cup for the tower). Language, specifically the abilities to name and to create analogical relationships, can be used as a *bridge* between contexts: in this case, between the context of the objects in front of me and that of the real world. These features of language are extremely useful, especially when the referents are themselves too large, too small, too far away, too cumbersome, or otherwise inaccessible. Perceptual systems that can interpret these shifts of interpretation may have advantages, such as the ability to take appropriate action whenever these denotations occur.

Therefore, we define *augmenting* with the specific notion that people already use tools to augment their environments. Augmenting can be defined as follows:

> *Extending or adding to a real world object to cause it to represent, denote, or be associated with something else or something more.*

This definition subsumes both *natural augmentation*, as we have described it above, and artificial augmentation as it applies to traditional augmented reality systems. However, this definition is substantially different than previous definitions for augmented reality in acknowledging the influence people take in augmenting objects in the real world with declarations and suppositions and how differently thereafter we perceive these objects.

For example, I am throwing a graduation party this weekend, and I'd like you to come. I tell you that it is at 7 pm this Friday, and then I pick up a pencil and a blank piece of paper. With the pencil, I begin to sketch out a map and directions to the party. What has happened to the piece of paper? It is still a piece of paper, certainly. However, it is not only a piece of paper; it is also a map and set of directions to the party. Most people would also consider it as a physical reminder to attend. A system that was augmenting my reality could react to my natural augmentation of the paper. For instance, as you prepare to attend my party, the augmented reality system could (1) display a countdown on the invitation, informing you if you are running behind schedule or remind you to pick up a congratulatory bottle of champagne; (2) assess the traffic patterns for the evening, and display the most efficient route to the party directly on the map or onto the ground in the real world; etc. Therefore, by observing the way that people extend the meaning of things, computing systems can pickup on the denotations and provide numerous benefits, such as those listed here. This type of natural augmentation leads to several satisfying properties for certain artifacts, which we have described above and which coincide with the constraints we identified in Section 2.5.

3.3 Augmenting decision environments with language

Users are already naturally augmenting paper—for example, creating denotation relationships between Post-its and the things they represent by drawing glyphs in a symbolic language on each note (see Figure 2.4 and Figure 2.5). If a *system* could perceive these augmentations by understanding the users' shared language, then they could continue to employ familiar tools, which would then be coupled to the digital world. By "language," we mean an arrangement of perceptible "tokens" that have both structure and meaning to their users. This definition is meant to subsume both natural spoken and written languages, as well as diagrammatic languages such as military symbology.

To accomplish this, systems should understand the users' language. From this multimodal language (e.g., drawing a unit symbol on a Post-it note), the system should create a denotation relationship from a real-world object (e.g., a military unit) to a particular virtual object in the digital world (e.g., virtual or simulated units), which itself denotes a real world entity. Precisely how the digital entity comes to have the intended meaning to the user is a complex issue that is beyond the scope of this dissertation. Because the system supports augmentation through understanding the user's language, the user need not even know that objects in his work practice have been further analyzed by a system, or even that a computer system is operating behind the scenes. Figure 3.2 depicts this model as it could be supported in the command and control domain. System components are adopted that can observe, recognize, and understand the natural denotation and physical manipulation of the Post-its and maps. This understanding supports the automatic population of an active database of virtual entities that correspond to the actual units, terrain, and controlling measures used. How we implemented this model is described in Chapter 4.

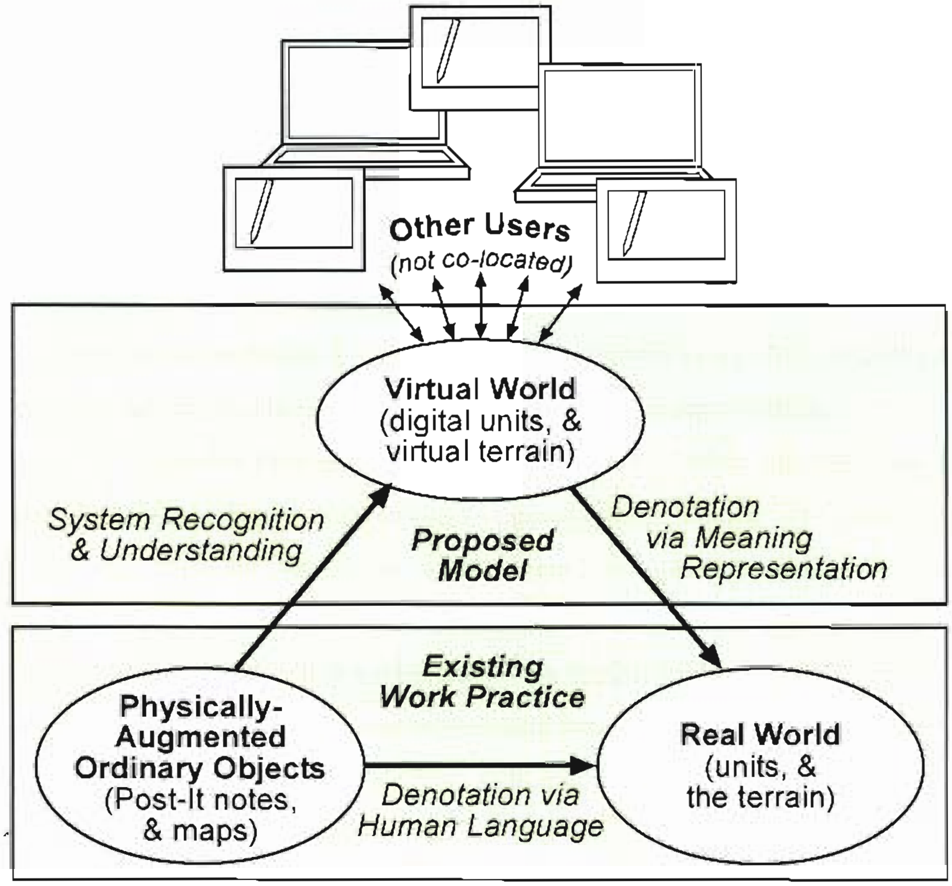


Figure 3.2 Using augmentations in the real world to produce meaning in the virtual world.

3.4 Related work in augmenting environments

This work was inspired by visions of ubiquitous computing and augmented reality (Krueger, 1991; Newman & Wellner, 1992; Weiser, 1993; Wellner, 1993), though our work most closely resembles the approaches of Ishii and his students (Ishii & Ullmer, 1997; Ullmer & Ishii, 1997, 1998; Underkoffler & Ishii, 1999), Moran and colleague's Collaborage (Moran, Saund, van Melle, Gujar et al., 1999), Passage (Streitz, Geibler, & Holmer, 1998), and the design prototypes by Mackay and colleagues (Mackay et al., 1998). Brief descriptions of these are provided in the following sections on augmenting tangible artifacts and paper.

3.4.1 Augmenting tangible artifacts

The Urp (Underkoffler & Ishii, 1999) system augments a natural, non-digital work setting. With Urp, planners use building models, rulers, clocks, and other physical objects to design an urban environment. Objects are tagged by patterns of colored dots, and if a pattern is recognized, the vision system sends Urp the associated object's location. Urp uses tools that are natural and familiar in their setting. In Urp, all interactions are physical in nature. With Urp, augmented objects "behave" as you would expect them to: rulers measure distances, clocks mark time, and so on. The object's physical characteristics and the environment it inhabits govern these expectations.

Within the Passage concept (Streitz et al., 1998), meaning can be linked to a physical object whenever that object is placed on a "bridge." In the initial prototype, the bridge is a scale and recognizes objects based on a precise measurement of their weight. The concept itself allows for arbitrary bridges. Users place physical objects on the bridge and select the electronic material they wish to link with the object, turning it into a *passenger*. When this occurs, the association is written into a central database. When the passenger arrives at another bridge, it is recognized, and the linked information is displayed on a computer display.

Mackay et al. (Mackay et al., 1998), as we detailed in Section 2.2.2, have analyzed why paper flight strips in air-traffic control centers have so far not been replaced by computerized artifacts, suggesting design alternatives that rely on augmenting the paper rather than replacing it. They describe a prototype for a flight strip holder that allows the controllers to digitally annotate each flight strip, organize the strips on strip boards, and continue to physically manipulate the strips in their new holders. These digitized strips, though suitably capturing the handwritten annotations for remote collaboration and storage and certainly robust to failure, do not capture the meaning of the annotations for digital processing, database update, and the like.

3.4.2 Augmenting paper

At least four other projects that have augmented paper are relevant to this research: Wellner's DigitalDesk, Ishii's transBOARD, Moran's Collaborage, though none can be classified as strictly applying to a particular work practice, as does Mackay's flight strip designs. After our initial design and implementation, Klemmer and his colleagues at Berkeley developed the Designers Outpost, a system that has a great deal in common with our approach.

The DigitalDesk (Wellner, 1993) augments office work by introducing paper into a workstation environment. Through computer vision, users can point at numbers on a real piece of paper. In response, the system performs optical character recognition and pastes the recognized number into a graphical user interface application: e.g., the system's calculator. Similarly, regions of real paper, like a sketch on a napkin, can be cut and pasted into a painting program. Consequently, users can *transfer* information from paper documents into a graphical user interface that supports pointing and touching in order to perform computational actions, like mathematical calculations.

The transBOARD (Ishii & Ullmer, 1997), a shared whiteboard, uses barcode-tagged cards to hold digital ink. The barcode, kept on a 3x5 card for example, can then be carried back to a person's office. The ink can be retrieved when scanned by a barcode reader connected to a desktop computer, where further interaction with the ink (e.g. printing, emailing, etc.) is possible.

The Collaborage concept (Moran, Saund, van Melle, Bryll et al., 1999), which characterizes augmented systems consisting of a board and various tagged physical information items, has been applied to build several prototypes at Xerox/PARC. One representative prototype is an In/Out board system. With the In/Out board, glyph-tagged magnetized photos can be slid from the *Out* column to the *In* column and vice-versa. Within seconds, a vision system recognizes the change in location of the glyph and an In/Out web page is updated to reflect the change in status. If the system were to fail, individuals could still check the physical In/Out board, move their picture from one column to the other, add hand-written annotations, and walk away with up-to-date information.

Berkeley's Designer's Outpost (Klemmer, Newman, Farrell, Bilezikjian, & Landay, 2001) is most similar to ours in approach. The Outpost also augments an existing paper-based work practice, i.e., that of a web-site design studio. It allows designers to rapidly and naturally construct an affinity diagram: a group decision-making tool, designed to sort a large number of ideas, process variables, concepts, and opinions into naturally related groups. When web designers use the diagram, they write ideas for web pages on the Post-its, they rearrange the nodes on the diagram by moving the Post-its themselves, and they connect the nodes on the diagrams by drawing arcs between the Post-its. Groups of Post-its are collected together and labeled. The Outpost captures all of this activity naturally. The diagram is built on a rear-projected SMART Board. A camera mounted within the SMART Board captures the precise location of each Post-it. The SMART Board itself captures interaction with each Post-it (in order to augment the functionality of the paper), and connects the arcs among them. Annotations are saved, however, the augmentations (the writing on the Post-it) is not analyzed.

3.5 Discussion of augmentation methods

In this section, we discuss how these related systems fare with respect to the five design constraints of Table 2.5: namely minimality, human performance, malleability, human understanding, and robustness. Table 3.5 previews each system's satisfaction of the design constraints, including those derivable from the primary constraints. An extended discussion of their limitations follows.

Table 3.5. Comparison of augmentation systems along the design constraints.

	Minimality	Human performance / Language	Malleability	Human understanding	Robustness	Invisible multimodal interface	System understands language
Urp	Perhaps, no studies	No	No	No, though natural objects	Yes	No, yet haptics[1] accessible	No
Mackay et al.	Probably, though no studies	No (annotations, but no associations)	Yes	Yes	Yes	Annotation	No
DigitalDesk	No work practice	No (annotations, but no associations)	Yes	Yes, limited	No	No	Yes, limited
TransBOARD	No work practice	No	No	No	No	No	No
Passage	No work practice	No	No	No	No	No	No
Collaborage	No work practice	No, though notes can be added	No	No, though notes can be read	Yes	Annotation	No
Designer's Outpost	Yes. Limited studies.	No (annotations, but no associations)	Yes	Yes	Yes	Annotation, vision, and touch	No (only the linkages are "understood")

Other than Urp, Mackay's flight strip designs, and the Designer's Outpost, these systems were not intended to support a pre-existing work practice like those we described

[1] Haptics support tactile and kinesthetic interaction.

in Chapter 2, and to date none of them have rigorously examined the benefits or costs of augmenting those environments. In fact, no study has yet been conducted that evaluates user satisfaction, job performance, error rate reduction, reliability or any of the other critical aspects of usability, when a work artifact has been augmented as we describe here.

Though three of the aforementioned systems support annotation, none interprets the human act of augmenting paper as the operation that creates a digital embodiment or association, which is our *human performance constraint*. Rather, to create an association, almost all of these systems require that the linkages between physical and digital artifact be encoded during the engineering phase of the tool (i.e., a part of the software architecture).

Mackay's flight strips, the DigitalDesk, and the Designer's Outpost meet our *human understanding constraint.* We attribute this success to their limited augmenting capabilities. Recall that these systems allow users to annotate directly on the physical artifact, but the annotations take on no new meaning within the digital world. With the other systems that support more complex augmentations, users can only learn what information is associated with an object if the user and the object are adjacent to a "detector." By contrast, since the physical characteristics of Urp's and the Collaborage's tools can be understood at all times, the state of their digital associations, which indirectly correlate with the physical tools, can be implied (e.g. the clock in Urp is used to adjust the simulated time). If these correlations are unclear, untrue, or the user wishes them otherwise, then a design modification or re-programming of the artifacts is required. Indeed, associational augmentation methods like these and others, such as the use of colored dots, glyphs, or bar codes, fail to present the linked digital information to the user without the assistance of technology. Thus, these methods in general would not satisfy our *human understanding constraint*.

It is because users in the environments in Chapter 2 are augmenting objects with written language, rather than simply associating physical objects with digital information, that these augmentations remain both visible and understood. Furthermore, with written language, additional content can be added to an augmented object, thereby recording a history of changes to the augmentation that remains permanent and visible. This particular aspect of *malleability*, incrementality with permanence, does not hold for all modalities of language. In particular, speech does not have this property. However, the Post-its in the command post as well as the flight strips in the air-traffic control centers are currently augmented with speech when the information being added tends to be transitory. These invisible adjustments to the meanings are shared with other users when necessary, or when questions regarding the objects arise. Furthermore, a system that observes and interprets these spoken changes has the option of making these changes visible, by maintaining a comprehensive representation of the augmented object. However, for many of these systems, objects are augmented using glyphs or tags rather than a natural language; consequently, users cannot easily add new digital information to them. For example, a new employee cannot use any magnet and photograph and expect it to work in the Collaborage, nor does a Passage user know what linked information he is carrying around on his key chain, unless of course he has scribbled it down somewhere—but that would be using language.

Using language does more than simply augment, it transforms: Post-its come to represent units, whereas before they had no prior meaning in the work setting. This enables these settings to have as many augmented objects as pieces of paper, whereas Urp, the Intelligent Room, Passage, and other systems that rely on only the physical properties of objects to denote specific meaning will likely have a smaller number of augmented objects whose meanings are fixed in advance by the developer. As such, the user cannot easily change them (cf., the *malleability constraint*).

Finally, according to the *robustness constraint*, the augmented environment must allow users to continue to work even in the face of a power or other type of failure. Users in our selected environments must be able to grab a radio, a flashlight, a ballpoint pen, a piece of paper, and continue working. After the failure, a system abiding by all of the aforementioned constraints can recover these changes with language common to the work practice. Since the other augmented environments discussed here rely heavily on computers and computer interfaces in order for the users to *understand* the augmentations and to *use* augmented objects, if any of those systems were to fail, the work would stop.

3.5.1 General approaches to augmenting physical objects

Three general approaches to augmenting physical objects are also relevant to this discussion: passive optical codes, the Passage approach, and radio frequency identifier (RFID) tagging. Each of these general approaches allows the system designer to associate electronic information with physical objects.

Passive optical codes (Billinghurst & Kato, 1999; Rekimoto, 1997, 1998; Rekimoto & Nagao, 1995; Underkoffler & Ishii, 1999) (example from Rekimoto 1995 shown in Figure 3.3)—the category of printed codes which includes bar, block, and glyph codes—are inexpensive to produce and do not require a conventional power source in order to communicate the intended code which links the physical object with some electronic information. Typically these types of codes "cannot be easily modified, appended to, or erased." (Want, Fishkin, Gujar, & Harrison, 1999)

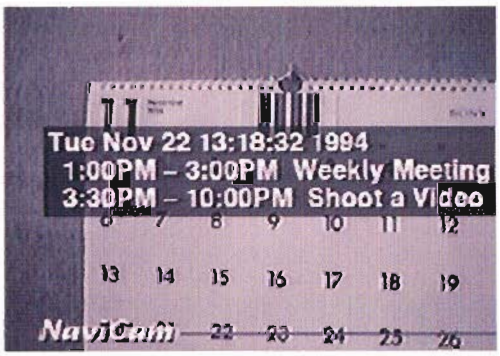


Figure 3.3 Augmenting reality with the NaviCam by recognizing dual-colored bar codes.

Electronic machine-readable tags (e-tags) can be hidden in books, documents, watches, etc (Want et al., 1999). As with Passage, associational augmentations can be formed, whenever the tags are detected. Some tags even support the storage of small amounts of information in them and a power source is no longer a necessary component of e-tags. The most common technological approach within this class of tags is RFID (radio frequency identification) technology. An RFID chip and an antenna comprise each tag. Tags are *interrogated* by the readers to either read or write the information stored therein. RFID tags are becoming increasingly cost-effective to produce and no longer require an internal power source, but are powered by the tag readers.

Hitachi's wireless mu-chip is 0.4 mm^2 and thin enough to be embedded in paper. Motorola uses capacitive coupling as a method for producing their BiStatix RFID tags, which can also be embedded in paper. In capacitive coupling, electric fields rather than magnetic fields are coupled to and from a reader and a tag. These tags are comprised of a silicon "flip-chip" attached to printed carbon ink (proprietary blend electrostatic ink) electrodes. Capacitive inks and standard printers can print the tags. Only the addition of the flip-chip[2] increases the cost of creating these tags beyond that of printing a sheet of paper.

Each of these approaches augments physical objects by adding some machine-readable, yet humanly indecipherable property to the physical object. Moreover, these approaches require a computing system to associate the electronic information with the code. The associated information cannot be retrieved or modified without a computing system present. Therefore, none is particularly robust to computing failures.

However, there is one category of passive 2D, black and white, optical code that can be applied and easily understood by people, can be easily modified and appended to, is robust and can convey its intended meaning irrespective of whether any technology is functioning— this "code" is alphabets and other forms of written languages.

3.6 Augmenting a command post map

One must consider at least two artifacts when augmenting a command post map tool: the Post-it notes and the map itself. Maps range in size from a few feet square to an entire wall, whereas Post-its are only a few square centimeters. These maps, in addition to being covered with the Post-its are often layered with several plastic overlays, each of which characterizes the ongoing real-world situation in a different way. In fact, the Post-its often reside on the overlays rather than the map itself. The augmentation of these tools must capture the meaning of objects placed upon them, their location, and their removal as well.

3.6.1 Augmenting large paper surfaces

To augment the map itself, a system must be able to perceive when users draw symbols directly on the map's overlay, when they place Post-its on the map, when they

---

[2] Flip chips are grafted directly, face-down onto substrates. By contrast, current wire technology uses face-up chips with a wire connection to each pad.

move them, and when they remove them. There are primarily two methods available for capturing this map input: vision-based and touch-sensitive.

Touch boards, such as the SMART Board manufactured by SMART Technologies, Inc. (D. A. Martin, 1995), can report when and where a single point on the surface is depressed. This information, when coupled with an understanding of the task could be used to determine the location of Post-its when they are added, moved, or removed and drawing when it occurs on the system.

However, touch board-based systems do not provide sufficiently direct coupling between the physical objects and their system representations. For example, the system should not necessarily take action whenever the map is touched; neither should the user always be required to touch the map when intending to modify the meaning associated with a Post-it note. For example, placing objects on the map is an intuitive way to track their locations; however, the need to always touch the board, rather than simply pick up the Post-it, can feel unintuitive and unnatural. The same is true for placing the Post-it in its new position on the board, though the odds of touching the board (either by rubbing along the gum line or dabbing a point thereon) does more readily contribute to the system's understanding the user's intentions in this case.

With a touch board, users are restricted and must follow a specific sequence of steps; they must draw a symbol on a Post-it note and then immediately position the note on the map before beginning to work with a new symbol. Any intervening action at the touch board would produce an error in understanding. However, in real-world situations, users often prefer to draw multiple symbols at one time, perhaps many hours before they are placed on the map, in anticipation of units expected on the field that day. If the system could track the unit placement and movement visually, the order of events could be more flexible and controlled solely by the user. Likewise, in order to move a unit, the user must "select" a Post-it note by depressing the board on the Post-it and then place it elsewhere on the map, again by depressing the touch board; she cannot pick up multiple notes and put them down in a different order. Finally, registration of the paper map is difficult to accomplish using the touch system alone, since so little information is being communicated (the location of a single touch point on the display at any one instance in time).

Alternatively, a machine vision system can be constructed that recognizes maps or other drawings used in their place (e.g., photographs, satellite images, etc.), providing an unobtrusive means of registration and calibration of the source image with its digital data. Moreover, these systems can typically *segment* the scene, separating notable objects (such as our Post-its) from the background using size, shape, color, and other attributes (McGee, Pavel, Adami et al., 2001).

Vision-based systems for tracking the changes mentioned above suffer from a different set of issues. First, vision algorithms are notoriously poor at dealing with anything but uniform lighting. Although this condition can be achieved in some environments, it is quite likely to be violated in the conditions expected for the military command post. Second, segmentation algorithms typically cannot separate objects when they overlap with one another, and therefore cannot necessarily determine that they exist at all.

The other objects that complete the toolset for command and control are used for tracking the movement and determining the composition of dynamic forces on the terrain (i.e. units, civilians, etc.). These objects, usually Post-its, possess interesting new augmentation requirements, which we cover in the following section.

3.6.2 Augmenting small, lightweight, positional placeholders

Post-it notes introduce specific challenges in augmentation, due to their most useful attributes: their size, mobility, and disposability. We compared three major techniques for augmenting Post-its: codes and tags, which have been used to augment small physical objects by several researchers (Want & Russell, 2000), and the symbolic language, which we described above 2.2.3). Using feasibility dimensions for Post-its as they are used in command posts, we summarize our findings in Table 3.6, below.

Table 3.6. Comparison of augmentation techniques

	Symbols on Post-Its	Passive optical codes	RFID tags
Cost/disposability	~0	~0	Inexpensive → expensive
Postage stamp size	Yes	No	Soon (MEMS)
Emissions-free	Yes	Yes	No
Customizable	Yes	No	No
Perceptible encoding	Human and computer	Computer	Computer
Reliable	Yes	No	Somewhat

First and foremost, the augmentation scheme must fit within the physical restrictions of the objects they are augmenting. Presumably, passive optical codes may one day be small enough such that they do not dominate the Post-it's visible area (undoubtedly relying upon, as yet unrealized, high-resolution cameras and image processing systems) or ink may be used that is visible to the machine vision system, yet invisible to people. However, to our knowledge such systems do not exist. And, though using Micro-Electromechanical System (MEMS) technology, engineers will eventually manufacture RFID tags so small that we can unobtrusively affix them to Post-it notes, to our knowledge such tags do not yet exist.

One of the primary reasons that Post-it notes are a useful tool is their disposability, which would be diminished by any increase in their costs. The cost of printing bar codes or glyphs is essentially nil (i.e., the cost of printing to a black and white printer and the paper, though the computer and printer require power to produce the glyphs). Similarly, the cost of creating a symbol on a Post-it is again the cost of the paper and ink. In this case, no powered devices are necessarily required to create the symbol.

Emissions, such as radio frequencies, are a valued resource within command and control environments. It is important that augmentation solutions remain as emissions-free as possible. This requirement also eliminates current RFID tagging technologies as a viable option, since they rely on active use of radio frequencies to communicate the information stored within them.

The encoding used in the symbology language written on Post-its can be understood by both humans and machines, though work on a machine vision recognizer for the symbols has only begun (McGee, Pavel, Adami et al., 2001). The benefit that this type of encoding delivers is stability during episodes of system failure: a benefit that the two competing technologies do not share. In these cases, the symbols represent the "stored" information sufficiently enough for commanders to continue to act in the face of failure. Again, human-understanding, robustness, and the other design criteria proposed in Table 2.5 are simply not met by either optical codes or RFID tags.

3.7 Discussion

In Section 3.2, we argue that people naturally use language to augment everyday and task specific artifacts, thereby transforming these items into elements within a story. We showed how these activities were related to the users depicted in Section 2.2.3. Finally, we argued that a system could be built to observe the use of physical tools, like the command post maps and Post-its that was able to meet the design constraints imposed in Section 2.5. Our thesis here is that many approaches can be taken that meet the customers need. However, we argue that by addressing the aforementioned constraints on design, systems engineers can articulate a transition path that helps to minimize the cost to end users while still maximizing their benefit from any radically new technology. Moreover, this approach seeks to guarantee a safe fallback to more primitive and more stable environment m situations that demand mission- or safety-critical operation.

Though several mechanisms for augmenting real-world objects were available prior to the execution of this thesis, none of these were deemed suitable, because none could act upon the language that people already use in the command post domain to create placeholders for other objects in the real-world. Indeed, none of the prior methods truly met any of the constraints that we set forth early in our design, increasing the costs to access the benefit of digital command and control systems.

Hence, for our initial implementation, which is described in detail in the chapter which follows, we chose to capture the use of the objects on the command post map using a touch-sensitive surface. To overcome several of the limitations we outlined above, such as the ability to register the map easily, we combined the touch modality of a SMART Board with natural spoken language. Consequently, when someone touches the map, the underlying system tracks the point in reference to the coordinates of the map. To ensure that the underlying system is aware of the meaning behind the objects placed on the map, especially those associated with the Post-its, we use both spoken language as well as handwriting recognition systems each time a symbol is drawn on a Post-it. This multimodal system, Rasa, is presented next in Chapter 4.

3.8 Summary

In this chapter, we have described an approach to "augmenting" information work by developing a system of sensors and effectors for observing the natural use of language to extend the meaning of everyday objects. Specifically, we reviewed current systems that we considered may already be capable of observing the use of paper maps and Post-it Notes and address the design constraints detailed in Section 2.5. As shown in Table 3.5 and discussed in that same section, none of the state-of-the-art systems reviewed meet the constraints or can be readily revised, in our estimation, to do so. Hence, we describe in Section 3.6 a set of technologies that could be collected (and indeed which we have collected) to augment a command post map.

No matter how good your interface is. No matter how cool your interface is. It would be better if there were less of it.

(Tufte, 2001)

Chapter 4 Rasa

Rasa is a tangible, multimodal augmented reality system that allows officers to update a command post's map using their standard operating procedures and natural physical tools (paper map and Post-it notes). It does so automatically, by taking advantage of the written and spoken language of that work practice. Rasa captures and updates the placement of units and other important elements on the battlefield digitally. With Rasa, users can write the symbol for a particular military unit on a Post-it note and attach it to a map. After recognizing this drawing of the Post-it note, Rasa stores a digital representation of that unit in the system and makes it available on the network. This digital representation can then be interacted with directly, by manipulating the Post-it note, using speech, among other modalities.

The following section describes our prior work in multimodal systems architecture for our QuickSet system, which underlies the Rasa system. Section 4.2 provides a high-level description of Rasa, how it is used, and its primary hardware sensing components. Section 4.3 shows how to setup Rasa. Section 4.4 describes the information flow within the system. Section 4.5 describes the software architecture, its components, their function and execution. In Section 4.6, we provide even more detail about the way Rasa combines input from its different sensors, the declarative rules which govern this fusion, and the constraint satisfaction system that ensures the rules are viable. Section 4.7 describes the kinds of interactions one can achieve with Rasa and the kind of dialogue a user can expect.

4.1 QuickSet

QuickSet is a tool for multimodal interaction with pen and voice on devices ranging from wireless, handheld computers to interactive wall-sized displays (Cohen et al., 1997). With it, users can create entities on a digital map display by simultaneously speaking and sketching on a touch-sensitive liquid-crystal display (LCD). The user can annotate the map on the LCD, creating points, lines, and areas of various types. QuickSet operates in a distributed multi-agent architecture (Section 4.1.2) and because of that on various heterogeneous hardware configurations (Figure 4.1), including tablet, desktop, and wall-sized form factors, and in 3D environments (Cohen et al., 1999). Moreover, QuickSet controls numerous backend applications, including military simulation, disaster management, and medical informatics. It has also been incorporated by the Naval Research Laboratory into a 3D virtual-reality environment (Cohen et al., 1999) and by Columbia University into an outdoor augmented reality environment (Feiner, MacIntyre, Hollerer, & Webster, 1997). We will provide a brief description of QuickSet's operation next, followed by a discussion of its multi-agent software architecture. For Rasa, we modified, extended, and adapted many of the software components we developed for QuickSet. We will describe those in detail in Section 4.5.

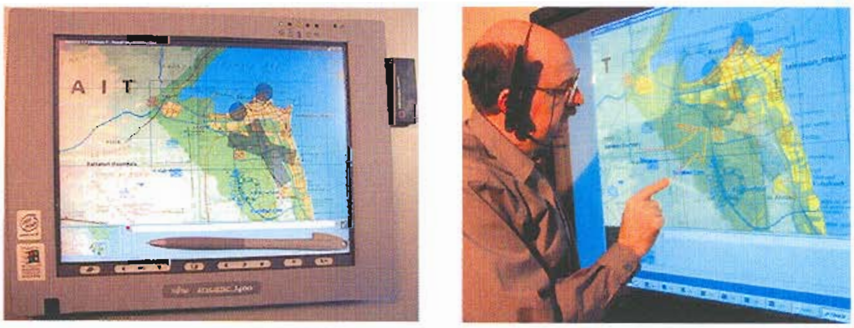


Figure 4.1 QuickSet operating on tablet computer (left) and on large touch-screen display (right).

4.1.1 Operating QuickSet

Entities displayed on QuickSet's digital map are registered to their positions on the terrain. The digital map provides pan and zoom capabilities, overlays, icons, etc. Using pen and voice together, the user can annotate the map, creating points, lines, and areas of various types, for example ditches, fortifications, minefields, and swamps. The user can also create entities (i.e., vehicles or groups of vehicles), give them behavior, and watch a simulation unfold on the displays.

Figure 4.2 depicts the variety of available back-end applications, the primary components of the agent-based software architecture, and its facilitated, inter-agent communication.

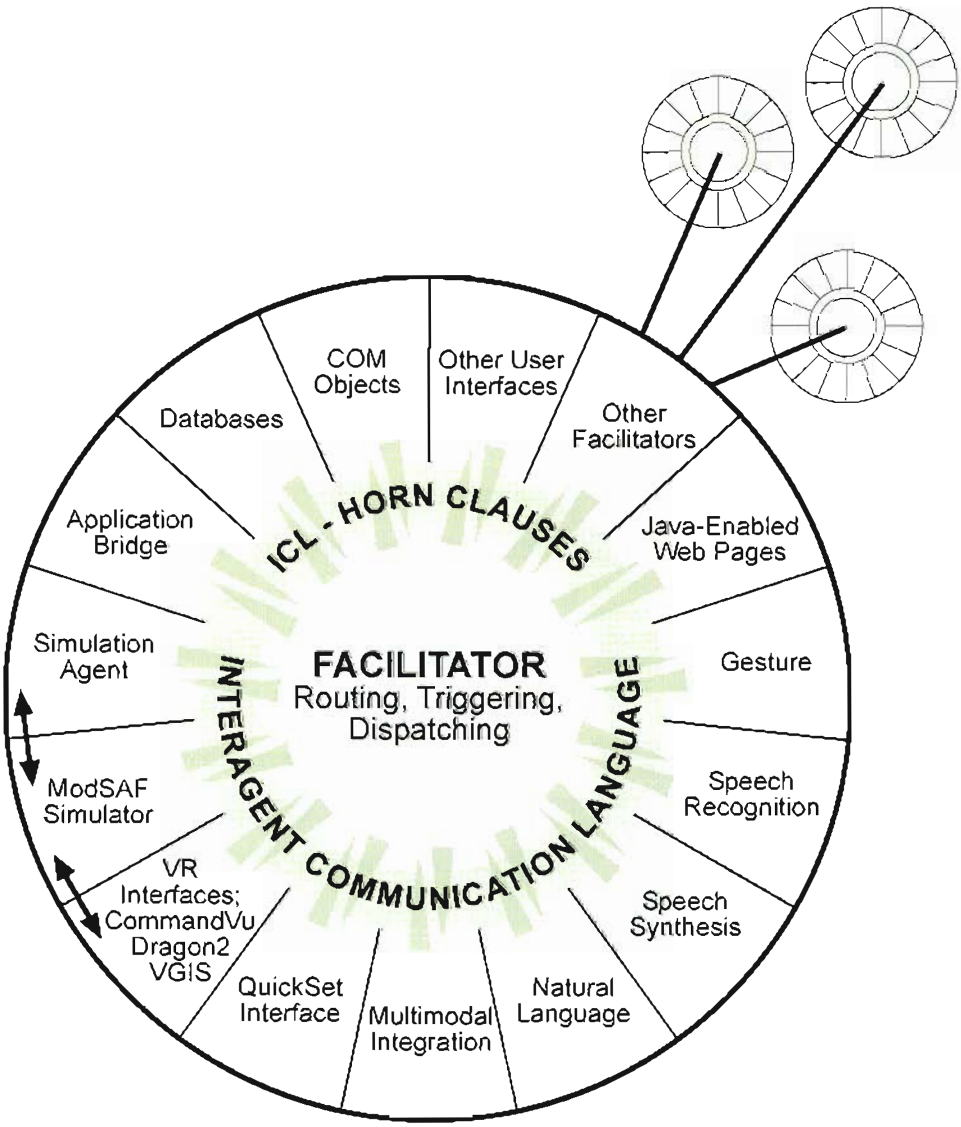


Figure 4.2 QuickSet's multi-agent software architecture, multimodal components, and back-end applications.

QuickSet operates as follows: when a pen is placed on the screen, the speech recognizer is activated, thereby allowing users to speak and gesture simultaneously. The user either points to a spot on the display and speaks the type of an entity to be placed there (e.g., "MECHANIZED COMPANY"), or draws a line or area while speaking its type (e.g., (e.g., "NO GO AREA," "PLATOON BOUNDARY," "BARBED WIRE," and "FORTIFICATION"). In response, QuickSet creates the appropriate icon on its map. To do so, QuickSet uses a combination of speech and gesture recognition components, parsers for these recognizers, a fusion engine, a number of user interfaces, and other types of multimedia output.

Continuous speech and gesture are recognized in parallel, with the speech interpreted by a definite-clause natural language parser. A variety of continuous, speaker-independent recognizers are supported, including Dragon Systems Naturally Speaking, IBM's Voice Type Application Factory, and Microsoft's Whisper. In general, analyses of spoken language and of gesture each produce a list of interpretations, with recognition scores (i.e., independent probabilities generated by each). The multimodal integration process searches among the sets of interpretations of the individual input streams for the best joint interpretation (Cohen et al., 1999; Wu, Oviatt, & Cohen, 1999a, 1999b), which often disambiguates both speech and gesture simultaneously (Oviatt, 1999).

Unification

Unification is a method of pattern matching used in logic programming languages, such as Prolog. It determines the consistency of two representational structures, and if consistent, merges them. Variables here are unlike those in imperative programming languages. Instead, they are simply an unspecified, untyped data object (i.e. term), however complex. During unification, variables are instantiated, or "bound," to patterns in order to attempt to match two (or more) representational structures.

When two features structures are unified, a composite containing all of the feature specifications from each component structure is formed. Any feature common to both feature structures must have a compatible value. If the values of a common feature are atoms, they must be identical. If one is a variable, it becomes bound to the value of the corresponding feature in the other feature structure. If both are variables, they become constrained to always receive the same value. If the values are themselves feature structures, the unification operation is applied recursively. Importantly, feature structure unification results in a DAG structure when more than one value uses the same variable. Whatever value is ultimately unified with that variable will fill the value slot of all the corresponding features, resulting in a DAG.

To demonstrate, let T1 and T2 be two terms.

- If T1 and T2 are both constant terms (i.e., neither contain any variables) then they unify only if they are the same term.
- If T1 and T2 are both variables, then they unify. Each becomes an alias for the other: as soon as one is further instantiated to some value, the other variable will have the same value.
- If T1 is a variable and T2 is any term, then they unify and T1 is instantiated to T2.
- If T1 and T2 are structured terms, then they unify only if:
  - their structures match, and
  - the corresponding arguments within each structure can be unified.

4.1.2 Agent Framework

Rasa consists of autonomous and distributed software components, i.e., agents that communicate using an inter-agent communication language (ICL), based on Horn clauses in Prolog, in the Adaptive Agent Architecture (AAA) (Kumar, Cohen, & Levesque, 2000). The AAA, which is backwards compatible with the Open Agent Architecture (OAA) (Cohen, Cheyer, Wang, & Baeg, 1994; D. L. Martin, Cheyer, & Moran, 1999), is a robust, facilitated multi-agent system architecture specifically adapted for use with multimodal systems. A multi-platform Java agent shell, C, C++, and Prolog libraries provide services that allow each agent to interact with others in the agent architecture.

Agents can dynamically join and leave the system. As they join, agents register their capabilities (the information that interests them and the requests for action that they are committed to performing) with a facilitator, which may be connected to a network of facilitators. Each facilitator provides brokering and matchmaking services for their network of agents. Agent messages are in an agent communication language based on "speech acts" (Austin, 1975; J. Searle, 1989; J. R. Searle, 1969) such as one agents *inform*ing another of some fact or one agents *request*ing another to perform a particular task. These are definite clauses with only one positive literal, i.e., a Horn clause. Kumar et al. provides complete description of the semantics of this communication language (Kumar, Huber, Cohen, & McGee, 2002; Kumar, Huber, McGee, Cohen, & Levesque, 2000).

On its surface, this approach is similar to modern publish and subscribe services. However, the facilitator unifies (*see text inset on unification*) every message it receives with each agent's declared capabilities to determine whether an agent has registered an interest in particular facts or has committed to performing particular actions. If the request unifies with a registration of capability, the facilitator forwards the message to that registered agent. Since, both the capabilities expressed by agents and the messages they produce can contain variables, the facilitator can act as a more intelligent filter on the content, based on the binding of variables during unification (i.e., at run time). Consequently, the resulting framework is more expressive than traditional distributed messaging approaches that provide multi-cast capability, such as Jini (*Jini Network Technology*, 2002).

4.1.3 Summary of QuickSet's impact on Rasa

Rasa is based largely on QuickSet and its language understanding and fusion components. In order to support handheld and mobile devices, we developed an agent-based, distributed component infrastructure to separate QuickSet's interaction and feedback components from one another and from its recognition, understanding, and integration components. This separation was a critical enabler for Rasa's paper-based tool, described below in detail.

Our intuitions about QuickSet's applicability in a new domain, command and control, led us to investigate the types of multimodal interaction that occurred naturally in command posts (or tactical operations centers). We video-taped interactions at maps and other physical tools, such as status boards. We thought that a better understanding of these interactions would lead us to an improved design of interaction mechanisms for handheld and wall-sized LCD displays. Instead, we returned from our studies with a skepticism towards traditional computing designs, learned from our military colleagues. Ultimately, this skepticism led to our design of Rasa. We never anticipated that our ethnography would so radically influence our notions on design.

In this next section, we will introduce Rasa by describing its configuration and use at a high-level. Technical details will follow.

4.2 High-level Description

When someone first sets up Rasa's map, he or she unrolls it and affixes it to a SMART Board, using an adhesive such as 3M Corp.'s spray-on mounting adhesive. Next, the user points at one corner of a Post-it and then the opposite, each time saying, "REGISTER POST-IT," thus, giving Rasa the size of the Post-its being used. Each time, the system responds audibly "POST-ITS REGISTERED." With this information, the system can organize its projection of overlaid information, i.e., determine how large to project its own symbols over the paper Post-its. Once the Post-its themselves are registered, two "buttons" can be added to the map. Since some configuration of the tactical map display will require different layouts, Rasa allows the users to place the buttons anywhere near the map and on the SMART Board. The user places a Post-it on the SMART Board, labeled CONFIRM, OK, etc. and says, "PUT THE CONFIRMATION BUTTON HERE." Rasa responds by saying "CONFIRMATION BUTTON REGISTERED," and by projecting borders around the mounted "button." Similarly, the cancel button is placed on the board. Either button can be put anywhere on the SMART Board, thus supporting arbitrary layouts of direct manipulation widgets in Rasa. This method is similar to that developed by Pederson et al. (Rønby Pedersen, Sokoler, & Nelson, 2000). However, while we rely on the act of writing on the Post-it to provide the association between the paper buttons and their actions, it is the process of printing index cards for each slide in the PaperButton prototype that establishes these links in the slide show presentation.

Users then "register" their map. With Rasa, any type of map (e.g., paper map, satellite photograph, and drawing), under Euclidean square earth assumptions, can be registered to its position in the real world by tapping at two points on it and speaking the coordinates for each. For example, the map in Figure 4.7 below can be registered by touching where gridline nine-eight meets gridline nine-two[3] and saying "REGISTER MAP AT NINE-EIGHT NINE-TWO," then touching where gridline nine-two meets gridline nine-seven and saying "REGISTER MAP AT NINE-TWO NINE-SEVEN." For each point registered Rasa responds "MAP REGISTRATION POINT NINE…" and the rest of the coordinate.

After both points are registered, Rasa says, "THE MAP IS REGISTERED" and begins projecting information on the paper map from its digital data sources, including its own database. Like QuickSet, units and linear features can be added to the map's overlay. However, the way that these annotations are added mimics the way they are normally added to paper maps. Consequently, Rasa's use of sensing mechanisms, covered next, differs greatly from the integrated touch display and microphone typically used in QuickSet's handhelds.

4.2.1 Sensors

Rasa comprises a set of hardware and software components that sense human activity. These sensors feed information into a network of autonomous agents that process the raw input from the different modalities, fuse the input according to a set of rules based in previous empirical research (Oviatt, 1996, 1997; Oviatt, DeAngeli, & Kuhn, 1997), and generate a multimedia response. In this section, we will describe the sensing

[3] This six-digit number is a valid coordinate in a standard military grid format for a small piece of terrain.

mechanisms currently employed by Rasa, provide an overview of its setup and use based on a high-level illustration of the information flowing from the sensors through the system and resulting in system output. This will be followed by more detailed descriptions of the architecture, software components, and algorithms in Section 4.5.

There are three basic sensors used by the initial Rasa prototype to capture the multimodal input used by commanders at their command post maps: touch, speech, and drawing. These sensors are embodied by three types of machines: a touch sensitive board, microphones, and a digital pen tablet.

First, Rasa uses a SMART Board™ (Figure 4.3) to capture touches and the drawing that occurs on its paper maps. Currently, the SMART Board is limited to reporting a single pixel position at any time instant, thus replicating a standard mouse interface. This limitation exists for most related technologies where sensing touch interaction requires physical contact with the board as their stimulus. This single-touch reporting of the sensor is a serious limitation for collaborative interaction, preventing one user from using both hands and preventing multiple users from working simultaneously with objects on the board. However, there are vision-based methods for tracking multiple hands at the surface of a table (May, Thomas, Lewis, & Decker, 1998; Pavlovic, Sharma, & Huang, 1997; Wu et al., 1999a, 1999b), and new devices are beginning to emerge that will allow two-handed touch interaction at their surfaces (Dietz & Leigh, 2001).

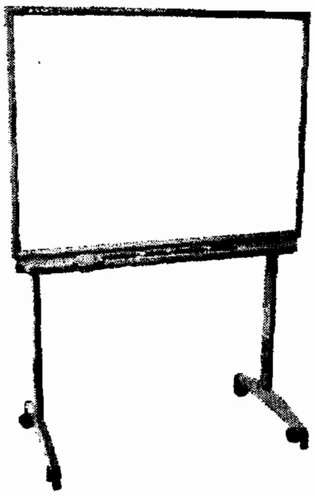


Figure 4.3 SMART Board interactive touch surface.

Spoken language is captured by close-talking microphones (e.g., Figure 4.4) worn by anyone directly interacting with Rasa, or from a microphone array (e.g., Figure 4.5) attached to the top of the SMART Board directly above the map. Completely wireless close-talking microphones, are now available reducing the amount of equipment worn by users. These microphones encapsulate the radio antennae and power source within the headset itself and can be worn comfortably without needing a belt pack.

Figure 4.5 Array microphone.

Figure 4.4 Wireless microphone headset.

Handwriting, on the Post-it notes, is captured by a Cross Computing iPenPro™ pen tablet (Figure 4.6) or similar device. The only requirement is that the device capture digital ink in real-time while at the same time leaving a real ink trail on paper. The iPen-Pro pen communicates wirelessly with the tablet through an RF signal produced by the pen and captured by the tablet. This signal tracks the location of the pen tip whenever it travels above the tablet, generating mouse movement events in the operating system. It also signals the depression of the pen tip on the tablet or on sheaves of paper above the tablet as a standard mouse down event.

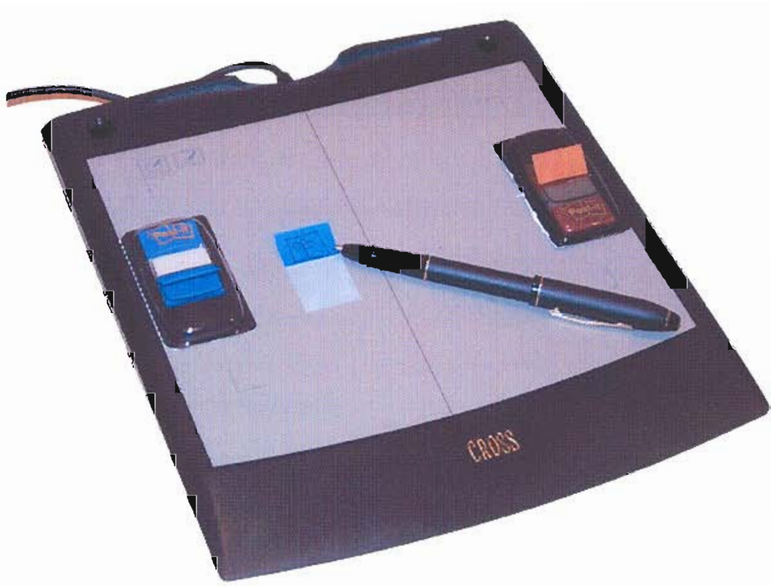


Figure 4.6 iPenPro pen tablet and pen.

These pen tablets and touch boards are essentially digitizing tablets, a technology widely used for the past thirty years for capturing and transcribing static paper map information into geographical information systems (GIS). Similar to Rasa, control points are entered to register arbitrary maps to the GIS. Users select the static linear features to be captured (e.g. edges, boundaries, rail lines, and roads) on a control palette on the digitizer. The capture device is typically a pen or circular mouse-like device with a transparent center and cross hairs for accuracy. Transcription occurs with a series of clicks along the feature or by drawing a continuous line while slowly tracing the feature on the map through the crosshairs. By comparison, Rasa retains the use of the paper maps[4] and employs digitizing sensors to capture highly dynamic information

These three sensors were initially supported by Rasa and were the sensors equipped during our usability evaluation (Chapter 5). However, different techniques could have been used to capture the same multimodal input without any change to the underlying architecture, which we will discuss shortly. Indeed, the fusion rules that enforce the combination of sensor inputs are engineered in a way that allows entirely new sensors to be added with ease, so long as they produce observations of the real-world objects that are similar to the ones currently employed. In fact, we recently added a vision sensor to more effectively capture interaction with Post-its on the map (McGee, Pavel, Adami et al., 2001).

Our vision sensor processes image frames from a black and white camera, captured by a frame grabber. The purpose of the vision module[5] is to recognize objects placed on the map and to estimate their location. Unlike the touch and handwriting sensors, we did not design the vision sensor to mimic the standard mouse interface. Instead, the vision sensor reports the addition, change, or removal of Post-its directly to Rasa. However, vision was not a modality that was used during the evaluation of Rasa (Chapter 5), but was added subsequent to that examination.

4.2.2 Visual feedback

Visual feedback, like that depicted in Figure 4.7 is projected directly onto the paper map, but is only required during setup and reconciliation after a power failure.

---

[4] The digitized maps are used in conjunction with Rasa to support linkages to traditional, laptop or desktop command-and-control systems.

[5] My colleagues Adriana Morelli, Guoping Wang, and Misha Pavel, OGI School of Science and Engineering developed the vision module.

Touching the map instantly leaves a trail of digital ink, which disappears after Rasa processes the command. This trail is left when Post-its are being placed on the map as well as when users draw symbols directly on the map.

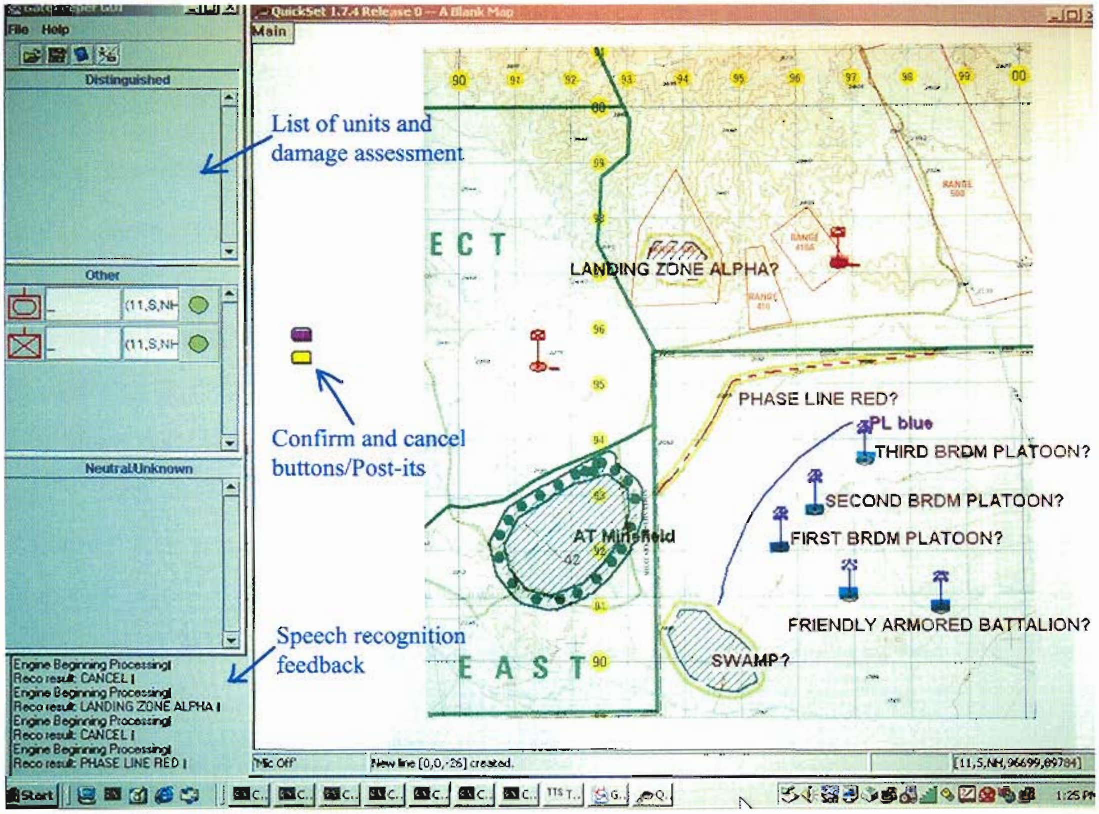


Figure 4.7 Projection of visual feedback. (List of units is on left. Annotations added by author—arrows in blue.)

As new units are placed on the map, a colored shadow indicating their position and their disposition (friendly or enemy) is overlaid on the Post-it. Users can rely upon the shadow as supplemental information regarding the unit's location. The unit symbol itself is optionally overlaid, offset from the shadow, and attached to it by a line. A table of all units present and their combat strength is projected next to the map as visual confirmation of the status of each unit. The table is actually the graphical component of Rasa's distributed database system. This type of table is often found next to the map tool in command posts. For control measures, the resulting military icon for that object is also projected onto the map. Rasa can also project unit symbology, other map annotations, 3D models, answers to questions, tables, etc. relative to the physical information.

Certain projections that are to some degree redundant (e.g., unit symbols) can be disabled by telling Rasa to "DISABLE PROJECTIONS," thereby reducing clutter. Other filtering and query options could be applied, such as "SHOW ONLY FRIENDLY UNITS," and "SHOW ME ALL MOVEMENTS IN THE PAST HOUR," though these and other commands have not been added as of this writing.

4.2.3 Auditory feedback

Soon after the user touches the map or draws on a Post-it, Rasa produces a "scratching" sound. The sound is terminated, when either the pen is lifted or the map is no longer being touched. This sound insures that the users are aware of when the system is listening to their commands (i.e., when the microphone and speech agent are enabled) and when their touch or pen input is being delivered to Rasa. In addition, speech synthesis is used to generate verbal feedback regarding state changes in the system (e.g., "CHARLIE FOUR ONE HAS BEEN SIGHTED AT NINE-THREE-NINE, NINE-TWO-SEVEN"), including those changes that have yet to be confirmed (e.g., "CONFIRM: CHARLIE FOUR ONE IS AT THIRTY PERCENT"). Like the video feedback, either type of auditory feedback can be enabled or disabled by the user.

4.2.4 Summary

Conceptually, Rasa's user interfaces act as transparent interaction surfaces on the paper. Whenever the user touches the map and the touch-sensitive surface beneath it or uses the pen on the iPenPro tablet, she is interacting with Rasa, activating all of its sensing mechanisms. As she does, messages describing the digital ink left behind on those surfaces are sent to the facilitator for distribution to the relevant agents. These physical interactions produce tangible, immediate, haptic and visual feedback. The effect of the interactions, during and after system processing, produce ongoing feedback amplified by the visual and auditory multimedia output Rasa generates.

4.3 Setup

In this section, we will present an overview of how to configure Rasa's computing, sensing, and multimedia output hardware. Figure 4.8 depicts a typical configuration. The touch-sensitive board is attached to a conventional Microsoft Windows™ computing system, through either a serial or USB interface. Attached to a separate computer via a serial interface is the tablet with both digital and real ink pen. The computing systems themselves remain hidden from view and, once installed, neither a keyboard nor a mouse is needed to operate Rasa, instead, the only interaction will be through the digital pen and the touch board. On the pen tablet, we place a stack of Post-it notes or lay individual Post-it flags (semi-transparent Post-its) as needed. The microphones are plugged into any available sound card: one microphone per sound card. The computer configured for audio input should be a conventional hi-powered desktop in order to effectively process the spoken language input. In fact, multiple microphones should be dispersed across computers, one active microphone per CPU. However, if only one microphone is used, the second system (i.e., pen tablet computer) need not be a full-powered CPU.[6]

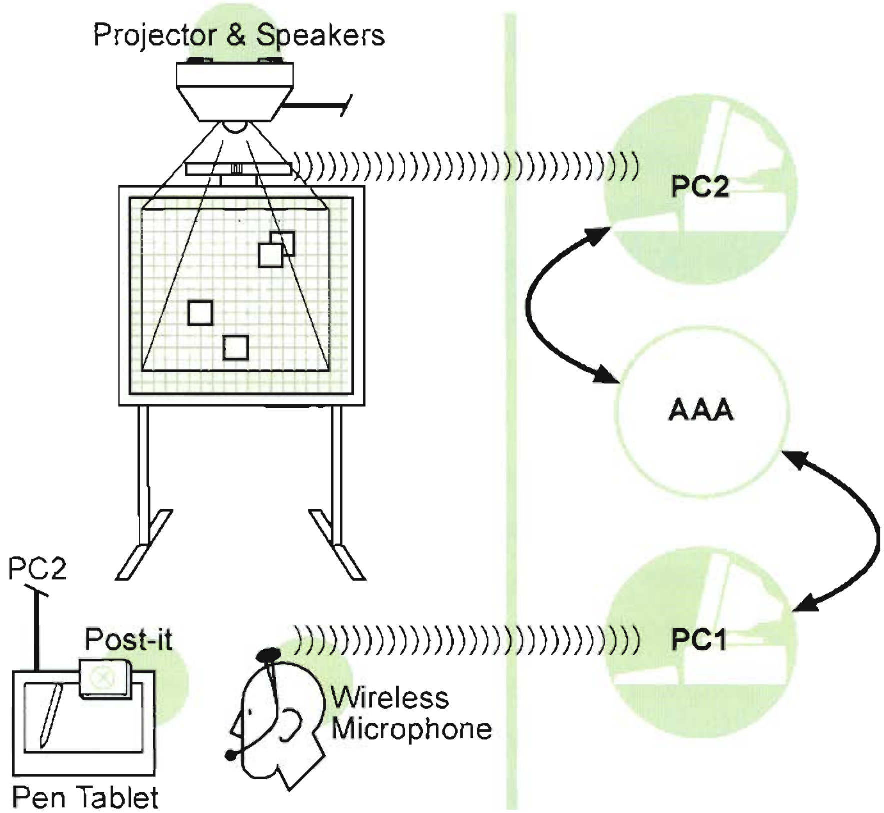


Figure 4.8 Layout of hardware components for Rasa.

Even though the computers and their associated displays, mice, and keyboards are hidden; however, the system generates multimedia responses for its users. This output is

[6] The specific hardware we adopted for our laboratory is provided in Section 5.3.1.

both auditory and visual. The headsets worn convey the auditory responses or, for demonstrations, speakers are used in their stead. A projector is installed on the same CPU as the touch board and the two are co-registered, so that touches on the board are translated into the Windows coordinates of the connected computing system. This registration application is part of the board's own setup.

4.4 Information flow

Before we provide a detailed description of the software architecture and the individual components within Rasa, we will first give (1) an example of interaction with Rasa and (2) the information flow within the system: from the data generated by Rasa's sensors, to the fusion of information within its multimodal integrator, and finally to the presentation of information by its multimedia output processors and devices. From this description, we will identify each of the primary software components within Rasa. A detailed description of each component appears in the following section.

Figure 4.9 provides an example of the information flow in Rasa. As a user receives a radio report identifying an enemy reconnaissance company, (1) he draws a symbol denoting the unit on a Post-it. Simultaneously, he can choose to modify the object with speech. For instance, he draws a reconnaissance company unit symbol on a red Post-it flag and at the same time gives the unit the name "ADVANCED GUARD" via speech. (2) The system performs recognition of both speech and gesture in parallel, producing multiple hypotheses. In less than a second, (3) Rasa has recognized and parsed for potentially ambiguous meanings the reconnaissance company symbol, placing meaning representations in its fusion engine (4) for integration. After verifying the report in his notes, during which the system elapses through an adjustable delay of ten seconds, Rasa responds "WHERE IS THAT RECONNAISSANCE COMPANY?" prompting the officer to put the Post-it on the map. (5) Upon finding the reported coordinate on the map, the officer places the Post-it note there, pushing on the geo-registered map at coordinate 96-94 in the process.

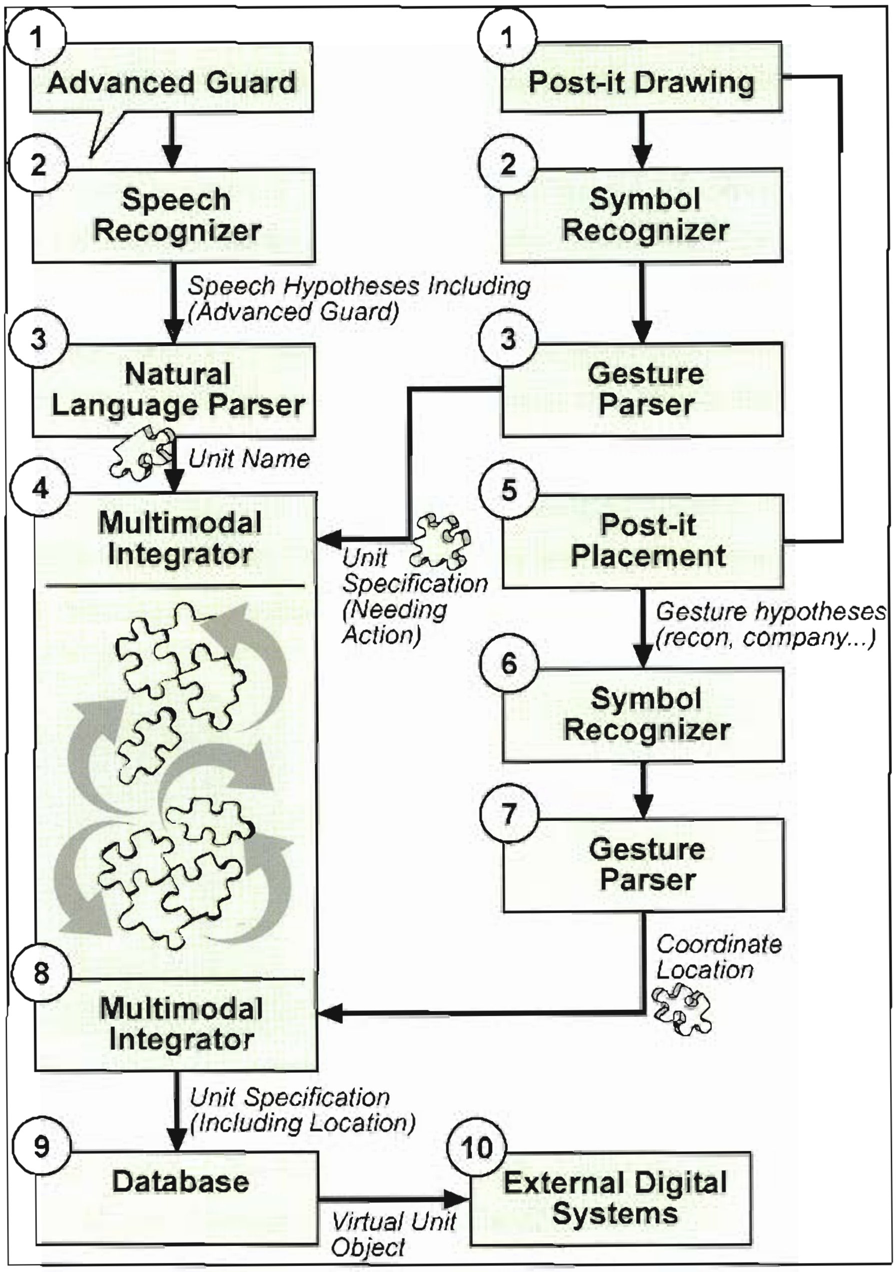


Figure 4.9 Information flow in Rasa

The geo-referenced coordinate provided by this touching action provides the remaining key piece of information needed to validly position a unit, according to a multimodal fusion rule. (6-8) This "gesture" is recognized and parsed, then also submitted for integration. In approximately 1 second, Rasa fuses these inputs and displays a visual confirmation (example shown in Figure 4.7), a mirror image of the recognized symbol and a textual query requesting the user to accept or reject the system's interpretation. A synthesized confirmatory response soon follows: "CONFIRM: ENEMY RECONNAISSANCE COMPANY CALLED 'ADVANCED GUARD' HAS BEEN SIGHTED AT NINE-SIX, NINE-FOUR." Two Post-its are mounted on the map, one for rejecting commands in error and one for confirming correct responses (in Figure 4.7 and magnified in Figure 4.10). To cancel erroneous actions, users can either press the button or press the map and say "CANCEL." Similarly, if the system's interpretation is correct, the user can press the appropriate button or touch the map and say "CONFIRM." However, the user need not confirm at all, since further action by the user, such as placing another unit or moving one, implies that the prior command should be confirmed (McGee et al., 1998). After confirmation, the unit is inserted into a database, which triggers a message to external digital systems. The officer then touches the Post-it saying, "ENEMY." Rasa responds by coloring the projected unit symbol red and saying, "RECONNAISSANCE COMPANY HAS CHANGED ALLEGIANCE FROM UNKNOWN TO ENEMY."

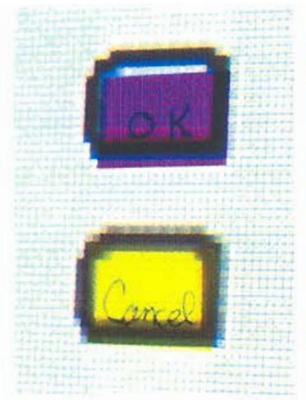


Figure 4.10 Direct manipulation confirmation in Rasa.

The next section describes the system architecture that makes this type of augmented, multimodal interaction possible. Following this description, we will examine this example of Rasa's use in further detail.

4.5 Architecture

Rasa's understanding of language is due to multiple recognition and understanding agents working in parallel and feeding their results via the AAA facilitator to the multimodal integration agent. These agents and the human computer interface agents are described below.

4.5.1 User interfaces

If a user is adding a unit to the map, she first draws on a pad of Post-it notes affixed to a digital pen tablet the symbol representing that unit. A paper interface agent, running on the computer system connected to the tablet, captures digital ink, while the pen itself produces real ink on each note. However, there is no user interface visible, other than the Post-its and the map itself. This user interface is that same as that used on the map. The only difference is the way in which the two paper interfaces are initially registered. The interface to the Post-it requires no additional contextual information to begin using it, since Post-its are treated as simply blank pieces of paper. However, optionally the users can tell Rasa where the enemy (i.e., red) Post-its are located on the tablet versus the friendly (i.e., blue) Post-its (see Figure 4.6).

4.5.2 Recognizers

Interaction with any of these paper surfaces results in ink being processed by Rasa's gesture agent. At the same time, speech recognition is enabled. Each recognizer provides input to the integrator. These agents and their abilities are discussed below.

4.5.2.1 Symbol recognition[7]

Rasa's symbol recognition agent identifies symbolic and editing gestures, such as points, lines, arrows, deletion, and grouping, as well as military symbology, including unit symbols and various control measures (barbed wire, fortification, boundaries, axes of advance, etc.), based on a hierarchical recognition technique called Member-Team-Committee (MTC) (Wu et al., 1999a; Wu, Oviatt, & Cohen, 2002). Examples of the classes of symbols recognized is contained in Appendix A.

The MTC weighs the contributions of individual member recognizers of pattern features based on their empirically derived relative reliabilities, and thereby optimizes pattern recognition robustness. It uses a divide-and-conquer strategy, wherein the members produce local posterior estimates that are reported to one or more "team" leaders. The team leaders apply weighting to the scores, and pass results to the committee, which

---

[7] Lizhong Wu developed the MTC. Before him, Jay Pittman developed the original machine learning algorithms for the symbol recognizer. I developed the agent platform that the recognizer is embedded within.

weights the distribution of the team's results, examines confidence levels, and makes a recognition decision.

The features used in our MTC recognizer are as follows: Eigen components of gestural images from a principle component analysis (Fukunaga, 1990; Moghaddam & Pentland, 1997), number of strokes, normalized stroke length (the number of pixels in the pattern normalized by the total number of pixels in the image), and the image centroid (the average pixel location normalized by the size of image). Individual features differ in their relative contribution to pattern recognition. Unlike conventional approaches, the MTC does not treat individual features equally or unilaterally. Instead, we build different recognizer "members" for each extracted feature and use the most appropriate set of models and model parameters for each feature member. The member with Eigen components of gestural images is modeled by a mixture of Gaussian distributions (Duda & Hart, 1973). The number of strokes is a discrete variable and its associated member is modeled by a frequency table of the number of strokes. The normalized stroke length is modeled by a Gamma distribution (Hahn & Shapiro, 1994). The image centroid is modeled by a two-dimensional Gaussian process. In addition, multiple image sizes and Eigen dimension cut-offs are modeled. In total, there are 60 different combinations of modeling specifications and therefore 60 recognizer members are built.

Using the MTC, the symbology recognizer can identify 200 different military unit symbols, while achieving a better than 90% recognition rate.

4.5.2.2 Handwriting recognition

Paragraph's writer-independent Calligrapher handwriting recognition engine has been incorporated as an agent into Rasa.[8] Like the gesture agent described previously, the handwriting agent receives input from interactions on the paper surface in the form of digital ink. The ink is sent from the user interfaces as individual strokes of time-stamped, contextualized, x-y pairs with supplementary information. Calligrapher can recognize natural letter shapes, including cursive, printed, and mixed case. Furthermore, given a vocabulary from the domain, it can distinguish between vocabulary, non-vocabulary, and non-handwriting (other ink-drawn gestures). Combining this ability with Rasa's other

[8] R. Matthews Wesson is primarily responsible for the integration of handwriting with QuickSet and Rasa.

pen-based recognizers, Rasa can recognize and understand mixed symbolic and handwritten drawings, like that shown in Figure 2.4. Rasa's inclusion of handwriting recognition as one of its multimodal inputs is preliminary and untested; we have not yet included it as a mode in our evaluations thus far.

4.5.2.3 Speech recognition

The speech agent uses Dragon Systems NaturallySpeaking or other Microsoft SAPI-compliant engines. These are continuous, speaker-independent recognizers, though training can be used to increase their accuracy. The recognizers use context-free grammars, producing n-best lists for each phrase. An n-best list is simply the top *n* phrases recognized and their probabilistically derived scores. Rasa's vocabulary is approximately 675 words, and the grammar (Appendix B) specifies a far greater number of valid phrases.

These phrases include commands that allow users to speak the type and echelon of units (e.g. "MECHANIZED INFANTRY") in order to update Post-its already placed on the map. They can also speak distinguishing information, e.g., giving the unit a designation, such as "ADVANCED GUARD," or in a number of combinations "ENEMY MECHANIZED INFANTRY CALLED WHISKEY FOUR SIX." Users can *Update* the entity's properties (e.g., touching a company icon and speaking "FIFTY PER CENT" in order to specify that the unit is at 50 percent of capacity). They can *Remove* a unit from Rasa's map by touching it and saying something such as "DELETE THIS UNIT." Users can draw a line on the map and speak or draw its type, e.g., "FORTIFICATION," draw a closed curve and speak "LANDING ZONE ZULU," or point at the map or entities on the map and ask questions such as "WHERE IS SCOUT SIX?"

4.5.3 Parsers

Each recognizer generates a set of hypotheses, scores, and time stamps, which they forward to their respective parsers. The parsers translate these phrases into typed feature structures—directed acyclic graphs (DAGs) of attribute value pairs, which we use throughout the Rasa architecture as a way of formally representing the meaning of utterances. The parsers ensure that ambiguities in the language are modeled within these structures (i.e., within the meaning representation).

Definite-clause grammars (DCGs)
A definite-clause grammar is a set of production rules, where the head contains only a single non-terminal (e.g., head --> body), meaning that whenever the body is satisfied, the head is likewise satisfied. The body of a DCG is composed of terminal and non-terminal symbols and conditions, separated by commas. As an example, here is a small portion of the Rasa grammar. The syntax is Prolog.

```
create_unit(UNIT,COMPLETE,NUMBER) --> unit_spec(CONTENT, COMPLETE, DEF_FORCE, NUMBER),
    {unify([object:[fsTYPE:unit|_], force: DEF_FORCE|_],CONTENT,UNIT)}.
unit_spec([object:UNIT|_],[location:[fsTYPE:point,coord:_|_]|_], FORCE, NUM) --> unit(UNIT, FORCE, NUM).
unit([fsTYPE:unit, description:DESC, symbol:SYM, type:T, subtype:SUBT, equipment: EQ, echelon:ECH|_],NUM) -->
        unit_type(DESC,_SIMS,SYM,_DEF_FORCE,T,SUBT,EQ,ECH,NUM).
unit_type('M1A1 company', [modsaf], unit_USMC_M1A1_Company, _, armored, none, none, company, singular) -->
        armored,[company].
armored --> [armored].
armored --> [tank].
```

Each statement ends in a period. In the six statements, the variables (i.e., capitalized terms) are scoped locally within each statement. Words in lower case and within square brackets stand for words spoken in a sentence. It is easier to read the statements from the bottom-up, than from the top-down.

If the phrase received from the recognizer is 'ARMORED COMPANY,' then the grammar rule *armored* will hold, thus the grammar rule above it, *unit_type*, will hold passing the eight parameters up the grammar hierarchy. The rule *unit* will fire, and so on, each variable unifying with those passed along. Many rules in the grammar will have more complex binding evaluations to insure that parameters are set appropriately. The *unit_spec* rule begins to build the feature structure, the *object* type, for the language that will ultimately be used by the multimodal integrator.

4.5.3.1 Natural language parsing[9]

For this task, the spoken language consists of map-registration predicates, noun phrases that refer to, create, and label entities, adverbial and prepositional phrases that supply additional information about the entity, and a variety of imperative constructs for supplying behavior to those entities or to control various systems. Rasa uses a definite-clause grammar (for further description, see text inset below) to process the phrases that it receives as input from the speech recognizer. As the phrases are processed, the ambiguities of each are enumerated, thereby ensuring that each potential meaning is represented.

4.5.3.2 Symbol parsing

The symbol parser also produces typed feature structures, based on the list of recognition hypotheses and probability estimates supplied by the symbol recognizer. Typically, there would be multiple interpretations for each hypothesis. For example, recogni-

[9] I collaborated with Michael Johnston on the parser for QuickSet upon which Rasa's DCG was based.

tion of the ink in Figure 4.13 without additional information could result in confusion. The arc (left) in the figure provides some semantic content, but it may be incomplete. The user may have been selecting something or she may have been creating an area, line, or route. On the other hand, the circle-like gesture (middle) might not be designating an area or specifying a selection; it might be indicating a circuitous route or line. Similarly, a pointing gesture has at least three meaningful interpretations—a selection is being made, a location is being specified, or the first of perhaps many point locations is being

Interpreting a touch

Here are two of the feature structures that are constructed within the symbol interpretation code when the symbol recognition agent recognizes a single touch on the map. Note that the content of the two feature structures (a pointing gesture and a pointing selection) is structurally similar. However, the interpretation of each is very different. Selection gestures are created by the interpreter whenever one possible interpretation of the action was that the user intended to choose objects. Point gestures are created whenever one explanation was that the user intended to point at some new location, rather than an object. The interpreter examines the input from the gesture recognizer to determine if any objects were touched during the pointing operation. If none were found, a `selection_gesture` is not produced. If objects are found, references to them are placed in the POID_IDS variable. Moreover, the probability value of a `point_gesture` is reduced. Ideally, the two would never be confusable; however the ambiguity present allows for other modalities to clarify the user's intentions.

```
                 [ fsTYPE: point
                 [ coord:  CENTROID
point_gesture    [ selection: POID_IDS
```

Figure 4.12 point_gesture feature structure

```
                 [ fsTYPE: point_selection
                 [ coord:  CENTROID
selection_gesture[ selection: POID_IDS
```

Figure 4.12 selection_gesture feature structure

As previously mentioned, feature structures are directed acyclic graphs of attribute-valve pairs. In the figures above each feature contains three attributes: *fsTYPE*, *coord*, and *selection*. In a *point_gesture* the *fsTYPE* attribute is bound to the constant point. Similarly, for a selection due to a pointing action, the same attribute is bound to the constant *point_selection*. Other attributes are assigned variables, other constants, or even other feature structures. These variables will usually be bound to constants or feature structures during fusion. The complexity of these feature structures will depend largely upon the richness of each modality's contribution to the multimodal utterance.

specified. Without more information from other modalities, it is difficult to guess the intentions behind these gestures. It is the job of the parser to enumerate these possibilities as thoroughly as possible. A simple example of this interpretation process is described in the text inset below (*interpreting a touch*). In the example's figures, the brackets denote a set of values associated with the attribute on the left (e.g., the attribute point_gesture is denoted by the values fsTYPE, coord, and selection, which each have values of their own opposite the colons). In the next section, we will examine how these multiple interpretations are weighed in the multimodal integrator.

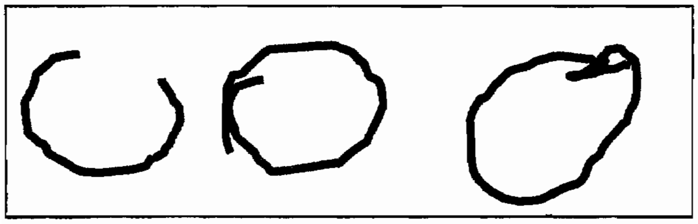


Figure 4.13 Ambiguous Gestures

4.5.4 Multimodal integrator[10]

Rasa's multimodal integration technology uses declarative rules to describe how the meanings of input from speech, gesture, or other modalities must be semantically and temporally compatible in order to combine. This fusion architecture was preceded by the seminal "Put-That-There" (Bolt, 1980), and other approaches (Cohen, 1991; Koons, Sparrell, & Thorisson, 1993; Neal & Shapiro, 1991; Nigay & Coutaz, 1995). However, as we reported in (Johnston et al., 1997) these prior approaches are limited in four ways.

- They are generally restricted to simple deictic gestural expressions.[11]
- They are primarily driven by the spoken modality; whereas first-class language exists in other modalities as well (e.g., the written symbols discussed earlier).
- They have not provided a well-understood, generally applicable common meaning representation.
- They have not provided a formally well-defined declarative mechanism for multimodal integration.

Our approach to overcoming each of these limitations supports:

- Multiple parallel recognizers and "understanders" that produce meaning fragments from continuous, parallel, coordinated input streams.
- A common meaning representation—*typed feature structures*, as described above.

[10] Rasa's multimodal integrator was enhanced as part of this work, but is based largely on the work of Michael Johnston (Johnston, 1998; Johnston et al., 1997).

[11] A deictic gesture contributes to the identification of an object (or a group of objects) merely by indicating their location.

- A general application of *rule-based constraints* that satisfy, among other things, an empirically based (Oviatt et al., 1997), time-sensitive grouping process.
- A well-understood and semantically well-defined fusion algorithm that uses declarative rules for combining compatible meaning fragments—*unification*. Unification combines both complementary and redundant information, but rules out incompatible attribute values.
- A set of declarative multimodal grammar rules that enable parsing and interpretation of natural human input distributed across multiple simultaneous spatial dimensions, time, and speech.
- An algorithm that chooses the best semantically complete, joint interpretation of multimodal input, thus allowing one mode to compensate for another mode's errors (Oviatt, 1999).

In Rasa, multimodal inputs are recognized, and then parsed, producing meaning descriptions in the form of typed feature structures. The integrator fuses these meanings together by evaluating any available integration rules for the type of input received and those partial inputs waiting in an integration buffer. Compatible types are subject to unification, and any candidate meaning combinations are subject to constraints specified in the rule (e.g., spatial, temporal, etc.). Successful unification and constraint satisfaction results in a new set of merged feature structures. The highest ranked semantically complete feature structure is executed. If none is complete, they all wait in the buffer for further fusion, or contribute to the ongoing discourse as discussed below. Each feature structure contains a timeout attribute that specifies when it is discarded from the buffer.

Due to the complexity of the fusion process, we will describe it, its rules, and the constraint satisfier more thoroughly in the next section.

4.6 Fusion

Rasa's fusion approach uses a multidimensional parser, or *multiparser*, based on chart parsing techniques from natural language processing (Carpenter, 1990, 1992). A chart parser is essentially a bottom-up parser that, rather than building up and discarding structures during the parsing process that are often built again and again, a chart parser stores these intermediate results—sub-trees, the rules that generated them, and the location (or span) of those rules in the sentence being parsed—into a "chart" or blackboard. These intermediate results make up the "edges" in a chart.

Edges in our multimodal chart are processed by declarative multimodal grammar rules. In general, these rules are productions LEFTHANDSIDE ⇐ DAUGHTER1 DAUGHTER2 … DAUGHTERN; wherein, daughter features are fused via unification, under the constraints given, into the left-hand side. Rasa's multidimensional parser for multimodal fusion is described fully in (Johnston, 1998). For completeness, we will describe the algorithm, the rule declaration, and provide an example of fusion. We will also describe improvements we have made so that the multiparser can support a mixed-initiative dialogue and rules specialized to support Rasa's needs.

4.6.1 Algorithm

The algorithm for the multiparser is outlined in the sidebar below *(Algorithm for multimodal chart parser)* and described throughout this section. The multimodal integrator is (1) constantly awaiting primarily two sorts of agent messages: a reminder to discard any edges that have expired and new input for multimodal processing. (2) If the former is received, no edges are added to the chart. So, (3) the chart is evaluated, (4) sorted, and typically (5) no edges are ready to be executed. Consequently, the only action is that (6) each edge's timeout feature is tested to determine if the edge can be safely removed from the chart. However, if (1) input from one or more modalities arrives, (2) the input is transformed into an edge and added to the chart. (3) Then every edge in the chart is applied against any valid rule. Valid rules are determined strictly by their daughter feature type's compatibility with the feature types of the edges in the chart. Once a valid rule is identified, the edges are unified into the right-hand side of the rule and the new edge is

Algorithm for multimodal chart parser

1. Received new input and turn it into edge(s) or wake-up signal
2. Add new edges to chart, if any
3. Evaluate chart
   i. Search rules in chart for daughter feature matching edge types
   ii. Perform unification within rules for all edges with matching types
   iii. Satisfy all constraints in the rule
   iv. Add new edges to chart produced from successful rule evaluation
4. Sort successfully fused edges according to the edge's probability
5. Choose and execute top edge in sort
6. Remove edges that have exceeded their timeout
7. Request a wakeup at expiration of closest timeout

pushed back into the chart. Again, the chart is (4) sorted. The top edge in the chart that is a fully formed, syntactically complete command is then (5) executed, (6) it and any expired edges are removed from the chart, and (7) a new time-out request is made.

Semantic compatibility and the actual fusion of Rasa's multimodal inputs within each rule is assured via unification over typed feature structures (Carpenter, 1992), augmented with functional constraints (Wittenburg, 1993). Typed feature structures are an extension of the representation, whereby feature structures are assigned to hierarchically ordered types. Typed feature structure unification requires pairs of feature structures to be compatible in type (i.e., one must be in the transitive closure of the subtype relation with respect to the other). The result of a typed unification is the more specific feature structure in the type hierarchy. The shared variables in the rules, denoted by numbers in square brackets in the figures to follow, must unify appropriately with the inputs from the various modalities. Typed feature structure unification is ideal for multimodal integration because it can combine complementary or redundant input from different modes, but rules out contradictory inputs.

After successful fusion, any constraints are then satisfied using a Prolog meta-interpreter. The meta-interpreter guarantees that shared variables remain bound and within scope throughout constraint evaluation. It also guarantees that all constraints are satisfied via backtracking or the fusion rule fails.

During this process, the multimodal probability of the edge is calculated. Currently, this probability is a simple joint probability estimate of the individual modality's contribution to the rule output. If both unification and constraint evaluation succeed, a new edge is formed from the structure given in the left-hand side of the rule. This edge with its associated joint probability is then added to the chart, for further processing. Edges are not consumed by rule evaluation, since they may be suitable for fusion under the same or other rules with other edges in the chart, or they may be awaiting input that has yet to arrive.

Once all compatible rules in the chart have been evaluated, all of the complete edges are removed from the chart and sorted from most probable to least likely. The edge with the highest joint probability is chosen as the prime interpretation and sent to a post-processor for execution, while the other complete edges are discarded.[12] Executions of an edge can result in various actions depending on the semantics embedded within that edge, including requests to add a new unit to a database, to update the known location of a unit, to send a query to the database for information on a unit, or to update its damage assessment.

4.6.2 Example

To demonstrate how multimodal fusion works in Rasa, let us return to the simplified example given above, in which an officer adds a new unit to Rasa's augmented map. Speaking the unit's name ("ADVANCED GUARD") generates a typed feature structure of type CREATE_UNIT with an OBJECT attribute (Figure 4.14). The OBJECT attribute contains a feature structure of type UNIT. The UNIT feature structure contains one attribute, called NAME. The NAME attribute, in this example, stores the value for the name Advanced Guard, 'AG'. To name a new unit, the user should speak the name while drawing a symbol that specifies the remaining constituents for a unit, such as the reconnaissance company symbol shown in Figure 2.4. This symbol is most likely recognized as a reconnaissance company and ultimately assigned the feature structure in Figure 4.15.

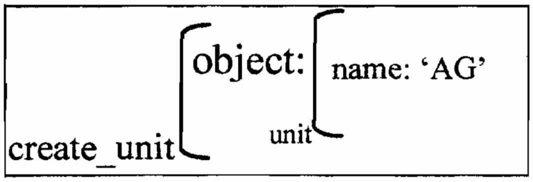


Figure 4.14 Typed feature structure from spoken utterance "Advanced guard."

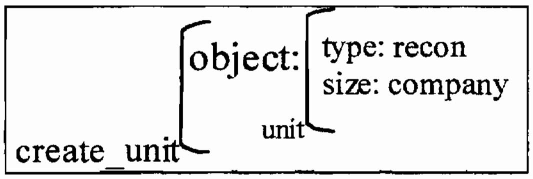


Figure 4.15 Typed feature structure resulting from drawing recon company.

One of Rasa's multimodal grammar rules, see Figure 4.16, declares that partially specified units (DAUGHTER1 and DAUGHTER2) can combine with other partially specified units, so long as they are compatible in type, size, location, and name features, and they

[12] To enable potential repair strategies, we acknowledge that discarding these fused interpretations is self-defeating.

meet the constraints. This rule will be successfully evaluated when the user attempts to create the particular unit using different modalities synchronously. DAUGHTER2 in this example is a placeholder for gestural input (note the location specification) and DAUGHTER1 for spoken input, but this need not be the case.

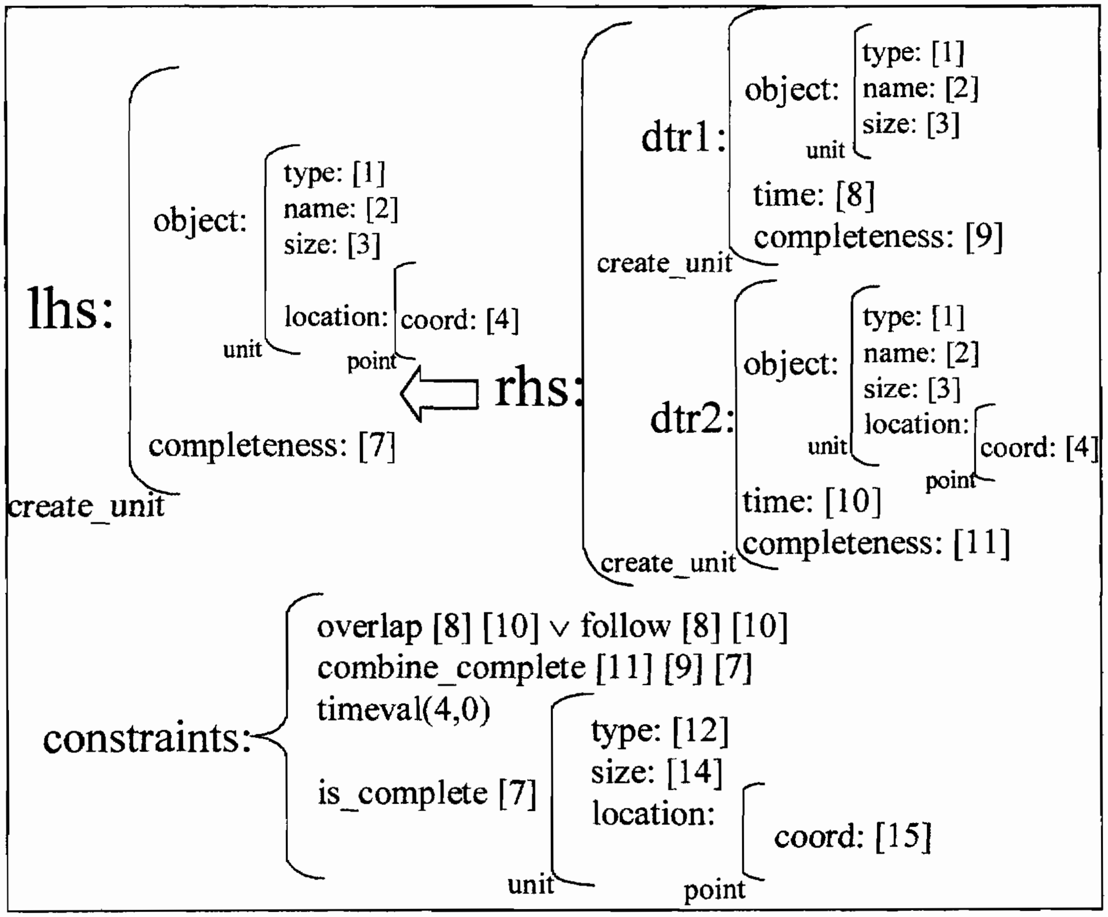


Figure 4.16 Multimodal grammar rule for partial unit fusion.

Constraints are then examined for validity. The timing constraints for this rule (the "overlap or follow" rule specification) guarantee that the two inputs will temporally overlap or that the gesture will precede the spoken input by at most 4 seconds. Figure 4.17 demonstrates partial application of the rule and shows that after fusion the left-hand-side is still missing a location feature for the unit specification. Numbers in square brackets represent variables. In the next three sections, we will describe in more detail how to declare a fusion rule, how Rasa supports a mixed-initiative discourse, and how constraints are specified and evaluated.

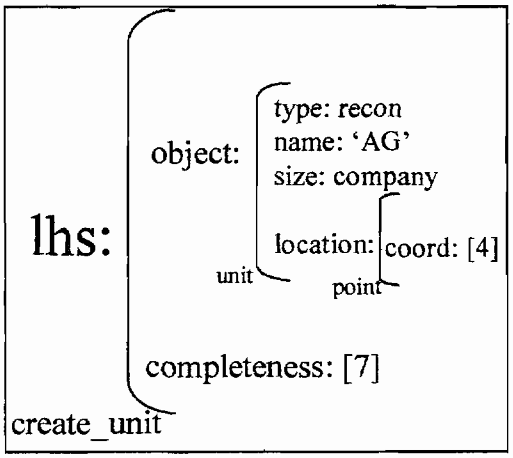


Figure 4.17 Applied rule.

4.6.3 Rules

Rasa's multimodal grammar rules, such as the rule depicted above, add to those developed by Johnston during his initial implementation of the multiparser (Johnston, 1998). What distinguishes these rules from those developed previously for Quickset is the expansion of the set beyond two input modalities, to include modes such as vision, and the ability to specify what meaning elements are required to make sense of a feature structure. This information on a command's "completeness" is used by Rasa to initiate its own clarification subdialogue in order to query the user for information missing partially complete commands. This capability is described fully in Section 4.7.2 below.

4.6.4 Constraint satisfaction

Constraints in Rasa are declared as an ordered list (potentially empty) within each rule specification and are satisfied during the multimodal fusion algorithm.

Constraints are timing-based restricting operations pertaining to intervals of modal input (i.e., speech occurred within interval T1):

- FOLLOW(T1, T2, T3) is true if T1 follows T2 by time T3.
- BEFORE(T1, T2) is true if T1 occurs before T2
- OVERLAP(T1, T2) is true if T1 and T2 overlap in time

They are numeric:

- AVERAGE(N1, N2, AVERAGE) is average if N1 and N2 are numbers.

- ROUNDED_AVERAGE(N1, N2, AVERAGE) is average rounded to nearest whole number if N1 and N2 are numbers.
- ADD(N1, N2, SUM) is sum if N1 and N2 are numbers.

They induce restrictions on modes:

- ASSIGN_MODALITY(MODE1, MODE2, MODE3) if MODE1 and MODE2 are modes of the constrained types, then MODE3 is the type declared.

And induce restrictions on objects in the scene:

- PICK_FIRST_UNIT(LIST, OBJECT_ID), examines the LIST and binds OBJECT_ID to the first unit, if any, in the LIST.
- PICK_FIRST_LINE(LIST, OBJECT_ID), examines the LIST and binds OBJECT_ID to the first line, if any, in the LIST.

They are spatial:

- CLOSE_TO(OBJECT_ID1, OBJECT_ID2, DISTANCE) is true, if OBJECT_ID1 and OBJECT_ID2 are within DISTANCE of one another.

And they enforce completeness:

- COMBINE_COMPLETE(CC1, CC2, CC3), combines the attributes needed for completeness coming from one of the edges, CC1, with that from the other edge, CC2, into the left-hand side of the rule, CC3. This constraint is usually added for rules that may not result in a complete edge.
- IS_COMPLETE(CC, FS), whether the criteria for completeness in CC is satisfied by the feature structure, FS.

Constraints can be logically evaluated in parallel using traditional conditionals (AND, OR) and antecedent/consequence relations. The ordered list is evaluated recursively using Prolog unification over a list processor. Certain constraints are added to the feature structures of the gestural and spoken linguistic constituents throughout the parsing phase as they are proposed. Others are added at the onset of fusion itself. These constraints are derived from empirical analyses or by trial and error.

4.6.5 Recent fusion improvements

Recently (Kaiser & Cohen, 2002; Wu et al., 2002) have proposed using MTC, recall description in Section 4.5.2.1) to overcome limitations in earlier approaches to multimodal fusion, including the one described here. These earlier systems simplified the computation of the fused multimodal probability for each set of constituents by assuming that individual modes are independent of one another (Sharma, Pavlovic', & Huang, 1998). Under this assumption, the cross product of the posterior probabilities of the associated constituents can be used to calculate a joint probability for the fusion of the elements. However, if the input is multimodal, the modes in these systems are, by definition, not independent (Oviatt et al., 1997). Usually, a semantic constituent in one mode only associates with a subset of constituents in the other mode. Recognition accuracies vary from one mode to another. Even in the same mode, they vary from one constituent to another. Moreover, as new modes are added, there will be some that are not utilized at any particular time. Consequently, compensating for modes that are contributing vs. not contributing to an utterance (a simple example is a unimodal gesture, such as drawing a symbol directly on the map, while ignoring microphone input) becomes extremely difficult. To overcome this deficiency, we increased the value for probability estimates of all multimodal rules in the parsers, allowing us to reduce the inherent bias toward unimodal input.[13] This solution does not scale well, nor does it deal effectively with any of the other issues identified here. Instead, a comprehensive solution would model each mode, each constituent from each mode, and each potential combination of constituents differently. This approach would compensate for a large number of recognition errors that occur in individual recognizers.

The proposed MTC technique developed in our labs appears well-suited to integrating multiple modes in this fashion, improving recognition results by 54.3% in an early test case (Kaiser & Cohen, 2002). Using MTC, multiple recognizers of different modes become the members of a statistical integrator (Wu et al., 2002). Multiple teams can be built in the MTC integrator and trained to coordinate and weight the output from different modes. Each team establishes the posterior estimate for a multimodal command, given a corpus of multimodal commands. The committee of the MTC integrator analyzes the empirical distribution of the posteriors and establishes the N-best ranking for

---

[13] The reader will note that a perfect probability score (1.0) can only be diminished when combined with another modality, especially when other modes probability estimates may be inherently weaker (i.e. substantially less than 1).

each act of multimodal fusion. However, Rasa does not currently employ these new MTC techniques for multimodal integration.

4.7 Multimodal mixed-initiative discourse

In this section, we will discuss Rasa's ability to support a mixed initiative discourse (Biermann, Guinn, Hipp, & Smith, 1993; Novick, 1988; Smith, 1997; Walker & Whittaker, 1990) with its users. Mixed-initiative describes a property of systems that support a dialogue in which either the user or the system can take up responsibility for continuing the conversation. Rasa and its predecessor, QuickSet, were the first systems to demonstrate that confirmation acts performed by multimodal systems should occur after fusion rather than before. Section 4.7.1 presents that argument and describes how confirmations are presented and acted upon in Rasa. Building on similarities to Smith et al.'s "Missing Axiom Theory" (Smith, Hipp, & Biermann, 1995), for goal-oriented subdialogue execution, we extended the constraint satisfaction meta-interpreter described above to support multimedia queries for values missing from feature structure attributes. Consequently, Rasa's mixed-initiative discourse is multimodal in both its input and output. The discourse implementation is described in Section 4.7.2.

4.7.1 Confirmations in multimodal systems

Systems that attempt to understand natural human input make mistakes, *even humans*. However, humans avoid misunderstandings by adopting several strategies. One of these strategies is *confirming* doubtful input. Prior to QuickSet, multimodal systems either did not confirm input or confirmed only their primary modality—speech. This is reasonable, considering the evolution of multimodal systems from their speech-based roots. However, simply confirming the results of speech recognition was problematic—users had the expectation that whenever a command was confirmed, it would be executed. Moreover, we observed that confirming speech prior to multimodal integration led to three possible cases where this expectation might not be met: ambiguous gestures, non-meaningful speech, and delayed confirmation.

The first problem with speech-only confirmation was that the gesture recognizer produced results that were often ambiguous. Figure 4.13 demonstrates how, oftentimes, it is difficult to determine which interpretation is correct. Some gestures can be assumed to be fully specified by themselves (at right, an editor's mark meaning "cut"). However, most rely on complementary input for complete interpretation (Lefebvre, Duncan, & Poirier, 1993; Morin & Junqua, 1993; Oviatt, 1997). If the gesture recognizer misinterprets the gesture, failure will not occur until integration. The speech hypothesis might not combine with any of the gesture hypotheses. In addition, earlier versions of one of our supported speech recognition engines were limited to a single recognition hypothesis per spoken utterance, and one that might not even be syntactically correct, in which case integration would always fail. Finally, the confirmation act itself could delay the arrival of speech into the process of multimodal integration. If the user chose to correct the speech recognition output or to delay confirmation for any other reason, integration itself could fail due to time sensitivity in the multimodal architecture.

In all three cases, users were asked to confirm a command that could not be executed. An important lesson learned from observing early users of Quickset is that when confirming a command, users think they are giving approval; thus, they expect that the command can be executed without hindrance. Based on these early observations, we conducted an experiment, which showed that by delaying confirmation until after modalities have combined (i.e., late confirmation) the human-computer dialogue in multimodal systems is enhanced (McGee et al., 1998). Prior to this finding, spoken language recognition results were solely used for confirmation, which always occurred before integration (i.e., early).

When comparing late with early confirmation, our evidence shows that in the late confirmation mode: 1) subjects completed commands in fewer turns (the error rate and the number of turns per command were reduced, resulting in a 30% error reduction); 2) they complete turns at a faster rate (the number of turns per minute increased by 21%); and 3) they completed more commands in less time (the number of commands per minute increased by 26%).

By delaying confirmation until after fusion, we are able to disambiguate hypotheses using the *multimodal language specification*, i.e., the rules that allow modalities to combine (Section 4.6.3). Since different modalities tend to capture complementary information we can leverage this facility by combining ambiguous spoken interpretations with dissimilar gestures in our rules. For example, we might specify that a selection gesture (circling) combines with the spoken utterance "SELECT" to produce a selection command of the circled unit, rather than relying solely on the gesture. Another way of disambiguating the spoken utterance is to enforce a precondition for the command. We call these and related techniques that impart discriminatory power into the fusion process *multimodal disambiguation* or *mutual disambiguation* techniques (McGee et al., 1998; Oviatt, 1999).

We discovered that late-stage confirmations lead to three improvements in multimodal dialogue. First, because late-stage systems can be designed to present only feasible commands for confirmation, blended inputs that fail to produce a feasible command are immediately flagged as a non-understanding and presented to the user as such, rather than as a possible command. Second, because of multimodal disambiguation, misunderstandings are reduced, and therefore the number of conversational turns required to reach mutual understanding can be reduced as well. Finally, a reduction in turns combined with a reduction in time spent leads to reducing the "collaborative effort" in the dialogue.

There are two likely reasons why late confirmation outperforms early confirmation: implicit confirmation and multimodal disambiguation. Heisterkamp theorized that implicit confirmation could reduce the number of turns in dialogue (Heisterkamp, 1993). In a speech-only digit-entry system, Rudnicky showed that implicit confirmation improved throughput when compared to explicit confirmation (Rudnicky & Hauptmann, 1992), and our results confirm their findings. Lavie and colleagues have shown the usefulness of *late-stage disambiguation*, during which speech-understanding systems pass multiple interpretations through the system, using context in the final stages of processing to disambiguate the recognition hypotheses (Lavie, Levin, Qu, Waibel, & Gates, 1996). However, we have demonstrated empirically elsewhere (McGee et al., 1998), the advantage in combining these two strategies in a multimodal system.

It can be argued that implicit confirmation is equivalent to being able to undo the last command, as some multimodal systems allow (Vo & Wood, 1996). However, commands that are infeasible, profound, risky, costly, or irreversible are difficult to undo. For this reason, we argue that implicit confirmation is often superior to the option of undoing the previous command. Implicit confirmation, when combined with late confirmation, contributes to a smoother, faster, and more accurate collaboration between human and computer.

4.7.1.1 Implementing confirmations in Rasa

Confirmations are projected visually onto the maps just as they appear when ultimately confirmed, but highlighted to distinguish them from accepted commands. Accompanying the visual evidence is a verbal request for the user's confirmation, such as "CONFIRM: BARBED WIRE." Because of the findings mentioned in the previous section, Rasa adopted implicit confirmation as a default behavior; thus, commands are executed when the user accepts them or when the next command occurs, unless otherwise cancelled.

Confirmation acts are treated much like the commands themselves. Each act, flagged as a confirmation request, is passed to interested user interfaces via the agent architecture. These confirmation requests contain a reference to the command (e.g., moving a unit) that will be executed, if the confirmation request is accepted. Unless the interface (i.e. map) was the originator of the command, these confirmation requests are usually ignored.[14] Otherwise, the confirmation is displayed visually and the user interface generates a request for the text-to-speech agent to vocalize the confirmation request. If the user who is presented with the confirmation rejects the request, the confirmation is eliminated. Although other potentially valid commands, such as the fused command with the second-highest probability, could then be retrieved from the multimodal integrator and presented for confirmation, Rasa currently eliminates these contenders during the execution of the confirmatory act.

The user can point anywhere on the map and use spoken commands to accept or reject an outstanding confirmation. He can use the *confirm* and *cancel* buttons first mentioned in Section 4.3 by pressing them. Each of the buttons produces a sound when hit, much like the user would expect from a graphical user interface. Users can make confirmations explicit rather than implicit. When this happens, confirmations stack up and can be confirmed by pointing at the projected confirmation object and saying "CONFIRM" or by hitting the confirmation button and accepting all outstanding requests. Users can also

[14] We have used this distribution mechanism to support Wizard-of-Oz studies. In these cases, specially designed interfaces do not ignore the confirmation requests, but can control them in the user's place.

disable confirmations entirely. When they do so, all commands are executed as soon as fusion occurs.

4.7.2 Clarification subdialogues

During feature structure evaluation, rules can execute that instruct Rasa to communicate with the user or another agent in order to make requests for the values of certain attributes that might be otherwise missing or incomplete in the current feature structure. Ultimately, among other effects, the request can add a new edge to the chart, merging the missing attribute into the feature structure. Each feature structure can carry along such a rule, which specifies the particular features that are needed to "complete" the structure; henceforth, we will refer to these as *completeness criteria*. The required attributes necessary to ensure that a feature structure is complete are domain-dependent, but such constraints can be declared either bottom-up during the parsing process, where they can be added one part at a time, or top-down during declaration of the fusion rules. Constraint satisfaction is described in detail below in Section 4.6.4.

For example, Figure 4.18 shows the completeness criterion for units. This feature structure captures the attribute names for each of the attributes that must have a value before the feature structure is deemed complete. Currently these values are not typed. In the example shown here, the criterion stipulates that every unit creation is only complete when it contains an object that is a unit, with values for *type, size*, and *location.* The location feature must have a feature-structure of type *point*. This point feature must have a value for its *coord.* Other attributes are allowed and can be added without restriction.

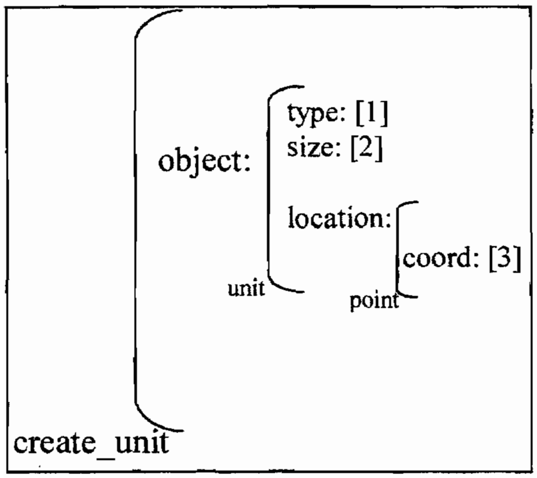


Figure 4.18 Completeness criteria for units.

Rasa uses this information during constraint satisfaction (Section 4.6.4) to produce queries whenever one of these values is missing. After a delay that is also declared within the constraints, Rasa asks the user for the position of the unit that is missing a location feature. Users respond by placing the Post-it note on the map or by disconfirming the operation and throwing the Post-it away. For example, if the Post-it note representing the reconnaissance company has not been placed on the map within 10 seconds, Rasa would respond by saying "WHERE IS THE RECONNAISSANCE COMPANY CALLED 'ADVANCED GUARD'?"

4.7.3 Confirmation and clarification summary

Within Rasa, we have implemented repair and clarification subdialogues as part of a growing strategy to support more complex human-machine discourse. Though discourse rules can be declaratively specified in Rasa to promote mixed-initiative, collaborative dialogue, Rasa's capability for extended dialogue is more limited than current dialogue systems (e.g., Trindikit (Larsson & Traum, 2000)). Even so, our work seeks to extend the state-of-the-art in two ways. First, the determination of when confirmation occurs within a dialogue, especially a multimodal dialogue, can have far-reaching impacts on the system's performance and its usability. Experiments with QuickSet have demonstrated that confirmation of human multimodal input best occurs after fusion, reducing the potential for presenting useless, even confusing, information.

Second, the expectation that all multimodal input and relevant information for fusion should be processed bottom-up is questionable. Therefore, Rasa's design allows for both bottom-up and top-down processing of information. A consequence of supporting both top-down and bottom-up processing within Rasa's integrator is the flexibility with which new data can be handled. With bottom-up processing, new information can be injected as edges into Rasa's multimodal chart parser for fusion. Declarative rules fuse the new edges into a shared, high-level representation. With top-down processing each edge can have an attached rule that specifies information required to further process the edge toward one final state another. These attachments allow Rasa to process a set of potential discourse rules via constraint satisfaction.

Confirmation and clarification are especially important in the domain outlined within this thesis due to the critical nature of the accuracy of the captured information. Other domains or situations of use may have less of a requirement for confirmation because the impacts of errors are in some way minor. However, since confirmations can be seen as a proactive undo capability that appears well-suited to augmented environments such as Rasa, future work is needed to determine when undo may be preferred over implicit, late confirmation.

4.8 Summary

In this chapter, we have described Rasa, a system that captures human multimodal input in order to augment a military commander's map tool set. We have given background on the QuickSet pen-tablet system and its agent architecture (AAA), upon which we based Rasa. We have described the hardware and software configuration of Rasa, how to set it up, how information flows within its multi-agent architecture, its functional components, its fusion system, rules, constraints, and dialogue processing.

"System *reliability* and responsiveness are the *sine qua non* elements of usability. If the system is unavailable, it cannot be used. If the system is unreliable, users will avoid it regardless of how good it may be when it works. (Gould, 1995)"

Chapter 5 Empirical investigation and evaluation

In this chapter, we present empirical findings of a comparison between Rasa and the paper tools that it augments. We demonstrate that the additional cost of using Rasa over paper and pen alone is the cost for repairing recognition errors. We also show that despite a considerable number of errors experienced with this initial prototype, users favor Rasa over paper alone. We present evidence that by combining paper and digital tools, we have constructed a hybrid system that supports the continuation of work in spite of power, communications, and hardware or software failures.

5.1 Introduction

In the next section, we present the goals of this comparison. Section 3 of this chapter describes the methodology applied, and Section 4 the results.

5.2 The study

Rasa's development was informed by on-site observations at exercises conducted at the US Marine Corps Base at Twentynine Palms, California and US Army Fort Leavenworth. An initial version of the system was pilot-tested at the First Marine Expeditionary Force, Camp Pendleton, California. In that pilot test there were nine subjects, eight male and one female, six of whom worked individually and three as a team.

Subjects ranged in experience from two officers who were mostly unfamiliar with maps and their requisite symbology language to experienced commanders with 20 years on the battlefield. Qualitative descriptions of their opinions are included below. Numerous improvements were made, and the system was then evaluated at the Center for Human-Computer Communication's human interfaces laboratory on the OGI School of Science & Engineering campus. In this study, six male subjects from the Oregon Army National Guard used Rasa and paper alone to track an ongoing military situation. The purpose of these studies were to address the following questions:

Is it possible to design computationally augmented artifacts and processes:

1. That retain all of the important properties of the physical objects, such as their malleability, resolution, permanence, and tangibility?
2. That are as easy to use as or easier to use than the natural, physical tools?
3. That are resistant to power and digital communication failures?
4. That do not significantly increase the task-based performance cost of digitally capturing the information represented by the physical objects?

Questions 1 and 2 were addressed through questionnaires filled out by the subjects after their participation. With respect to the system's failure-resistance (Question 3), we measured (a) the amount of work stoppage when a failure occurred during the task; and (b) the amount of recovery time. With respect to question 4, we measured (c) the cost of annotating, moving, and removing Post-its, and (d) the number, types, and cost of errors imposed by the system.

5.3 Method

The subjects acted as the "map plotter," whose job was to ensure that the annotated map was as accurate as possible, while attempting to establish situational awareness for a report to their commander. One of us acted as the "radio man" for the subjects, providing the reports as time elapsed, and verifying the reports as questions arose; other researchers ensured the system remained functional, conducted the interviews, and took notes as interesting behavior developed.

5.3.1 Instrumentation and materials

Two Windows NT© workstations were configured with Rasa. The first was an HP Vectra, 850 MHz Pentium III workstation with 512 MB of RAM running all of the Rasa agents described above and attached to a front-projected SMART Board™. The other system was a Fujitsu Stylistic 2300, 233 MHz Pentium II hand-held with 96 MB of

RAM running the Rasa user interface only, with the Cross iPenPro digital pen tablet attached to it. The speech recognition chosen was the Dragon Systems' speech recognizer, Naturally Speaking.

Subjects interacted verbally with the map and Post-it notes using an EmKay close-talking wireless microphone. An iPenPro radio frequency-based digital pen tablet from Cross Company's Pen Computing Group enabled pen-based interaction with the Post-it notes. By placing the officers' paper map on the SMART Board, touch-sensitive interaction with the map was available. A visible overlay of the digital objects that were created was projected onto the SMART Board whenever Rasa was online.

We videotaped all subjects' interactions. Moreover, digital ink was logged and spoken audio was recorded for each interaction, either at the map or when Post-it notes were being created. Messages passed from one agent to another within the agent architecture over a wireless LAN (including results of speech and gesture processing, parsing, multimodal fusion, etc.) were logged.

5.3.2 Training, task, and procedure

Each subject was introduced to Rasa's use, to the map, and to the simulated military scenario, the training for each was approximately 15 minutes (see Appendix C and Appendix D). During the Rasa introduction, each person used the system to place several units on a map and move, update, and remove them. They were instructed on how to perform confirmations, how to cancel commands that were in error, and, in case Rasa or the user had made an error, how to reconcile Post-its that had already been physically added to the map.

Subjects were instructed that Rasa would not understand everything that might need to be added to the map. For example, there was no symbol for a downed pilot, even though the scenario contained such. Indeed, in service of scenario realism, the subjects were asked by the scenario no fewer than four times each to add symbols to the map that the system was not trained to recognize. Invariably, these attempts produced errors. Their instruction was to make multiple attempts to ensure that the paper map and the projected symbols from Rasa were reconciled, but the most important thing was to keep the *paper* map up-to-date and reconcile the computer system later, if need be.

Each subject was asked to complete a short written form regarding his or her participation in the experiment. Finally, at the conclusion of the simulation, an interview was conducted to elicit open-ended responses.

5.3.3 Task scenario

The scenario was a 90-minute simulation of a two-day-long event. During the scenario there were 14 reports of new units on the scene (requiring the construction of a symbol on the Post-it denoting the unit), 13 reports of unit movement (requiring Post-it movement on the map), 2 reports of units leaving the scene or being destroyed (requiring the units' removal from the map), several reports that required no immediate action at the map, and 1 update (a report of damage requiring the subject to further augment the unit with spoken language). These are only the minimum number and types of commands possible; many of these operations, such as updates, were often used either to complete an operation only partially specified in one modality or during error repair. Moreover, subjects often chose to adjust their physical tokens (i.e., Post-its) multiple times, resulting in additional commands. For example, several subjects performed a move command correctly, but revised the placement of the unit on the map anyway.

5.3.4 Simulated power failure

Approximately midway through the scenario, we simulated a system failure by disabling Rasa. Unaware of the deception, subjects were told to ignore the failure and continue with their task using paper alone, just as they would in the field. After approximately nine reports, Rasa was turned back on and subjects were instructed to reconcile the paper map information with that displayed by Rasa, which projects on the map the last known position of the units in its database. The subjects were asked to update the position of those units that moved and to create digital counterparts for any new units created only on paper while Rasa was disabled.

5.4 Results

Combined, the subjects produced 171 *initial* commands (first-time *Create*, *Move*, or *Update* commands) to Rasa with an additional 80 *related* commands (e.g., naming a new unit already added to the map, which is also an *Update*) for a total of 251 commands. Multimodal utterances fashioned to repair Rasa errors, correct subject's mistakes, or recover from system failure accounted for an additional 191 commands. On average, for each command it took the system 1.22 seconds (s=1.98) to respond, accounting for the time taken to recognize, fuse, and distribute the results for confirmation. Throughout the results section, we will use the standard symbols: $\bar{x}$ to represent mean observation, s to represent standard deviation, and n to represent number of observations.

For the basic operations (i.e., *Move* and *Create*), Rasa emulates the typical way that this information is captured by the paper artifacts. Therefore, it was hypothesized that Rasa would not contribute to the cost of using these tools.

5.4.1 Comparing paper and Rasa

The Paper condition was measured during the so-called "outage" of Rasa. The breakdown of commands was 2/3 in the Rasa condition and 1/3 in the Paper condition. There were a total of 55 initial commands and 9 repairs in the paper condition. Because there were fewer repairs and updates in the Paper condition, the proportion of Rasa to Paper commands was 5 to 1.

Table 5.1 summarizes the quantitative results for the operation time (e.g., time spent drawing the unit and putting it on the map = *Create* operation time) of the initial commands given to Rasa that did not produce an error, and compares them to the same operations performed in the paper condition without error. The t-test results are a two-tailed comparison (unequal variances) between the two modes.

Table 5.1 Means and standard deviations of operations in Paper and Rasa mode conditions and two-tailed t-test results.

Operation (sec)	Paper			Rasa			Comparison
	$\bar{x}$	s	n	$\bar{x}$	s	n	*t-test*
Create	38.29	12.48	17	32.10	13.30	54	$p < 0.0898$
Move	39.83	18.59	35	33.50	14.31	28	$p < 0.1318$

5.4.1.1 Creates

When an officer received a report that a new unit had been spotted, he proceeded to draw the unit and then place it on the map. They spent approximately 7 seconds drawing the Post-it, another 4 seconds walking to the board, and the remainder of time, approximately 24 seconds, finding the grid for placement; times were averaged over both Rasa and paper conditions.

Surprisingly, the mean paper *Create* time, taken over all subjects, was slightly higher than the mean Rasa *Create* time. The t-test between paper and Rasa showed only a non-significant trend toward differences between the two modes. A two-way analysis of variance showed no interaction between subject and mode. This analysis was not able to show that individual subjects interacted differently with regard to observed times to perform *Create* operations, whether using Rasa or paper.

5.4.1.2 Moves

Subjects adopted their own strategies regarding how to move units. Some would find the unit to move, grab it from the map, locate its new place on the map, and then put it down. Other subjects would find both locations on the map, and then move the unit from one location to the other. Most subjects used each strategy at one time or another but relied primarily on the latter when interacting with Rasa. When moving units, more time was spent finding the unit and its new location (24 sec.) than actually moving the unit (less than 6 sec.). Because of these variations, move operations were measured from the moment that the report was received, in order to include the time spent searching for locations on the map. The variations in subject behavior would account for the strong subject effect observed when we performed a two-way analysis of variance with interactions on the time users spent either looking for locations on the map or for particular units before moving each unit: $F=11.1325$, $Pr(>F)=2.78e\text{-}7$. A two-tailed t-test on error-free move times, which includes the search time, does not demonstrate that the time to move units in the paper condition is discernibly different from the Rasa condition.

5.4.2 Rasa-only operations

Two operation types were measured that occurred only in the Rasa condition (1) *removal* of units from the map, and (2) the *updating* of unit information (such as status and identification) via spoken interaction and pointing. The performance summaries of correctly executed (initial or related) commands are given in Table 5.2. There were no *removal* commands in the Paper condition due to time constraints. Updating Post-its in the paper condition (e.g., via handwriting after placement on the surface of the board) was not possible at the time.

Table 5.2 Means and standard deviations of Rasa-only operations.

Operation (sec)	$\bar{x}$	s	n
Other updates: status, etc.	1.85	0.64	58
Removal of unit	2.15	0.67	15

5.4.3 Errors

Table 5.3 is a classification of errors found during our experiments. Twenty-eight percent (57) of all initial or related commands resulted in an error of one type or another. Commands were flagged as *recognition errors* whenever the system misunderstood what the user intended and the utterance was one that the system should have understood based on its grammar. Of the 10 *recognition errors*, one was a gesture recognition error; the remainder were speech errors. All of the speech errors except one were single-word utterances. If the subject made a mistake by producing an utterance form that the system was not programmed to understand, this utterance was tagged as a *performance error*. These could be either "out-of-grammar" errors or simple user mistakes. If the system made an error due to a correctable experimental or system design flaw, we counted these as *system errors* (e.g. static produced by the wireless microphone). At times, the scenario indicated that the subject should perform an operation that the subject would consider valid but that could not be properly recognized by Rasa. These operations were classified as *guaranteed errors*. These were out-of-grammar conditions (e.g., "DOWNED PILOT"), some instances of which will likely occur with any grammar. In addition to the errors reported in Table 5.3, which were made while using Rasa, the subjects made two performance errors on initial commands while in the paper condition.

Table 5.3 Errors for Rasa: initial or related (251) commands

Error type	#	Percentage
Recognition	10	5.3%
Performance	27	10.8%
System	15	6.0%
Guaranteed	15	6.0%

5.4.4 Effect of system failure

Subjects responded to the simulated failure of Rasa with only a moment's hesitation in the task. The explanation of the failure and direction to proceed was the only time spent dealing with the failure, since the tools for continuing the task were unchanged.

5.4.5 Cost of repairing errors or recovering from failure

Table 5.4 summarizes the human performance time for individual repair operation times that resulted in a correct command. In the Rasa condition, 82 of the 191 repair attempts resulted in successful completion of the command (i.e., errors corrected on the first attempt); the remaining errors required more than one repair attempt (i.e., spirals [15]of length two or more). In this section, we examine the performance cost for correcting errors and report on the number of spirals based on recognition errors observed in the experiment.

Table 5.4 Means and standard deviations of error-free repair operation times (74/82).
+, ++, *, **: table items referenced below.

Operation (sec)	$\bar{x}$	s	n
Repair of Create (gesture)+	32.97	12.95	9
Repair of Create (verbal)++	3.11	0.84	19
Repair of error: Move*	5.30	8.23	20
Recovery of failure: Move**	2.01	1.10	15
Repair of Update	1.88	0.64	19
Repair of Delete	2.11	0.857	16

[15] An error spiral develops when users are forced to or choose to repeat the same error-prone utterances.

5.4.5.1 Recovering/repairing a create

Subjects could use two methods to correct an error made when placing a new unit on the map. They could draw the symbol representing the unit again, almost doubling the amount of time needed to complete the command (21 cases, 9 were successful on the first attempt).[+] Otherwise, if the symbol was correct, they could point at the unit on the map and speak its type and size, which is the very same technique used when recovering from system failure (53 cases, 19 were successful on the first attempt).[++] If the repair was successful on the first attempt, this operation added on average only an additional 10% to the operation time. A t-test demonstrates that this reduction is significant (t = 17.8419, df = 59.423, p-value = 2.2e-16).

5.4.5.2 Recovering/repairing a move

Recovering from errors in moving units is accomplished by repeating the simple pointing operation: once on the old location and then at the new location.[*] For repairs, this operation took significantly less time than original move operation attempts (t = 10.3797, df = 70.126, p-value = 8.122e-16). The move operation took even less time after Rasa itself has recovered from a failure (i.e. after the downtime that we simulated).[**] In this case, the new location is marked by the Post-it, while Rasa projects the old location from the data that it stored prior to the outage. The mean time for recovery of move operations after a systems failure was only 6% (2.01/33.50 seconds) of the original time (t = 10.059, df = 31.027, p-value = 2.232e-11). Since subjects typically need not search again for the unit that is being moved or its new location, this time is eliminated during repair and explains the reduction in time seen here.

5.4.5.3 Error Spirals

Recall that there were 191 repair attempts, of which 82 were corrected on the first attempt. The remaining 109 repair attempts clustered into 38 spirals. Five of these spirals were never completed. Eight of them were exclusively made up of *recognition* errors (i.e., error spirals). These were three spirals of length 2, two of length 3, one of length 4;

and two of length 5. Noteworthy is that all of these spirals were from single-word utterances ("DELETE," "ENEMY," "FRIENDLY," "DELTA," and "ALPHA").

5.4.6 Compound cost of errors

In addition to comparing individual error-free operations between Rasa and paper, we compared the end-to-end time for each operation including any subsequent repairs. Unlike other error types, initial guaranteed and system errors were consistently followed by spirals that can be eliminated. Moreover, they can be eliminated, a point we will argue in more detail below.

Once the correct location is found during the initial command attempt, there is typically no searching for coordinates during repair. Therefore, to effectively and consistently compare Move operations, we excluded this search time when it occurred. Via verbal feedback errors in Creates were identified by the subjects prior to this search activity. Otherwise, Table 5.5, compares total times for operations. These total times include any repairs, system presentation of results, confirmation by the user, etc. Recall, confirmations were not required, but would be inferred by the system if the user began a new command. Consequently, only 41 of the original 54 *Create* and 29 of the original 35 *Move* commands from Table 5.1 can be used for this comparison.

Table 5.5 Means and standard deviations of total time in performing error-free operations, compared to those of operations that had errors, including all repairs, confirmation, and presentation time.

Operation (sec)	Error-free operation			Compound repair operation			Comparison
	$\bar{x}$	*s*	*n*	$\bar{x}$	*s*	*n*	*t-test p-value*
Create	40.11	10.456	41	60.20	23.845	18	0.002648
Move	38.42	15.801	29	42.76	26.994	11	0.6246
Updates	3.478	1.033	58	9.324	16.199	17	0.1564

Create operations that include repairs take an average of 50% more time than non-repairs. Moves that include repairs take only slightly more time than non-repairs (not measurably significant). Updates require much more time for repairs than for non-repairs, but the variation in repair times is so large that the difference is not statistically significant. These comparisons provide an estimate of the significance that errors have on the efficiency of subjects' performance. The lengthening in the times of the error-free operations in this table corresponds to the addition of confirmation and presentation times.

5.4.7 Subject Response

Subjects provided feedback on their experiences with Rasa, first by completing a post-test questionnaire (Appendix E) and then in an open interview session (transcript excerpts in Appendix F). The summary of questionnaire responses is given in Table 5.6. We have included in the table, the subjective responses from our nine USMC pilot test subjects, who experienced considerably more errors, with an earlier version of the system, than these six National Guardsmen.

Table 5.6 Subjective responses to post-test questionnaire.

Response		Performance		Compared to paper		Preference		# of errors		Error Correction		Work stoppage		Recovery	
Always too long	1	Impossible		Not nearly as easy to use		Paper preferred over Rasa		Too many		Extremely difficult		Complete		Impossible	
Occasionally too long	1	Impeded significantly	2	Not as easy as paper		Paper preferred over similar systems		More than I would have liked	3	Difficult		More than 10 minutes		Extremely difficult	
Within tolerance	9	Impeded somewhat	1	As easy as paper	7	No preference	1	Acceptable	7	Moderately difficult	2	Several minutes		Difficult	
Better than expected	3	Impeded slightly	3	Easier than paper	7	Yes. With improvements	7	Few	3	Easy	13	Briefly	1	Moderately difficult	2
Immediate	1	Did not impede	6	Much easier than paper	1	Rasa	7	Almost none or none	2			None	14	Easy	13
		*Improved**	3												

*Subjects added a new value to the performance measure to indicate their belief that Rasa improved it

Here we summarize the statements made during the interviews, during the pilots and the final experiment, drawing on a few notable quotes from the transcripts. See the complete transcripts from the final experiment in Appendix F. Subjects told us that Rasa was as easy or easier to use than paper alone.

"THE BENEFITS THAT I SEE FOR THIS SYSTEM, FOR WHAT WE ARE DOING IN OUR JOB RIGHT NOW IS THE FACT THAT IT'S EASIER TO KEEP TRACK OF WHAT'S GOING ON."

"[I THOUGHT RASA WAS AS EASY TO USE AS PAPER] WITH A LITTLE MORE TRAINING ON THE SYSTEM."

Generally, Rasa did not impede performance. However, some subjects modified the questionnaire to note that in their estimation Rasa improved their performance of the task.

"[THE PERFORMANCE] WAS QUICKER THAN ME MOST OF THE TIME."

"[WHAT I DO IN THE FIELD IS] SIMILAR TO WHAT YOU'RE DOING HERE. THE ONLY DIFFERENCE IS THAT WE HAVE TO MANEUVER IT AT THE KEYBOARD NOT AT THE MAP. AND [THE MAP] IS A LOT QUICKER."

"IT MADE UPDATING EASY. ALL I HAD TO DO WAS... IF THE UNIT HADN'T BEEN NAMED, JUST POINT ON THE UNIT, NAME IT. IF THE UNIT HADN'T BEEN MOVED, DO THE MOVE. UPDATING THE ELECTRONIC SIDE WAS NOT A NIGHTMARE, WHICH IT WOULD HAVE BEEN WITH [OUR CURRENT COMMAND AND CONTROL SYSTEM]."

"THE SYSTEM PERFORMED WELL. I SAY THAT GOING FROM A MANUAL SYSTEM, WHERE IT TOOK A WHILE TO RECIEVE INFORMATION, BEING ABLE TO PROCESS IT, AND THEN PUTTING IT ON THE BOARD. THIS RASA SYSTEM ALLOWED [ME] TO DO IT ALL AT ONCE, REDUCING THE TIME TO UPDATE THE BOARD. YES, CORRECT. [THE SYSTEM RESPONDED ALMOST IMMEDIATELY.]"

Rasa was even preferred to paper alone. Subjects often attributed this preference and improvement to Rasa's elegant simplicity and reliance on known pen and paper techniques.

"IT'S AS CLOSE TO REAL-TIME PLOTTING AS YOU CAN GET."

"SIMPLISTIC OBSTACLE MAKING: DRAW A LINE. TELL IT WAS IT IS. AND THERE IT IS."

We were concerned about the number of errors that were encountered. However, most subjects told us the number of errors was acceptable, and those present were easy to correct. As is customary with these types of evaluations, the subjects blamed themselves for a majority of the errors even when they exposed a system or design flaws.

"IT WASN'T MORE RECOVERING FROM RASA'S ERRORS, BUT MY OWN."

"I HAD A LITTLE TROUBLE WITH DELETING THE UNIT, BUT THAT WAS MORE OF A PROCEDURAL THING."

"I WOULD SAY THAT RASA MADE, OUT OF 10, 3 OR 4 ERRORS: 30-40%, OF THE MISTAKES. THE REST WERE USER ERRORS."

Finally, after working through the simulated failure, subjects generally believed work continued unhindered, and that recovery from errors was easy.

"[WHEN THE SYSTEM FAILED, I WAS ABLE TO KEEP MY WORK GOING.] WITH THE SYSTEM STOPPED, COULD I CONTINUE WITH THE DIGITAL SYSTEM? NO, I COULDN'T. BUT COULD I CONTINUE THE POSTING. OH HECK YEAH. NO PROBLEM."

"[WHEN THE SYSTEM FAILED,] IT WAS JUST A MATTER OF SAYING, 'IT'S DOWN, PLEASE FORGIVE US. DRIVE ON.' I GUESS. SO THAT WAS... MAINTAIN PLOT. NOT VERY LONG, I GUESS."

A number of subjects understood the importance of maintaining a paper record while executing at digital speeds:

"I LIKED THE FACT THAT YOU DID A FAILURE IN THE MIDDLE OF IT. THAT'S REALISTIC, VERY REALISTIC. THAT'S REALISTIC IN PEACE-TIME!"

"YOU'RE NOT DOING TWO SEPARATE RECORDS. YOU'RE DOING ONE: THE COMPUTER AND THE PAPER AT ONCE. THAT WAY YOU HAVE NOT TRANSLATION ERRORS WHEN YOU TRANSLATE FROM ONE TO THE OTHER."

"YOU'VE GOT TWO WAYS TO TRACK IT. RIGHT NOW, PUTTING ON THE PAPER ICONS AND HAVING THE SYSTEM TO USE AT THE SAME TIME. IT GIVES YOU TWO THINGS TO LOOK FOR. LIKE WHEN THE SYSTEM WHEN DOWN, TO PUT IN THE SYSTEM THE ICONS I'D POSTED BEFORE WAS RELATIVELY EASY, BECAUSE THERE WERE ALREADY THE ICONS PAPER AND THE COMPUTER-GENERATED WHERE THEY BELONGED AND IT WAS EASY TO SEE JUST BY GLANCING AT THE MAP WHERE YOU NEEDED TO MAKE THE CHANGES AT."

"WE HAVE TO USE BOTH [PAPER AND DIGITAL SYSTEMS]. IN OUR SITUATION, WE HAVE TO. BECAUSE WE MOVE, YOU CAN'T KEEP [THE COMPUTER SYSTEMS] GOING. I SUPPOSE YOU COULD WITH A LAPTOP, BUT IT'S JUST NOT FEASIBLE. WE NEED TO BE ABLE TO PICK IT UP AND RUN."

"YOU'VE GOT TO HAVE THAT BACKUP."

Overall, Rasa was looked upon quite favorably.

"YOUR SYSTEM'S GOT A LOT OF GOOD USES."

"I THINK IT MAKES THE JOB EASIER."

"I DIDN'T NOTICE THAT YOU WERE RUNNING WINDOWS."

"[I WOULD WANT SOMETHING LIKE THIS IN MY OWN TOC,] IF IT WAS AT LEAST AS EASY TO DO AS THAT WAS AND I WASN'T STANDING IN MY OWN LIGHT. I THINK THAT THOSE WERE THE ONLY TWO THINGS THAT REALLY BOTHERED ME. YEAH I THINK IT WOULD DEFINITELY BE AN ASSET."

"I WOULD PREFER PAPER AND DIGITAL AT THE SAME TIME. IF I HAVE THAT OPPORTUNITY. YES, I WOULD. IF FOR NO OTHER REASON THAN THE JUMPING IN THAT TOC. BECAUSE AS THINGS GET MOVED AROUND IN THE BACK OF A HUMVEE..."

5.5 Summary

In this chapter, we described a study which evaluates Rasa's ability to support one task common to Command Post maps, map plotting. The purpose of the study was to determine two things: (1) whether Rasa was an improvement over the current set of tools (map and Post-its) for the task and, if it was, (2) what is the potential cost of adding Rasa to those tools. Specifically, we hoped to judge the fitness of Rasa for the task and its support for coping with system failures and the inevitable mistakes users make.

Based on the interviews, quantitative, and qualitative examinations set forth in this chapter, we argue that our thesis claims are incontrovertibly supported by the Rasa prototype. We show that additional cost of using Rasa appears to be negligible when we intuitively compare it and our empirical studies specifically compare it to our subjects' paper-based map plotting. Moreover, given our subjects' insightful comments we have good reason to believe that these claims are more generally supportable to a wide variety of tools with a similar set of requirements.

In the next chapter, we will discuss the implications that arise from the findings presented here. We will enumerate the current limitations of Rasa, discuss proposals for overcoming those shortcomings, and argue the general relevance of this work.

"Human beings usually use computers not because they **want** to interact with them but because they want to reach their goals..." (Kaptelinin, 1996, p.49)

Chapter 6 Discussion

In this chapter, we will discuss the results of our experimental evaluation of Rasa (Section 6.1), examine limitations of the current techniques (Section 6.2), predict avenues for future work (Section 6.3), and address questions regarding some of the larger issues of design that are raised by Rasa (Section 6.4), and finally we address the lifetime of the ideas represented by Rasa (Section 6.5).

6.1 Experimental Findings

We have described an experiment that measures the cost of using Rasa compared with that of using the paper tools it is based upon. This comparison demonstrates that the only new cost in using Rasa is due to the repair of errors. Despite these costs, users prefer Rasa to their paper tools because they gain access to computing and do not have to give up what the paper has to offer. Moreover, due to the distinct persistence properties of paper combined with those of computer systems, Rasa exhibits a synergistic robustness seldom seen in computing tools.

The primary benefit of the system in its present form is to establish a link from the paper-based artifacts and task to digital systems. Because of this (1) natural redundancy is provided when the system fails, (2) communication of situational information can be made digitally, (3) current work practices are adopted with little modification, (4) the advantages of current tools are retained (persistency, malleability, portability, etc.), (5) collaboration is instantly available to remote personnel, (6) additional training is minimized, and (7) duplicative and error prone work and systems are eliminated.

Tangible tools must demonstrate (1) whether they are an adequate replacement for the existing set of physical tools, and (2) whether they are a more effective replacement than traditional computer interfaces. We attempted to answer the first question with this initial research (i.e., "Is Rasa at least as good as, if not better than, *the real thing*?").

Rasa uses the paper tools of the command post and thus retains the properties that people come to expect from the physicality of these tools. For example, the robustness of Rasa is like that of paper, making it resilient in the face of computing failure, unlike any other computing system. However, these and the other benefits of paper must be measured against the cost of preserving their use in particular working situations vs. that of adopting and adapting to new tools.

Although we cannot prove the null hypothesis, we found no evidence that the cost of using Rasa is significantly greater than the cost of using paper for error-free individual operations. Moreover, system response and human activity that we would typically characterize as computer interaction were both brief. Since Rasa parallels the use of the paper map and Post-its (i.e., most of the activity mimicked the expected physical actions), adding additional tasks only when there are mistakes made by the system, this finding is not surprising. It is the repair of these mistakes that, at least for some individual operations, measurably increases the cost over that of the paper tools. Indeed, reducing the frequency and cost of these errors is an important area of future work.

The use of Rasa and of paper for this task is dominated by the time to find locations and objects on the map. In order to investigate the potential for Rasa to improve the paper-based process, a small, follow-on study was conducted in which 5 male subjects each issued twenty commands whose purpose was to get Rasa to find various grid locations. For example, users said something like "SHOW LOCATION FIVE ONE SEVEN, TWO THREE SIX." In response, Rasa displayed a circle at the desired location and spoke the coordinates. Across all subjects, the mean time taken to find a location was 6.9 seconds, including time needed to correct recognition errors. Overall, Rasa provided a 93% utterance recognition rate, and a 99.5% word recognition rate. Comparing the 7 seconds spoken time to find a coordinate with the 24 seconds observed in the main study, we hypothesize that Rasa could improve the overall process substantially. Future research needs to test this hypothesis with military users in a realistic scenario.

Finally, we were surprised that, in general, subjects' preferred Rasa to paper and found it as easy or easier to use, despite the quantity of errors. We attribute many of these errors to the limited training subjects received and the prototypical nature of the system. The users' positive reaction could be explained by a Hawthorne effect. However, we attribute it to their ability to ignore Rasa's deficiencies, because the tool is part of their natural work environment, and therefore its "cost" is negligible despite the errors.

6.2 Limitations of Rasa

Rasa has several limitations, including an incomplete vocabulary and grammar. The number of sensors limits its perceptual ability. Though collaboration is supported, multiple simultaneous users working side-by-side cannot use the system simultaneously. Finally, Rasa has only a simple understanding of the context in which the tool is practically used. In this section, we will briefly discuss these challenges and proffer lines of investigation.

6.2.1 Natural language

In order to cover the language of this work practice more adequately, we would need to collect more observational data on multimodal language used in command posts to further refine our recognitional components. These refinements would include developing a robust natural language grammar, spoken language models, adding new symbols to the symbol recognizer, etc.

There is an important set of design choices that must be made regarding the quantity and naturalness of the language that we support with these types of tools. There is a trade-off between the efficiency and robustness of the grammar and its naturalness (Cole, Mariani, Uszkoreit, Zaenen, & Zue, 1996). This trade-off also affects the amount of training required for the tool. The more natural the interaction (language, gesture, etc.), the less training needed. So, do we increase the number of utterances, thereby enhancing naturalness, yet potentially adding more opportunities for recognition error? Or, does the new language more accurately match the user's natural model of the tool's use, thereby reducing out-of-grammar conditions? Most likely, both of these options should be supported to some degree and must be managed in tandem as the design evolves. Since the primary limitation of Rasa is its error rate, the importance of managing and evaluating this trade-off should not be underestimated.

Recall that a significant number of spoken commands and all error spirals observed during the experiment were due to single word utterances. These short, typically one word phrases cannot be sufficiently discriminated from one another due to the lack of available data (i.e., phonemes: word sounds). Therefore, one way of reducing errors is by eliminating single word utterances from the grammar. However, eliminating these natural, short phrases would tend to make the grammar more unwieldy and unnatural. Hence, though the recognition errors in the grammar would be reduced, the number of out-of-grammar phrases would generally increase.

Another solution is somehow to increase the probability of successful recognition of both these and other phrases in Rasa's natural language grammar. Rasa's fusion and dialogue components can employ a more detailed model of the context of the tool's use to improve its understanding of the users' activity. Information about the tendency of officers to move naturally from one activity to another is a potential input into Rasa's hierarchical fusion system. Such context can also greatly aid the error recovery process. Whereas today Rasa is essentially stateless with respect to its current dialogue and error repair strategies (i.e., a new command is expected equally as much as an attempt to repair an error), systems that can recognize potential failure can be prepared to identify input that is specifically targeted toward repairing the previous error. The ability for Rasa, and other multimodal systems, to have a greater understanding of likely multimodal input classes in terms of the context of the tool's use is one area of potentially fruitful research.

6.2.2 Improving Rasa's senses

From Section 4.2.1, recall that Rasa currently employs haptic, auditory, and visual sensors to observe spoken and written language, touch input, and object (Post-it) placement and movement. As described in the previous section, Rasa's understanding of its users' language use could be improved in a number of ways, thereby reducing the errors observed in our empirical studies. However, this is not the only perceptual input that could be further developed to improve Rasa's ability to understand the task. In this section, we will discuss issues around the precision of sensory inputs as well as limitations in its visual perception.

6.2.2.1 Sensory precision

One of the biggest limitations in the current prototype application is the imprecision of locations given to entities from the current sensors developed. The SMART Board is certainly precise enough to capture the width of users' fingers. However, the actual precision needed for these objects can be many times smaller than a finger width, depending on the scale of the mounted map.[16] To accommodate the usual resolution of maps mounted in command posts and the precision of location data for entities on them, other methods (sensors or otherwise) of data capture must be added. As recounted in Section 3.6, we have added a machine vision component, which provides redundancy in the location information. However, though not yet fully tested, we do not expect with current technology that we will obtain the precision needed from this additional sensor for increasing the accuracy of location information in Rasa.

One method of attaining the needed precision, which we gave a brief description of in Section 6.1, is speech input. In this case, we would add the ability to have users provide simultaneous, redundant input on the location of particular entities (via vision, touch, and speech) by placing the Post-it at the location and speaking precise coordinates at the same time. The speech, touch, and visual input all process a location. Errors can be reduced by ensuring that, within some level of tolerance, the three inputs agree, while the spoken language portion provides the most detailed description of the location. Rasa would provide visual and audible feedback at the precision requested. Another possible approach would be to provide additional variable modes for switching from one mode of input to another as errors arise (Mankoff, Hudson, & Abowd, 2000; Oviatt, Cohen, & Wang, 1994). Such modes could include handwriting within a form specifically designed for capturing coordinates, a soft- or hardware keyboard, etc.

However, to reduce the complexity of providing precise input, Rasa could also adopt more direct manipulation techniques. Like the cancel and confirm buttons, we

---

[16] Some subjects used a stylus or pencil that they picked up in the laboratory to improve the resolution of their pointing gestures.

could add a numeric keypad to the map and into the workings of the system's multimodal dialogue. Some will argue that supporting keyboard input from the beginning of the design would have ensured fewer errors, but this argument misses the larger point.

Keyboard input alone does not solve the problems outlined above. Instead, redundancy and immediate feedback ensure that errors are obvious. Rasa's ability to naturally support simultaneous, multi-sensory input should reduce both the frequency of errors entering the system via the human interface (as opposed to via digital systems such as GPS), as well as the latency of incorrect information (i.e., the time that such errors remain uncorrected in the system).

6.2.2.2 Improving its Vision

In this environment, Post-it notes are likely to overlap and thus impair their segmentation from the scene and subsequent recognition by machine vision and understanding systems. Moreover, these systems typically rely on target objects that are specifically constructed so as to be highly discriminable and usually unique. The drawings on the Post-its are very similar in some cases, and mismatches may occur because traditional pattern matching algorithms can only distinguish objects that differ sufficiently at a given fixed scale. An increase in the resolution of the current cameras or a design that includes multiple cameras at differing focal lengths may increase the likelihood of correct recognition of the objects.

A sensor for hand tracking has not yet been added. However, we recognize that being able to understand how hands are used to manipulate objects could substantially improve Rasa's perception of what is occurring during task execution. With a hand tracking sensor, the system could predict when objects may have moved by assuming that they are unable to move on there own and that a hand must pass by the object in order for it to be lifted from the map. A hand tracking visual sensor could also be used to improve the Rasa's ability to recognize when objects are being touched, by adding to the evidence of such an action at the SMART Board and expanding to cases that include multiple hands, which SMART Boards cannot currently deduce. In order to support multiple users effectively, Rasa must adopt a strategy for sensing multiple hands touching the map. These techniques are part of a set of tracking algorithms developed by May and his colleagues (May et al., 1998) for their Human-Information Workspace or HI-Space. Machine vision techniques such as those cited here and those used by the earliest systems for augmenting paper (Wellner, 1993) are valid tactics for supporting this strategy.

6.2.3 Sensors and fusion

Experiments to evaluate the kinds of complex issues identified above are typically conducted in a Wizard-of-Oz fashion (Dahlbäck, Jönsson, & Ahrenberg, 1993; Grosz, 1977; Oviatt, Cohen, Fong, & Frank, 1992). Due to its agent-based infrastructure Rasa is particularly well-suited to being tested in a Wizard-of-Oz environment. This testing will become increasingly important as the number of multimodal inputs increases and the evolving context becomes far more complex. Thus far, our experiments and demonstrations have shown that Rasa's fusion rules readily accommodate two simultaneous inputs sufficiently. Though there are no theoretical limits imposed by the fusion engine's design on the number of inputs that can be included in one of the integration rules, it is still uncertain whether Rasa will scale effectively as the number of simultaneous inputs increases.

Smart homes, offices, and other augmented environments like those described herein will obviously require a substantial increase in the number of sensors, recognitional systems, and multimodal inputs. Hence, the limits of any fusion architectures' ability to scale effectively, including our multiparser's, will be crucial to solving the larger problems presented. Indeed, since adding a machine vision sensor, the number of potential simultaneous inputs available to Rasa has increased from two to three. Testing the effect of this new mode on Rasa's performance is certainly a timely challenge.

Since Rasa's multimodal input will be observed by multiple sensors, no single modality should be evaluated in isolation. Instead, we will need to conduct experiments that demonstrate which language constituents are typically used in combination at any given moment. This evidence can then be used to drive a probabilistic combination of inputs in Rasa (Kaiser & Cohen, 2002; Wu et al., 1999a, 1999b, 2002). A generic task model, such as the one described above, would guide the interpretation of each modality's inputs in the context of the task. However, the system should also factor in when the actual user input diverges from the model, and develop a means for tracking and adapting to this variance, such as a weighted mapping of deviations from the task model. Care must be taken in developing these models, because over-fitting them will produce behaviors that are too context-specific, even for the same user. Task models are not the only models that could improve Rasa's interpretation of context. User models, coordination models, and others could fine tune the performance of this kind of system.

As the number of sensors accumulates, the amount of ambiguous or erroneous (i.e., unintended) input in the fusion engine will also increase. One significant problem with current multimodal systems is determining when to terminate consideration of input and when to accept any conclusions drawn on the input. For example, so long as an input resides within Rasa's fusion engine, it is considered viable and evaluated each time another piece of input appears. Therefore, the longer an edge resides in the multimodal integrator's chart, the more likely that it will be ultimately considered and, consequently, the longer fusion itself takes, while it considers each edge against each matching rule. The other extreme is the premature elimination of input from the fuser. The failure condition in this case is not seen in the fusion performance, but is instead an error condition in which the intended action is not recognized because one or more of the viable edges of input have timed-out due to some unforeseen inactivity. Even more troublesome is that missing input or fusion failure is such a common event (it is a normal condition of having fusion of input from multiple sensors) that the condition cannot be reliably presented as feedback.

Traditional multimodal systems restrict themselves to input that is more or less, simultaneous in nature (Oviatt et al., 1997). However, the natural multimodal input that is common for the Rasa tool has both elements of simultaneous and non-simultaneous input. Consequently, a stricter reliance on the ordering of events and other elements of the context of the task are needed to effectively process these inputs.

6.3 Future work

In addition to future work that would address the limitations described above, there are several other next steps for the ideas expressed in this thesis. In this section, we will discuss three of these. First, we have identified collaboration and teamwork as an important component of the work situations that are somewhat addressed by paper tools. However, these tools clearly do not empower teaming among participants that are not in the same location. Below in Section 6.3.1, we discuss how designs like Rasa may be extended to help provide both improved collaboration at a distance, as well as side-by-side. In Chapter 5, we compared Rasa to its paper-based predecessor. However, many systems have already been installed to support these work practices. Indeed, some users have already overcome the barrier of learning and adapting to new technology offerings. To be thorough, we must compare Rasa to these existing systems. Section 6.3.2, proposes such a comparison. Finally, in Chapter 2, we compare our ethnography of the use of map tools in military command posts to similar work situations. In Section 6.3.3, we propose to further study these other work environments and design Rasa-equivalents to support them.

6.3.1 Teamwork

Command post staff do not merely track battles, nor do they restrict themselves to a single map. Rather, they coordinate multiple maps that support different job roles, deconflict information reports, and collaborate at multiple levels. Paper tools aid these tasks by making the information visible at a high-resolution in a shared physical location. However, the costs of coordination, deconfliction, and collaboration remain high. With Rasa's weaving the computing layer into the natural tools, these benefits are obtained without giving up the attractive properties of the paper tools themselves or increasing the cost associated with using them.

We mentioned above that one thing that Rasa cannot do effectively is reconcile potentially competing inputs on the same physical map from multiple users. However, Rasa does enable remote collaboration among multiple maps. For example, an officer can circle a set of Post-its on the map's overlay and have her "gesture" and the units being circled projected onto her remote collaborator's paper map. This could be especially useful in that currently, the cost of achieving this level of collaboration in real command posts—even simply capturing free-hand drawings on these overlays—is prohibitively expensive. However, Rasa's agent architecture (Kumar, Cohen et al., 2000) supports the multi-casting and intelligent filtering of messages via a centralized network of brokers. Consequently, digital ink and data objects can be readily distributed, the former via intelligent filtering and the latter via replication of a centralized data repository.

We intend to experimentally validate our hypothesis that Rasa is better equipped to support the kind and quality of collaborative activity present in command posts than current paper systems and existing computational systems developed for battle tracking. As noted above, we must first further develop Rasa's sensing to include tracking of both people and their gestures on and around the map board. Rasa's rules and constraints will need to be extended for these new types of inputs, and support of the new probabilitistic fusion engine based on the MTC will likely be necessary.

6.3.2 Comparison to existing tools

A more thorough experimental examination of the benefits of tangible tools would surely reach beyond the single-user, empirical laboratory study conducted in this case to a series of multi-user field trials, directly comparing paper, Rasa-like tools, and state-of-the-art GUIs. Despite the apparent disadvantages of digital systems for users, and users' resulting preference for paper, it is important to measure any solution against the perceived technological state-of-the art. Similar comparisons between multimodal and GUI-based systems for simulation initialization have shown both a preference and an efficiency advantage for multimodal map systems (Cohen, McGee, & Clow, 2000). This research showed that we can expect significant performance improvement of multimodal systems (on the order of eight to ten-fold) over traditional WIMP interfaces for map-based tasks, including the costs associated with errors in recognitional systems. Future research will compare Rasa's tangible multimodal interface, with the standard GUIs employed in modern command and control systems.

6.3.3 Bridging other chasms

We have argued that the design of tools that support collaboration and teamwork in environments where human lives are at risk must find innovative ways of introducing information technology non-disruptively. By uniting effective, robust, physical tools with sensing systems and multimodal recognition and fusion, the approach described here is one method for delivering information technology to these highly conservative user populations. Reaching these communities (medical, air traffic, etc.) is an important area for future pursuits.

6.4 The Larger Issues

In this section, we discuss the larger issues that Rasa raises in the context of the overall motivation for tangible systems, and how to design for evolutionary usability.

6.4.1 New motivation for tangible systems

Recent work in tangible systems assumes that they provide benefits over more traditionally designed computing systems: benefits that spring primarily from their reliance on haptics. However, this assumption bears some critical examination. In this thesis, we started with a set of tangible tools that have remained in use despite the introduction of replacements for them. Our argument is that these tool's (e.g., paper maps) physical benefits, tangibility among them, had already been proven. By designing systems that integrate these tangible tools into a pervasive computing environment, we no longer need to assume that tangibility provides some benefit. It does so in their current work practice and therefore does so as an element of Rasa.

With this initial design, we seek to demonstrate that tangible systems can be used to help introduce technology into work practices that have heretofore staunchly retained their preference for physical tools. These systems include those developed by people outside of the high-technology arena. It also includes tools developed for safety and mission-critical systems, examples of which are provided in Chapter 2.

6.4.2 Evolving high technology

A consequence of our approach is that high-technology computing solutions can be outfitted to existing sets of physical tools such that the transition to "high-tech" solutions can appear more gradual than other methods. Moore points out that conservative end-users appreciate non-disruptive technology, technology that requires only gradual changes to the current work practice or technology that has been proven to provide amazing productivity improvements through widespread adoption (Moore, 1991). We have shown how Rasa's design benefits the more cautious customers in one particular domain by marrying high-technology to existing low-technology tools. This marriage allows the more conservative end-users to potentially leap-frog today's graphical user interfaces and become adept at the inevitable mix of GUIs with multimodal interfaces. Hence, one of the primary benefits of Rasa is that it provides disruptive technology in the guise of non-disruptive technology.

6.5 The Lifetime of Rasa

We have argued that Rasa, and its design in general, provide an elegant solution to the integration of new technology with traditional conservative work practices and tools. Some will counter that this approach will eventually be supplanted by future technologies such as disposable, high-resolution, and touch sensitive digital paper (Brown, 1996, 1997; Gibbs, 1998; Negroponte, 1997); extremely small, low-cost, and inexpensive batteries; compact and energy efficient, wireless networking technologies; and extremely small tracking components (GPS and RF). There are two problems with this argument. First, *all* of these technologies must be available and reach a level of maturity that rivals paper before they can be put to effective use in the safety-critical environments at stake here. (Imagine disposable, digital Post-its with wireless communications and better than millimeter-accurate tracking.) Second, methods for fault-tolerant software engineering (Halang, 1999) still must be enhanced or, regardless of the advances in hardware, systems will continue to fail due to software faults. As Halang states,

> "When assessing their dependability, hardware and software have completely different qualities. Hardware is subject to wear. Faults occur at random and may be of a transient nature...Software failures, on the other hand, are neither caused by wear nor by environmental events such as radiation or electric impulses. Instead, all errors are requirement analysis, design, or programming errors, i.e., they are of a systematic nature, and their causes are *always* (latently) present."
> *(Halang, 1999, our emphasis)*

Whether hardware or software related, faults will always be present, and our design is tolerant to faults in both hardware and software, inasmuch as what remains during a fault is the current set of robust paper tools. None of the problems mentioned above are simple; and though one can imagine solutions for some in the next quarter century, solutions for all seem unlikely.

6.6 Summary

Officers prefer paper because it is fail-safe, malleable, lightweight, cheap, and high in resolution. By developing Rasa, we hoped to achieve the integration of these benefits with those that we can expect from computation—e.g., data distribution, remote collaboration, etc.

First, our experiments demonstrate that Rasa preserves the elements of the physical objects on which the tool is based. Second, although Rasa adds a cost of use that is directly related to the number of mistakes its recognition systems make, it can compensate for this lost time with computational aids, such as finding coordinates (Section 6.1). We acknowledge that the abundance of errors present and limited support for teams of users currently limits Rasa's introduction. However, the plans for future work outlined in this chapter present a comprehensive program of research toward the development of multimodal platforms for augmenting physical tools that should lead to the adoption of this design approach in a number of domains.

Chapter 7 Conclusion

This work exemplifies a new approach to the design of invisible computer interfaces that augment existing physical tools. In this final chapter, we summarize the thesis and its contributions to computer science and describe a vision for classes of future applications based on Rasa.

7.1 Summary

We have observed that users in a military command post augment objects with written language as part of the task of creating a map of the battle to support situational awareness for military decision-making. The properties and artifacts of that environment, and its tasks are similar to many other decision-making environments. Users of these environments are constantly dealing with uncertainty and making decisions where peoples' lives are at stake.

From our field ethnography, interviews, and observation, we derived a set of constraints on the design of systems to support command and control end-use tools. These constraints call for a design that is robust to failure typically associated with computing systems, and that ensures the tools and the information that they carry with them are understandable at all times and to all observers, just as they are with the physical tools they use at present.

Indeed, people use physical objects as robust placeholders of meaning during decision support tasks in order to communicate efficiently about objects of great import. Given the constraints (*minimal* perturbation of existing work, language use to promote *human understanding* and *performance* of augmentation, *malleable* changes that are incremental over time, and *robustness* in the face of failure), we provided a general comparison of tools used for the augmentation of physical objects, such as tags, barcodes, etc. We showed that existing tagging technologies were unable to meet the constraints imposed by the constraints, and we were force to turn to the method that the users themselves were using to tag the tools: language.

Consequently, we developed Rasa, a system that allows users to augment the preexisting physical tools (Post-its and maps), creating links to the digital objects they represent, which can then be manipulated both physically (by moving them on the map) and verbally (by annotating the map or the Post-its with language). We show that a multimodal system can be constructed that is quite capable of observing this type of input with recognitional components based on machine vision, spoken language understanding, and pattern recognition of handwriting. Taking advantage of a highly flexible multidimensional parser for fusion of input, Rasa extends the multimodal architecture previously developed for QuickSet, by refining its fusion processing, adding support for mixed-initiative human/agent clarification dialogue of missing information, and supplementing QuickSet's set of rules and constraints with new ones that support this dialogue and extend the types and formulations of multimodal input supported.

In a controlled study of its use, we confirmed that our initial claims—that a system can be designed that 1) is robust to computing failure, 2) is as easy to use as current tools, 3) is as malleable as physical artifacts like paper, 4) does not measurably increase the cost of interacting with the tools, 5) is as portable as paper, 6) has the resolution of paper, and 7) requires little training to adopt—are attainable today.

We discussed the implications that can be draw from our experiments with Rasa, we point out several of the limitations of the currently developed solution, and we propose several areas that deserve further research. We end that chapter by pondering some of the larger issues that Rasa raises in the design of computational tools that must be attuned to issues of health, safety, and welfare.

7.2 Summary of contributions

This thesis contributes to extending the state-of-the-art along several sub-fields in computer science: namely, tangible, multimodal, and intelligent interface design and implementation. Foremost, Rasa is the first tangible computing system of its kind to successfully extend physical tools by observing and processing the means by which people already augment these tools—a combination of spoken and written language, specifically reference and suppositions, and manual attention. As a consequence, the Rasa design has a patent pending (McGee, Cohen, & Wu, Pending).

This primary contribution is supported by findings from our field ethnography; namely, the discovery of new constraints on the design of computing interfaces for mobile command and control systems. Support for these new constraints is magnified when compared with similar findings from our colleagues working in other mission- and safety-critical domains. Our field research demanded that we reexamine why computer systems fail to meet these users' needs. This examination of constraints, which appear sufficient to meet the special needs of ground combat commanders, led us to a unique design which enables these commanders to capture, assess, and modify information by continuing to use their physical tools (McGee et al., 2000). The design's contribution is in developing a methodology for the extension of physical tools through observing their natural augmentation via language.

We embodied this design in Rasa (McGee & Cohen, 2001), subsequently adding a fourth modality—vision (McGee, Pavel, Adami et al., 2001) to overcome limitations imposed by current touch board technology. The design itself is unique, describing a multi-agent architecture for multimodal interaction that reaches beyond prior systems in its support for more modes, substantially more complex rules, and an active dialogue layer including a novel confirmation process. Our final contribution to this thesis is an empirical evaluation of Rasa (McGee et al., 2002), the first empirical evaluation of a tangible user interface.

According to Kaptelinin, "numerous experiments have shown that activities mediated by symbolic tools often undergo three developmental stages: (1) the initial phase, when performance is the same with and without a tool because the tool is not mastered well enough to provide any benefits, (2) the intermediate stage, when aided performance is superior to unaided performance, and (3) a final stage, when performance is the same with and without the tool but now because the tool-mediated activity is internalized and the external tool (such as a checklist or a visualization of complex data) is no longer needed. (Kaptelinin, 1996, p. 62)" We believe that tools like Rasa can be designed to circumvent these stages by supporting the evolutionary development of physical tools that have already been internalized, rather than replacing them with a radically new and complex technology that insists these cognitive shifts take place.

7.3 A vision

Consider planning activities, where what is at hand is limited to everyday objects, things like scraps of paper, the change in your pocket, a few pencils, some rocks and pebbles, and the sand from the desert nearby. This is sufficient for us humans to meet and plan, establish roles, sketch out a simple map, identify objectives, etc. Why do we continue to choose to execute these tasks with so-called primitive tools when we have access to the outstanding computational resources we have today? One of the reasons is that those simple, everyday objects can mean so many different things, so readily. Figure 7.1 below illustrates one such recent planning activity. Sand table mock-ups such as these continue to be a valued resource for military decision-makers.

Figure 7.1. Planning on a realistic terrain model (a.k.a. Sand Table). Photo courtesy of Ed Swan.

Our scratching in the dirt, for those few moments as our team gathers, is not just a place in the sand, but a vista of the terrain upon which we stand, or perhaps some faraway location. The pencils, the change in our pockets, those rocks and pebbles, come to represent whatever name we give them in whatever context is relevant. When placed on "the terrain" (our scratched out version of it), these objects come to represent particular real-world entities, even colleagues.

One long-term vision for this research is to allow computers to observe this type of interaction and contribute to the human discourse there in meaningful ways. For example, the computer could overlay the real world with projected simulations of future events from its understanding of the underlying domain models, such as terrain, teamwork, and the like. It could record, transcribe, and translate a journal of the planning process automatically into whatever digital format was suitable for publication, coordination, and remote collaboration. It could examine the proposed plan against a library of plans, and determine whether flaws similar to those in prior plans are in the current one. It could enable collaboration with remote participants, with certain physical objects acting as stand-ins for people.

How is this possible? Just as language is able to transform "blank slates" like Post-it notes into objects that represent military units in Rasa, so too can future versions of Rasa take advantage of spoken, symbolic, gestural, and visual input to transform the scene describe above into a three-dimensional planning environment. We see this as the next logical step, over the course of a multi-year program, for this research.

Brooks (Brooks, 1991) proposes that HCI will only be truly valuable as a discipline, when we can provide designers with three things:

1. A broad background of comparative understanding over many domains.
2. High-level analyses useful for evaluating the impact of major design decisions.
3. Information that suggests actual designs rather than simply general design guidelines or metrics for evaluation.

To do so, we must develop abstractions that "discard irrelevant details while isolating and emphasizing those properties of artifacts and situations that are most significant for design" (Brooks, 1991).

By examining the situated use of current tools in several domains, analyzing their properties, and proposing a design methodology that subsumes the physical tools and augments them, thereby turning late-adopters into early-adopters, we have in a small way met Brooks' challenge.

Appendices

Biography

David Ray McGee was born in Yakima, Washington on the 28th day of January, 1966. Three weeks later, he was adopted into the loving home of his parents, Jack and Sondra McGee, in Richland, Washington. He graduated with honors from Richland High School. During his senior year, in August of 1983, he joined Battelle's, Pacific Northwest National Laboratory (PNNL) as a student intern. He began his undergraduate career in 1984 spending the first two years at the University of Washington, in Seattle. He completed his B.S. in Computer Engineering at Washington State University, Tri-Cities in 1995, while working full-time at PNNL, and consequently was promoted to the rank of scientist there.

David's work at OGI, and in the Center for Human-Computer Communication, contributed to the following areas: multimodal interaction, augmented reality, and multi-agent systems. His dissertation research, funded under DARPA's Command Post of the Future program, demonstrates how people use language, both written and spoken, to enhance paper tools. He has built a system, Rasa, capable of understanding this language that enables these paper tools to be coupled to digital environments. This melding of digital and physical information spaces via understanding natural multimodal language is the first of its kind. Due to this work and his contributions to other areas of research, in 2000 David was nominated by his colleagues in the Computer Science and Engineering department as a "student of achievement."

David is now the Chief Scientist and Technical Lead of the Rich Interaction Environments thrust area in Pacific Northwest National Laboratory's Information Sciences and Engineering Division. He is leading a group of scientists toward developing new methods of interacting with pervasive computing environments that are: multimodal, socially-adept, environmentally adaptive, and capable of rich multimedia narrative. He is a principle investigator on the Geospatial Intelligence Information Visualization (GI2Viz) program with the Advanced Research and Development Activity (ARDA).

References

Austin, J. L. (1975). *How to do things with words* (Second ed.). Cambridge, MA: Harvard University Press.

Azuma, R. T. (1997). A survey of augmented reality. *Presence: Teleoperators and Virtual Environments, 6* (4), pp. 355-385.

Bentley, R., Hughes, J. A., Randall, D., Rodden, T., Sawyer, P., Shapiro, D., et al. (1992, Nov.). "Ethnographically-informed systems design for air traffic control" in the *Proceedings of the Conference on Computer-Supported Cooperative Work (CSCW'92)*, pp. 123-129.

Biermann, A. W., Guinn, C. I., Hipp, D. R., & Smith, R. W. (1993, March). "Efficient collaborative discourse: A theory and its implementation" in the *Proceedings of the ARPA Human Language Technology Workshop.*

Billinghurst, M., & Kato, H. (1999, March 9-11). "Collaborative mixed reality" in the *Proceedings of the International Symposium on Mixed Reality*, Yokohama, Japan, pp. 261-284.

Bolt, R. A. (1980). Put-That-There: Voice and gesture at the graphics interface. *Computer Graphics, 14* (3), pp. 262-270.

Brooks, R. (1991). Comparative task analysis: An alternative direction for human-computer interaction science. In J. Carroll (Ed.), *Designing Interaction: Psychology at the Human Computer Interface* (Cambridge: Cambridge University Press.

Brown, C. (1996). Polymer Material Promises an Inexpensive and Thin Full-Color Light Emitting Plastic Display. *Elecontric Design* (Jan. 8), p. 42.

Brown, C. (1997). Pumped polymers promise photonic technique. *Electronic Engineering Times* (April 7), p. 34.

Carpenter, R. (1990). "Typed feature structures: Inheritance, (in)equality, and extensionality" in the *Proceedings of the ITK Workshop: Inheritance in Natural Language Processing*, pp. 9-18.

Carpenter, R. (1992). *The logic of typed feature structures*. Cambridge England: Cambridge University Press.

Cohen, P. R. (1991). "Integrated interfaces for decision support with simulation" in the *Proceedings of the Winter Simulation Conference*, pp. 1066-1072.

Cohen, P. R., Cheyer, A., Wang, M., & Baeg, S. C. (1994, March). "An open agent architecture" in the *Proceedings of the AAAI Spring Symposium*, Madison, Wisconsin, pp. 1-8.

Cohen, P. R., Johnston, M., McGee, D. R., Oviatt, S. L., Pittman, J. A., Smith, I., et al. (1997, November). "QuickSet: multimodal interaction for distributed applications" in the *Proceedings of the International Multimedia Conference*, Seattle, WA, pp. 31-40.

Cohen, P. R., McGee, D. R., & Clow, J. (2000, April). "The efficiency of multimodal interaction for a map-based task" in the *Proceedings of the Applied Natural Language Programming Conference*, Seattle, WA, pp. 331-338.

Cohen, P. R., McGee, D. R., Oviatt, S. L., Wu, L., Clow, J., King, R., et al. (1999). Multimodal interactions for 2D and 3D environments. *IEEE Computer Graphics and Applications* (July/Aug), pp. 10-13.

Cole, R. A., Mariani, J., Uszkoreit, H., Zaenen, A., & Zue, V. (Eds.). (1996). *Survey of the State of the Art in Human Language Technology*. Cambridge: National Science Foundation.

Dahlbäck, N., Jönsson, A., & Ahrenberg, L. (1993). Wizard Of Oz Studies -- Why and How. *Knowledge-Based Systems, 6* (4), pp. 258-266.

Dietz, P. H., & Leigh, D. L. (2001, Nov.). "DiamondTouch: A Multi-User Touch Technology" in the *Proceedings of the Symposium on User Interface Software and Technology (UIST'01)*, Orlando, FL, pp. 219-226.

Duda, R. O., & Hart, P. E. (1973). *Pattern Classification and Scene Analysis*. New York, NY: Wiley and Sons.

Feiner, S., MacIntyre, B., Hollerer, T., & Webster, A. (1997, October 13-14). "A Touring Machine: Prototyping 3D mobile augmented reality systems for exploring the urban environment" in the *Proceedings of the International Symposium on Wearable Computing*, Cambridge, MA, pp. 74-81.

Feiner, S., Macintyre, B., & Seligmann, D. (1993). Knowledge-based augmented reality. *Communications of the ACM, 36* (7), pp. 53-62.

Fitzmaurice, G. W., Ishii, H., & Buxton, W. A. S. (1995, May). "Bricks: laying the foundations for graspable user interfaces" in the *Proceedings of the Conference on Human Factors in Computing Systems*, Denver, Colorado, pp. 442-449.

Fukunaga, K. (1990). *Statistical Pattern Recognition* (2nd ed.). New York, NY: Academic Press.

Gibbs, W. (1998). The Re-Invention of Paper. *Scientific American* (Sept.).

Gibson, J. J. (1979). *The Ecological Approach to Visual Perception*: Houghton-Mifflin.

Gorman, P., Ash, J., Lavelle, M., Lyman, J., Delcambre, L., & Maier, D. (2000). Bundles in the Wild: Managing Information to Solve Problems and Maintain Situation Awareness. *Library Trends, 49* (2), pp. 266-289.

Gould, J. D. (1995). How to design usable systems (excerpt). In R. M. Baecker, J. Grudin, W. A. S. Buxton & S. Greenberg (Eds.), *Human-Computer Interaction: Toward the Year 2000* (San Francisco, CA: Morgan Kaufmann Publishers, Inc., pp. 93-121.

Grosz, B. J. (1977). *The representation and use of focus in dialogue understanding*. Unpublished Ph.D., University of California, Berkeley, Berkeley, CA.

Hahn, G. J., & Shapiro, S. S. (1994). *Statistical Models in Engineering*. New York, NY: Wiley and Sons.

Halang, W. A. (1999). Cognition oriented software engineering. In J. P. Marsh, B. Gorayska & J. L. Mey (Eds.), *Humane Interfaces: Questions of method and practice in cognitive technology* (Holland: Elsevier Science B.V.

Harper, R. R., Hughes, J. A., & Shapiro. D. Z. (1991). Harmonius Working and CSCW: Computer technology and air traffic control. In J. M. Bowers & S. D. Benford (Eds.), *Studies in Computer Supported Cooperative Work: Theory, Practice and Design* (Amsterdam, North-Holland: pp. 225-234.

Hawkes, N. (1971). *The Computer Revolution*. London, England: Thames and Hudson.

Heath, C., & Luff, P. (2000). *Technology in Action*. Cambridge, UK: Cambridge University Press.

Heisterkamp, P. (1993). "Ambiguity and uncertainty in spoken dialogue" in the *Proceedings of the EUROSPEECH'93 Conference*, Berlin, Germany, pp. 1657-1660.

Hughes, J. A., Randall, D., & Shapiro, D. (1992, Oct. 31-Nov. 4). "Faltering from ethnography to design" in the *Proceedings of the Conference on Computer-Supported Cooperative Work*, Toronto, Canada, pp. 115-122.

Ishii, H., & Ullmer, B. (1997, March). "Tangible bits: towards seamless interfaces between people, bits and atoms" in the *Proceedings of the Conference on Human Factors in Computing Systems*, Atlanta, GA, pp. 234-241.

Jini Network Technology. (2002). Retrieved June, 2002, from http://wwws.sun.com/software/jini/

Johnston, M. (1998, August). "Unification-based multimodal parsing" in the *Proceedings of the International Joint Conference of the Association for Computational Linguistics and the International Committee on Computational Linguistics*, Montreal, Canada, pp. 624-630.

Johnston, M., Cohen, P. R., McGee, D. R., Oviatt, S. L., Pittman, J. A., & Smith, I. (1997, March). "Unification-based multimodal integration" in the *Proceedings of the Meeting of the Association for Computational Linguistics*, Madrid, Spain, pp. 281-288.

Kaiser, E. C., & Cohen, P. R. (2002). "Implementation testing of a hybrid symbolic/statistical multimodal architecture" in the *Proceedings of the International Conference on Spoken Language Processing (ICSLP 2002)*, Denver, CO.

Kaptelinin, V. (1996). Computer-Mediated Activity. In B. A. Nardi (Ed.), *Context and Consciousness: Activity Theory and Human-Computer Communication* (Cambridge, MA: MIT Press.

Klemmer, S. R., Newman, M. W., Farrell, R., Bilezikjian, M., & Landay, J. A. (2001, Nov. 11-14). "The Designer's Outpost: A tangible interface for collaborative web site design" in the *Proceedings of the Symposium on User Interface Software and Technology (UIST'01)*, Orlando, FL, pp. 1-10.

Koons, D. B., Sparrell, C. J., & Thorisson, K. R. (1993). Integrating simultaneous input from speech, gaze, and hand gestures. In M. T. Maybury (Ed.), *Intelligent Multimedia Interfaces* (Cambridge, MA: AAAI Press/MIT Press, pp. 257-276.

Krueger, M. W. (1991). *Artificial Reality II*. Reading, MA: Addison-Wesley.

Kumar, S., Cohen, P. R., & Levesque, H. J. (2000, July 7-12). "The Adaptive Agent Architecture: Achieving Fault-Tolerance Using Persistent Broker Teams" in the *Proceedings of the International Conference on Multi-Agent Systems*, Boston, MA.

Kumar, S., Huber, M. J., Cohen, P. R., & McGee, D. R. (2002). Towards a formalism for conversation protocols using joint intention theory. *Computational Intelligence, 18* (2), pp. 174-228.

Kumar, S., Huber, M. J., McGee, D. R., Cohen, P. R., & Levesque, H. J. (2000, July 30-August 3). "Semantics of agent communication languages for group interaction" in the *Proceedings of the National Conference on Artificial Intelligence*, Austin, Texas, pp. 42-47.

Larsson, S., & Traum, D. (2000). Information state and dialogue management in the TRINDI Dialogue Move Engine Toolkit. *Natural Language Engineering, 6* (3-4), pp. 323-340.

Lavie, A., Levin, L., Qu, Y., Waibel, A., & Gates, D. (1996). "Dialogue processing in a conversational speech translation system" in the *Proceedings of the International Conference on Spoken Language Processing (ICSLP'96)*, pp. 554-557.

Lefebvre, P., Duncan, G., & Poirier, F. (1993). "Speaking with computers: A multimodal approach" in the *Proceedings of the EUROSPEECH'93 Conference*, Berlin, Germany, pp. 1665-1668.

Mackay, W. E. (1999). Is paper safer? The role of flight strips in air traffic control. *ACM Transactions on Computer-Human Interaction, 6* (4), pp. 311-340.

Mackay, W. E., Fayard, A.-L., Frobert, L., & Médini, L. (1998, April 18-23). "Reinventing the familiar: Exploring an augmented reality design space for air traffic control" in the *Proceedings of the Conference on Human Factors in Computing Systems*, Los Angeles, CA, pp. 558-565.

Mackay, W. E., Velay, G., Carter, K., Ma, C., & Pagani, D. (1993). Augmenting reality: adding computational dimensions to paper. *Communications of the ACM, 36* (7), pp. 96-97.

Mankoff, J., Hudson, S. E., & Abowd, G. D. (2000, Apr. 1-6). "Providing integrated toolkit-level support for ambiguity in recognition-based interfaces" in the *Proceedings of the Conference on Human Factors in Computing Systems (CHI'00)*, The Hague, Netherlands, pp. 368-375.

Mann, C. C. (2002). Homeland Insecurity. *The Atlantic Monthly, Sept.*, pp. 81-102.

Martin, D. A. (1995). Interactive display system, *United States Patent Office*. United States: Smart Technologies Inc.

Martin, D. L., Cheyer, A. J., & Moran, D. B. (1999). The Open Agent Architecture: A framework for building distributed software systems. *Applied Artificial Intelligence, 13* (January-March), pp. 91-128.

May, R. A., Thomas, J. J., Lewis, R. R., & Decker, S. D. (1998, April 23-26). "Physical human information interface workspace" in the *Proceedings of the Western Computer Graphics Symposium*, pp. 25-31.

Maybury, M. T. (Ed.). (1993). *Intelligent Multimedia Interfaces* AAAI/MIT Press.

McGee, D. R., & Cohen, P. R. (2001, Jan. 14-17). "Creating tangible interfaces by transforming physical objects with multimodal language" in the *Proceedings of the International Conference on Intelligent User Interfaces*, Santa Fe, NM, pp. 113-119.

McGee, D. R., Cohen, P. R., & Oviatt, S. L. (1998, August). "Confirmation in multimodal systems" in the *Proceedings of the International Joint Conference of the Association for Computational Linguistics and the International Committee on Computational Linguistics*, Montreal, Quebec, Canada, pp. 823-829.

McGee, D. R., Cohen, P. R., Wesson, R. M., & Horman, S. (2002, Apr. 20-25). "Comparing paper and tangible, multimodal tools" in the *Proceedings of the Conference on Human Factors in Computing Systems (CHI'02)*, Minneapolis, MI, pp. 407-414.

McGee, D. R., Cohen, P. R., & Wu, L. (2000, April 12-14). "Something from nothing: Augmenting a paper-based work practice with multimodal interaction" in the *Proceedings of the Conference on Designing Augmented Reality Environments*, Helsingor, Denmark, pp. 71-80.

McGee, D. R., Cohen, P. R., & Wu, L. (Pending). Augmenting and not replacing paper-based work practice via multimodal interaction.

McGee, D. R., Pavel, M., Adami, A., Wang, G., & Cohen, P. R. (2001, Nov. 15-16). "A visual modality for the augmentation of paper" in the *Proceedings of the Workshop on Perceptive User Interfaces (PUI'01)*, Orlando, FL.

McGee, D. R., Pavel, M., & Cohen, P. R. (2001). Context shifts: Extending the meaning of physical objects with language. *Human-Computer Interaction, 16* (special issue on context-aware computing), pp. 351-362.

Moghaddam, B., & Pentland, A. (1997). Probabilistic visual learning for object representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 19*, pp. 696-710.

Moore, G. A. (1991). *Crossing the Chasm: Marketing and Selling High-Tech Goods to Mainstream Customers.* New York: Harper Business.

Moran, T. P., Saund, E., van Melle, W., Bryll, R., Gujar, A. U., Fishkin, K. P., et al. (1999, May 15-20). "The ins and outs of collaborative walls: Demonstrating the Collaborage concept" in the *Proceedings of the Conference on Human Factors in Computing Systems*, Pittsburgh, PA, pp. CHI'99 Extended Abstracts: 192-193.

Moran, T. P., Saund, E., van Melle, W., Gujar, A. U., Fishkin, K. P., & Harrison, B. L. (1999, Nov. 7-10). "Design and technology for Collaborage: Collaborative collages of information on physical walls" in the *Proceedings of the Symposium on User Interface Software and Technology*, Asheville, NC, pp. 197-206.

Morin, P., & Junqua, J. (1993). "Habitable interaction in goal-oriented multimodal dialogue systems" in the *Proceedings of the EUROSPEECH'93 Conference*, Berlin, Germany, pp. 1669-1672.

Neal, J. G., & Shapiro, S. C. (1991). Intelligent multi-media interface technology. In J. W. Sullivan & S. W. Tyler (Eds.), *Intelligent User Interfaces* (New York, New York: ACM Press, Frontier Series, Addison Wesley Publishing Co., pp. 45-68.

Negroponte, N. (1997). Surfaces and displays. *Wired* (January), p. 212.

Newman, W., & Wellner, P. (1992, May). "A desk supporting computer-based interaction with paper documents" in the *Proceedings of the Conference on Human Factors in Computing Systems*, Monterey, CA, pp. 587-592.

Nigay, L., & Coutaz, J. (1995, May 7-11). "A generic platform for addressing the multimodal challenge" in the *Proceedings of the Conference on Human Factors in Computing Systems*, Denver, CO, pp. 98-105.

Norman, D. A. (1988). *The Design of Everyday Things*. New York, NY: Currency/Doubleday.

Novick, D. G. (1988). *Control of mixed-initiative discourse through meta-locutionary acts: a computational model.* Unpublished Technical report, University of Oregon, Eugene, OR.

Oviatt, S. L. (1996). "Multimodal interfaces for dynamic interactive maps" in the *Proceedings of the Conference on Human Factors in Computing Systems*, pp. 95-102.

Oviatt, S. L. (1997). Multimodal interactive maps: Designing for human performance. *Human-Computer Interaction, 12* (special issue on "Multimodal interfaces"), pp. 93-129.

Oviatt, S. L. (1999). "Mutual disambiguation of recognition errors in a multimodal architecture" in the *Proceedings of the Conference on Human Factors in Computing Systems*, Pittsburgh, PA, pp. 576-583.

Oviatt, S. L., Cohen, P. R., Fong, M. W., & Frank, M. P. (1992). "A rapid semi-automatic simulation technique for investigating interactive speech and handwriting" in the *Proceedings of the International Conference on Spoken Language Processing (ICSLP'92)*, Alberta, Canada, pp. 1351-1354.

Oviatt, S. L., Cohen, P. R., & Wang, M. Q. (1994). Toward interface design for human language technology: Modality and structure as determinants of linguistic complexity. *Speech Communication, 15* (3-4), pp. 283-300.

Oviatt, S. L., Cohen, P. R., Wu, L., Vergo, J., Duncan, L., Suhm, B., et al. (2001). Designing the user interface for multimodal speech and gesture applications: State-of-the-art systems and research directions. In J. Carroll (Ed.), *Human-Computer Interaction in the New Millennium* (Vol. 15). Reading, MA: Addison-Wesley, pp. 263-322.

Oviatt, S. L., DeAngeli, A., & Kuhn, K. (1997, March). "Integration and synchronization of input modes during multimodal human-computer interaction" in the *Proceedings of the Conference on Human Factors in Computing Systems*, Atlanta, GA, pp. 415-422.

Parasuraman, R., & Mouloua, M. (Eds.). (1996). *Automation and Human Performance: Theory and Applications*. Mahwah, NJ: Lawrence Erlbaum Associates, Publishers.

Pavlovic, V., Sharma, R., & Huang, T. (1997). Visual interpretation of hand gestures for human-computer interaction: A review. *IEEE Transactions on PAMI, 7* (19), pp. 677-695.

Rekimoto, J. (1997, October). "Pick-and-drop: a direct manipulation technique for multiple computer environments" in the *Proceedings of the Symposium on User Interface Software and Technology*, Banff, Alberta, Canada, pp. 31-39.

Rekimoto, J. (1998, July 15-17). "Matrix: A realtime object identification and registration method for augmented reality" in the *Proceedings of the Asia-Pacific Computer and Human Interaction*, Kanagawa, Japan.

Rekimoto, J., & Nagao, K. (1995). "The world through the computer: computer augmented Interaction with real world environments" in the *Proceedings of the Symposium on User Interface Software and Technology*, pp. 29-36.

Rekimoto, J., & Saitoh, M. (1999, May). "Augmented surfaces: a spatially continuous work space for hybrid computing environments" in the *Proceedings of the Conference on Human Factors in Computing Systems*, Pittsburgh, PA, pp. 378-385.

Rønby Pedersen, E., Sokoler, T., & Nelson, L. (2000). "PaperButtons: Expanding a tangible user interface" in the *Proceedings of the Designing Interactive Systems (DIS'00)*, Brooklyn, NY, pp. 216-223.

Rudnicky, A. I., & Hauptmann, A. G. (1992). Chapter 10: Multimodal interaction in speech systems. In M. M. Blattner & R. B. Dannenberg (Eds.), *Multimedia Interface Design* (New York, NY: ACM Press, pp. 147-171.

Searle, J. (1989). How Performatives Work. *Linguistics and Philosophy, 12*, pp. 535-558.

Searle, J. R. (1969). *Speech acts: An essay in the philosophy of language*. Cambridge: Cambridge University Press.

Sharma, R., Pavlovic´, V. I., & Huang, T. S. (1998). Toward multimodal human-computer interface. *Procedings of the IEEE, (Special Issue Multimedia Signal Processing), 86* (5), pp. 853-869.

Smith, R. W. (1997). "Practical issues in mixed-initiative natural language dialog: An experimental perspective" in the *Proceedings of the AAAI Spring Symposium on Computational Models for Mixed Initiative Interaction*, pp. 158-162.

Smith, R. W., Hipp, D. R., & Biermann, A. W. (1995). An architecture for voice dialog systems based on Prolog-style theorem proving. *Computational Linguistics, 21* (3), pp. 281-320.

Streitz, N. A., Geibler, J., & Holmer, T. (1998, February). "Roomware for cooperative buildings: integrated design of architectural spaces and information spaces" in the *Proceedings of the International Workshop on Cooperative Buildings: Integrating Information, Organization, and Architecture*, Darmstadt, Germany, pp. 4-21.

Tufte, E. R. communicated to D. R. McGee, June, 2001.

Ullmer, B. (2001). Emerging frameworks for tangible user interfaces. In J. Carroll (Ed.), *Human-Computer Interaction in the New Millennium* (Vol. 15). Reading, MA: Addison-Wesley.

Ullmer, B., & Ishii, H. (1997, October). "The metaDESK: models and prototypes for tangible user Interfaces" in the *Proceedings of the Symposium on User Interface Software and Technology*, Banff, Alberta, Canada, pp. 223-232.

Ullmer, B., & Ishii, H. (1998, July). "mediaBlocks: physical containers, transports, and controls for online media" in the *Proceedings of the Conference on Computer Graphics and Interactive Techniques*, Orlando, FL, pp. 379-386.

Underkoffler, J., & Ishii, H. (1999, May). "Urp: a luminous-tangible workbench for urban planning and design" in the *Proceedings of the Conference on Human Factors in Computing Systems*, Pittsburgh, PA, pp. 386-393.

Vo, M. T., & Wood, C. (1996). "Building an application framework for speech and pen input integration in multimodal learning interfaces" in the *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, Atlanta, GA.

Walker, M., & Whittaker, S. (1990). "Mixed initiative in dialogue: An investigation into discourse segmentation" in the *Proceedings of the 28th Meeting of the Association for Computational Linguistics (ACL'90)*, pp. 70-78.

Want, R., Fishkin, K. P., Gujar, A., & Harrison, B. L. (1999, May). "Bridging physical and virtual worlds with electronic tags" in the *Proceedings of the Conference on Human Factors in Computing Systems*, Pittsburgh, PA, pp. 370-377.

Want, R., & Russell, D. M. (2000, Sept. 15). *Ubiquitous Electronic Tagging*. Retrieved 2000, Dec. 4, from http://www.computer.org/dsonline/articles/ds2wan.htm

Weiser, M. (1993). Some computer science issues in ubiquitous computing. *Communications of the ACM, 36* (7), pp. 75-84.

Wellner, P. (1993). Interacting with paper on the DigitalDesk. *Communications of the ACM, 36* (7), pp. 87-96.

Wittenburg, K. (1993, June 22-26). "F-PATR: Functional constraints for unification grammars" in the *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, Ohio State University, pp. 216-223.

Woods, D. D., Cook, R. I., & Billings, C. E. (1995). The impact of technology on physician cognition and performance. *Journal of Clinical Monitoring, 11*, pp. 5-8.

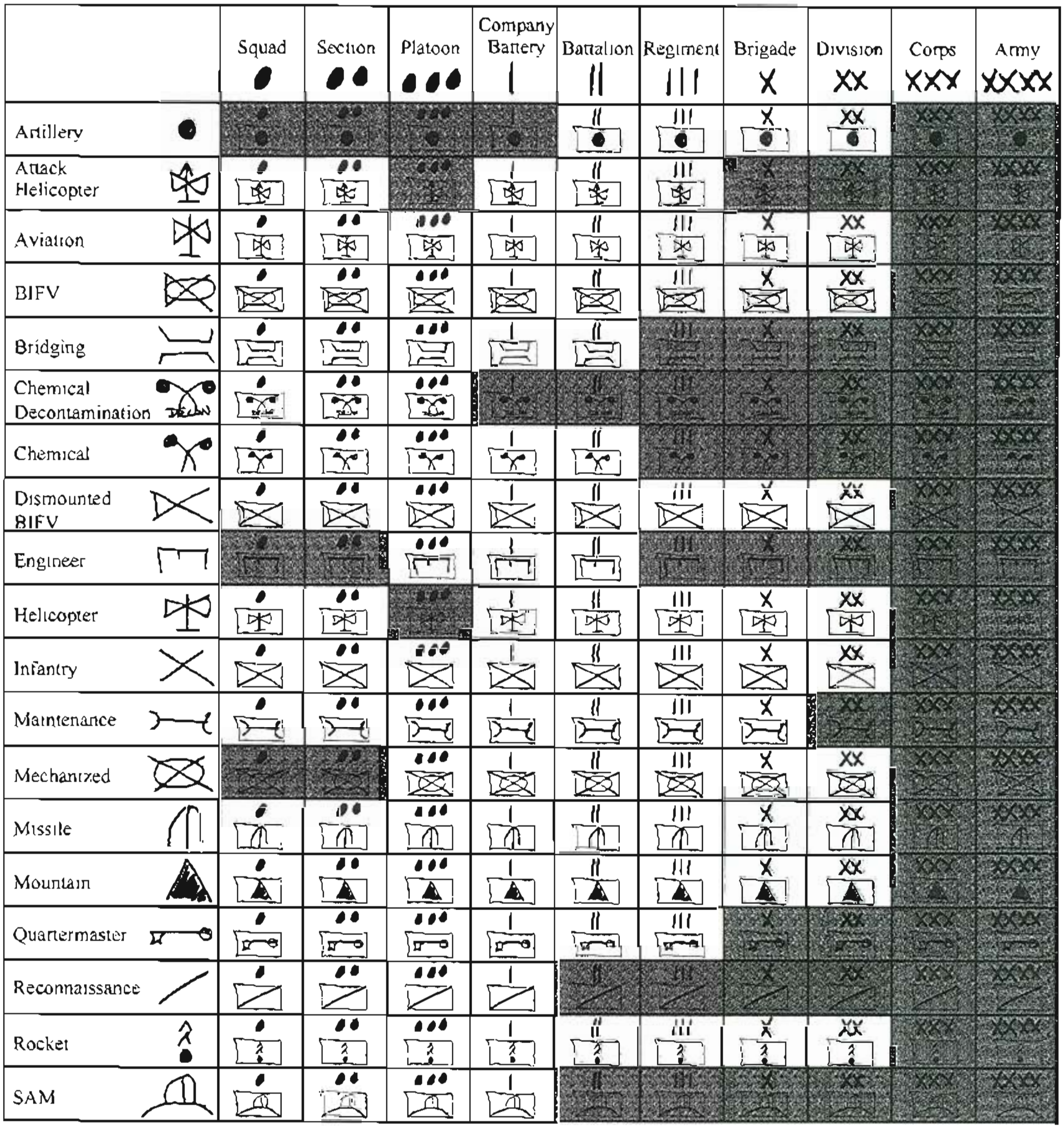
Wu, L., Oviatt, S. L., & Cohen, P. R. (1999a). Multimodal integration - A statistical view. *IEEE Transactions on Multimedia, 1* (4), pp. 334-341.

Wu, L., Oviatt, S. L., & Cohen, P. R. (1999b). Statistical multimodal integration for intelligent HCI. In J. L. Y.H Hu, E. Wilson & S. Douglas (Ed.), *Neural Networks for Signal Processing IX* (pp. 487-496.

Wu, L., Oviatt, S. L., & Cohen, P. R. (2002). From members to teams to committee -- a robust recognition approach to gestural and multimodal signal processing. *IEEE Transactions on Neural Networks, 14* (4: Special issue on Intelligent Multimedia), p. in press.

Appendix A. Symbology library

There are two elements to the table below. Unit *types* along the rows and unit *sizes* (echelons) in the columns. Each element in the table is a unit symbol. Symbols that are recognized but are not valid in the domain appear in gray (e.g., there are no such things as artillery squads or attack helicopter armies).



Appendix B. Speech Grammar

The semantics of the grammar below is EBNF. Phrases in the grammar typically take the form of non-terminal = series of terminals or nonterminals. Nonterminals are contained in angle brackets and are in lower case for readability (e.g., <non_terminal>). Terminals are all in upper case (e.g. UNIT). To enable scoping, the first non-terminal specification is in square brackets (e.g. [<Start>]). Within non-terminal specifications themselves, optional elements are preceded by square brackets. For example, in the phrase <salt_request>=MAY I HAVE THE SALT [OPT] PLEASE, the PLEASE can be either present or absent.

```
//RASA EXPERIMENTAL SPEECH GRAMMAR
//

[<Start>]
<Start>=[OPT] <suspected_confirmed> <unit_creation>
<Start>=[OPT] <suspected_confirmed> <point_creation>
<Start>=[OPT] <suspected_confirmed> <line_creation>
<Start>=[OPT] <suspected_confirmed> <area_creation>
<Start>=<unit_designator>
<Start>=<unit_strength>
<Start>=<register_map>
<Start>=<register_artifact>
<Start>=<place_button>
<Start>=<confirmation>
<Start>=<enable_rasa>
<Start>=<unit_removal>
<Start>=<suspected_confirmed>
<Start>=<new_scenario>
//<Start>=<unit_designator> FOLLOW THIS ROUTE
<Start>=<projection>

///////////////////////////////////////////////////////////////////////////////////////
       /
// unit_designator
///////////////////////////////////////////////////////////////////////////////////////
       /
[<unit_designator>]
<unit_designator>=<enemy> [OPT] <force>
<unit_designator>=<friendly> [OPT] <force>
<unit_designator>=<enemy> <echelon>
<unit_designator>=<friendly> <echelon>
<unit_designator>=<enemy_designator>
<unit_designator>=<friendly_designator>
<unit_designator>=[OPT] THIS <unit_creation>
<unit_designator>=<object_deictic> [OPT] UNIT

<force>=UNIT
<force>=FORCE
<force>=UNITS
<force>=FORCES
```

```
///////////////////////////////////////////////////////////////////////////////
       /
// unit_creation
///////////////////////////////////////////////////////////////////////////////
       /
[<unit_creation>]
<unit_creation>=[OPT] <enemy> <unit> [OPT] HERE
<unit_creation>=[OPT] <enemy> <unit> [OPT] <callword> <enemy_designator> [OPT]
       HERE
<unit_creation>=<enemy_designator> [OPT] HERE
<unit_creation>=[OPT] <friendly> <unit> [OPT] HERE
<unit_creation>=[OPT] <friendly> <unit> [OPT] <callword> <friendly_designator>
       [OPT] HERE
<unit_creation>=<cardinal> <unit> [OPT] HERE
<unit_creation>=<friendly_designator> [OPT] HERE

<unit>=<unit_type> <echelon>
///////////////////////////////////////////////////////////////////////////////
       /
// unit_removal
///////////////////////////////////////////////////////////////////////////////
       /
[<obj_removal>]
<obj_removal>=<delete>
<obj_removal>=<delete> <object_deictic>
<obj_removal>=<unit_removal>
<obj_removal>=<line_removal>
<obj_removal>=<area_removal>
<obj_removal>=<point_removal>

[<delete>]
<delete>=DELETE
<delete>=REMOVE
<delete>=WITHDRAW

[<unit_removal>]
<unit_removal>=<delete> <unit_designator>
[<line_removal>]
<line_removal>=<delete> [OPT] <object_deictic> LINE
[<area_removal>]
<area_removal>=<delete> [OPT] <object_deictic> AREA
[<point_removal>]
<point_removal>=<delete> [OPT] <object_deictic> POINT

[<enemy>]
<enemy> = RED
<enemy> = ENEMY

[<friendly>]
<friendly> = BLUE
<friendly> = FRIENDLY

// CALL SIGNS
[<callword>]
<callword>=WITH CALLSIGN
<callword>=CALLED
<callword>=CALLSIGN

<enemy_designator>=ADVANCED GUARD
<enemy_designator>=MAIN BODY
<enemy_designator>=REAR GUARD

<friendly_designator>=<cardinal>
```

```
<friendly_designator>=<letter>
<friendly_designator>=<letter> <digit>
<friendly_designator>=<letter> <digit> <digit>
<friendly_designator>=<letter> <digit> <digit> <digit>

// Cardinal Number Specifications
<cardinal>=<card>
<cardinal>=<teenth>
<card>=FIRST
<card>=SECOND
<card>=THIRD
<card>=FOURTH
<card>=FIFTH
<card>=SIXTH
<card>=SEVENTH
<card>=EIGHTH
<card>=NINTH
<teenth>=TENTH
<teenth>=ELEVENTH
<teenth>=TWELFTH
<teenth>=THIRTEENTH
<teenth>=FOURTEENTH
<teenth>=FIFTEENTH
<teenth>=SIXTEENTH
<teenth>=SEVENTEENTH
<teenth>=EIGHTEENTH
<teenth>=NINETEENTH

<tens>=TWENTY
<tens>=THIRTY
<tens>=FORTY
<tens>=FIFTY
<tens>=SIXTY
<tens>=SEVENTY
<tens>=EIGHTY
<tens>=NINETY
<teens>=TEN
<teens>=ELEVEN
<teens>=TWELVE
<teens>=THIRTEEN
<teens>=FOURTEEN
<teens>=FIFTEEN
<teens>=SIXTEEN
<teens>=SEVENTEEN
<teens>=EIGHTEEN
<teens>=NINETEEN

<one_to_hundred>=<num>
<one_to_hundred>=<teens>
<one_to_hundred>=<tens>
<one_to_hundred>=<tens> <num>
<one_to_hundred>=ONE HUNDRED
<one_to_hundred>=A HUNDRED

[<num>]
<num>=ONE
<num>=<numAbove1>

<numAbove1>=TWO
<numAbove1>=THREE
<numAbove1>=FOUR
<numAbove1>=FIVE
<numAbove1>=SIX
```

```
<numAbove1>=SEVEN
<numAbove1>=EIGHT
<numAbove1>=NINE

[<unit_type>]
<unit_type>=[OPT] HEAVY <armor>
<unit_type>=<mechanized>
<unit_type>=<anti_armor>
<unit_type>=INFANTRY
<unit_type>=<arty>
<unit_type>=[OPT] ARMORED <recon>
<unit_type>= [OPT] ATTACK <helicopter>
<unit_type>=[OPT] COMBAT <engineer>
<unit_type>=<ace>
<unit_type>=MAINTENANCE
<unit_type>=[OPT] MOTORIZED MEDICAL
<unit_type>=MILITARY INTELLIGENCE
<unit_type>=M I
<unit_type>=COUNTER INTELLIGENCE
<unit_type>=SUPPORT
<unit_type>=<ada>
<unit_type>=<cav>
<unit_type>=AVIATION

<anti_armor>=ANTI <armor>
<anti_armor>=A T
<armor>=ARMOR
<armor>=ARMORED
<armor>=TANK
<armor>=M1A1
<armor>=T72
<engineer>=ENGINEER
<engineer>=ENGINEERING
<helicopter>=helicopter
<helicopter>=HELO
<mechanized>=MECHANIZED
<mechanized>=MECH
<mechanized>=BMP
<recon>=REC
<recon>=RECON
<recon>=RECONNAISSANCE
<recon>=SCOUT
<recon>=BRDM
<recon>=LAR
<cav>=CAVALRY
<cav>=CAV
<ada>=ADA
<ada>=AIR DEFENSE ARTILLERY
<ada>=ZSU
<ace>=ACE
<ace>=ANALYSIS AND CONTROL ELEMENT

[<arty>]
<arty>=<gun>
<arty>=2S1 [OPT] <gun>
<arty>=<onetwotwo> <millimeter>  <gun>
<arty>=2S3 [OPT] <gun>
<arty>=<onefivetwo> <millimeter>  <gun>
<arty>=MLRS
<arty>=<onefivefive> <millimeter>  <gun>
<arty>=SA9

[<gun>]
```

```
<gun>=ARTILLERY
<gun>=HOWITZER

<onefivefive>=ONE HUNDRED AND FIFTY FIVE
<onefivefive>=ONE FIVE FIVE
<onefivetwo>=ONE HUNDRED AND FIFTY TWO
<onefivetwo>=ONE FIVE TWO
<onetwotwo>=ONE HUNDRED AND TWENTY TWO
<onetwotwo>=ONE TWO TWO
//<sp>=S P
//<sp>=SELF PROPELLED

[<millimeter>]
<millimeter>=M M
<millimeter>=MILLIMETER

//////////////////////////////////////////////////////////////////////////////
       /
// BEGIN point_creation COMMAND RULE
//////////////////////////////////////////////////////////////////////////////
       /
[<point_creation>]
<point_creation>=<point_designator>

[<point_designator>]
<point_designator>=OBJECTIVE <letter>
<point_designator>=COMMAND POST [OPT] <num>
<point_designator>=COORDINATION POINT
<point_designator>=CONTACT POINT [OPT] <num>
<point_designator>=<trp> [OPT] <num>

<trp>=TRP
<trp>=TARGET REFERENCE POINT

//////////////////////////////////////////////////////////////////////////////
       /
// BEGIN line_creation COMMAND RULE
//////////////////////////////////////////////////////////////////////////////
       /
[<line_creation>]
<line_creation>=PHASE LINE <color>
<line_creation>=BOUNDARY [OPT] LINE
<line_creation>=BARBED WIRE
<line_creation>=WIRE
<line_creation>=FRONT
//<line_creation>=FEBA
//<line_creation>=FLOT
<line_creation>=FORWARD LINE OF OWN TROOPS
<line_creation>=FORTIFICATION
<line_creation>=FORTIFIED LINE
//<line_creation>=[OPT] EARTHEN BERM
<line_creation>=DITCH
<line_creation>=DEFENSIVE POSITION
<line_creation>=ANTI TANK DITCH
<line_creation>=LINE OF DEPARTURE
<line_creation>=L O D
<line_creation>=<echelon> BOUNDARY
<line_creation>=SUPPLY ROUTE [OPT] <letter>
//<line_creation>=MAIN SUPPLY ROUTE
//<line_creation>=MSR

//////////////////////////////////////////////////////////////////////////////
       /
```

```
// BEGIN area_creation COMMAND RULE
///////////////////////////////////////////////////////////////////////////////
       /
[<area_creation>]
<area_creation>=LANDING ZONE [OPT] <letter>
<area_creation>=L Z [OPT] <letter>
<area_creation>=DROP ZONE [OPT] <letter>
<area_creation>=ANTI PERSONNEL MINEFIELD
<area_creation>=A P MINEFIELD
<area_creation>=ANTI TANK MINEFIELD
<area_creation>=A T MINEFIELD
<area_creation>=MINEFIELD
<area_creation>=MINES
<area_creation>=LAND MINES
<area_creation>=<engagement_area> [OPT] <letter>
<area_creation>=<assembly_area> [OPT] <letter>
<area_creation>=AREA OF OPERATIONS
<area_creation>=OPERATIONS AREA
<area_creation>=A O
<area_creation>=AREA OF INTEREST
<area_creation>=A I
<area_creation>=NO GO AREA
<area_creation>=NO GO
<area_creation>=SLOW GO AREA
<area_creation>=SLOW GO
<area_creation>=NO FIRE AREA
<area_creation>=N F A
<area_creation>=RESTRICTED FIRE AREA
<area_creation>=R F A
<area_creation>=SWAMP
<area_creation>=<echelon> <battle_position> [OPT] <letter> [OPT] <num>

<engagement_area>=AREA OF ENGAGEMENT
<engagement_area>=A E
<engagement_area>=ENGAGEMENT AREA

<assembly_area>=AREA OF ASSEMBLY
<assembly_area>=ASSEMBLY AREA
<assembly_area>=A A

<battle_position>=BP
<battle_position>=BATTLE POSITION

[<letter>]
<letter>=ALPHA
<letter>=BRAVO
<letter>=CHARLIE
<letter>=DELTA
<letter>=ECHO
<letter>=FOXTROT
<letter>=GOLF
<letter>=HOTEL
<letter>=INDIA
<letter>=JOLIET
<letter>=KILO
<letter>=LIMA
<letter>=MIKE
<letter>=NOVEMBER
<letter>=OSCAR
<letter>=PAPA
<letter>=QUEBEC
<letter>=ROMEO
<letter>=SIERRA
```

```
<letter>=TANGO
<letter>=UNIFORM
<letter>=VICTOR
<letter>=WHISKEY
<letter>=X RAY
<letter>=YANKEE
<letter>=ZULU


[<digit>]
<digit>=OH
<digit>=<num>
<digit>=NINER

[<echelon>]
<echelon>=SQUAD
<echelon>=SECTION
<echelon>=PLATOON
<echelon>=COMPANY
<echelon>=BATTERY
<echelon>=BATTALION

///////////////////////////////////////////////////////////////////////////////
       /
// deixis
///////////////////////////////////////////////////////////////////////////////
       /
[<deixis>]
<deixis>=<spatial_deictic>
<deixis>=<coordinate>
<deixis>=<object_deictic> <echelon>
<deixis>=<unit_designator>

///////////////////////////////////////////////////////////////////////////////
       /
// object_deictic
///////////////////////////////////////////////////////////////////////////////
       /
[<object_deictic>]
<object_deictic>=THIS
<object_deictic>=THAT

///////////////////////////////////////////////////////////////////////////////
       /
// spatial_deictic
///////////////////////////////////////////////////////////////////////////////
       /
[<spatial_deictic>]
<spatial_deictic>=HERE
<spatial_deictic>=THERE
<spatial_deictic>=THIS POSITION


///////////////////////////////////////////////////////////////////////////////
       /
// BEGIN color COMMAND RULE
///////////////////////////////////////////////////////////////////////////////
       /
[<color>]
<color>=YELLOW
<color>=BLACK
<color>=GREEN
<color>=RED
```

```
<color>=BLUE
//<color>=PURPLE
//<color>=ORANGE
//<color>=WHITE
//<color>=GRAY

[<enable>]
<enable>=ENABLE
<enable>=SHOW [OPT] ME
<enable>=DISPLAY
<enable>=TURN ON

[<disable>]
<disable>=DISABLE
<disable>=TURN OFF

////////////////////////////////////////////////////////////////////////////////
       /
// enable_rasa
////////////////////////////////////////////////////////////////////////////////
       /
[<enable_rasa>]
<enable_rasa>=<enable> [OPT] RASA
<enable_rasa>=<disable> [OPT] RASA

////////////////////////////////////////////////////////////////////////////////
       /
// projections
////////////////////////////////////////////////////////////////////////////////
       /
[<projection>]
<projection>=<enable> [OPT] THE PROJECTIONS
<projection>=<disable> PROJECTIONS
<projection>=TURN [OPT] THE PROJECTIONS OFF
<projection>=TURN [OPT] THE PROJECTIONS ON

////////////////////////////////////////////////////////////////////////////////
       /
// suspected_confirmed
////////////////////////////////////////////////////////////////////////////////
       /
[<suspected_confirmed>]
<suspected_confirmed>=SUSPECTED
<suspected_confirmed>=PROPOSED
<suspected_confirmed>=CONFIRMED

////////////////////////////////////////////////////////////////////////////////
       /
// new_scenario
////////////////////////////////////////////////////////////////////////////////
       /
[<new_scenario>]
<new_scenario>=NEW SCENARIO

////////////////////////////////////////////////////////////////////////////////
       /
// display_imperative
////////////////////////////////////////////////////////////////////////////////
       /
[<display_imperative>]
//<display_imperative>=GRACE <system_name> [OPT] PLEASE

//<system_name>=COMMANDVU
```

```
//<system_name>=MODSAF
//<system_name>=P V D

////////////////////////////////////////////////////////////////////////////////
       /
// BEGIN registration COMMAND RULEs
////////////////////////////////////////////////////////////////////////////////
       /
[<place_button>]
<place_button>= [OPT] <place> [OPT] THE <button_type> BUTTON [OPT] HERE
<place>=PUT
<place>=PLACE
<button_type>=CONFIRM
<button_type>=CONFIRMATION
<button_type>=CANCEL

[<register_artifact>]
<register_artifact>=[OPT] REGISTER ARTIFACT
<register_artifact>=[OPT] REGISTER POSTIT

[<register_map>]
<register_map>=[OPT] REGISTER MAP [OPT] AT <coordinate>

[<coordinate>]
<coordinate>=<utm>
<utm>=[OPT] <location_prefix> <grid>
<location_prefix>=GRID <location_noun>
<location_noun>=LOCATION
<location_noun>=POSITION
<location_noun>=COORDINATES
<location_noun>=ENEMY POSITION
<grid>=<digit> <digit> <digit> <digit>
<grid>=<digit> <digit> <digit> <digit> <digit> <digit>
<grid>=<digit> <digit> <digit> <digit> <digit> <digit> <digit> <digit>

////////////////////////////////////////////////////////////////////////////////
       /
// BEGIN confirmation COMMAND RULE
////////////////////////////////////////////////////////////////////////////////
       /
[<confirmation>]
<confirmation>=<confirm>
<confirmation>=<disconfirm>
<confirmation>=<set_confirm>
<confirm>=CONFIRM [OPT] THAT
<confirm>=AFFIRMATIVE
<confirm>=YES
<confirm>=DO IT
<confirm>=OKAY
<confirm>=ACCEPT [OPT] THAT
<confirm>=ROGER [OPT] THAT
<disconfirm>=NEGATIVE
<disconfirm>=NEGATORY
<disconfirm>=NO
<disconfirm>=CANCEL [OPT] THAT
<disconfirm>=REJECT [OPT] THAT
<set_confirm>=MAKE CONFIRMATIONS EXPLICIT
<set_confirm>=MAKE CONFIRMATIONS IMPLICIT

////////////////////////////////////////////////////////////////////////////////
       /
// unit_strength
```

```
////////////////////////////////////////////////////////////////////////////////////
       /
[<unit_strength>]
<unit_strength>=[OPT] <strength> <str_amt>
<unit_strength>=<str_amt> [OPT] <strength>
<unit_strength>=DEAD

<strength>=STRENGTH
<strength>=CAPABILITY

<str_amt>=[OPT] AT <one_to_hundred> [OPT] PERCENT
```

Appendix C. Instructions

Rasa interprets spoken and written commands that occur simultaneously or independently on paper. With it you can draw unit symbology on a post-it note, while at the same time speaking things not on the post-it. After drawing the symbol on the post-it, you simply place it on the map.

What can you do?

With Rasa you can place and move enemy and friendly units. Each time you place a unit on the map a confirmation will appear. The unit remains local to your map and will not be entered into a remote database until it has been confirmed. You can confirm directly by saying 'ROGER' or 'AFFIRMATIVE' or by using the CONFIRM and CANCEL buttons. Units are also confirmed, if you proceed to the next command. Units that have not been confirmed cannot be moved or otherwise updated. To move a unit you will point at the post-it, pick it up, and move it to its new location. You must pause between pointing at the unit that is being moved, and placing it in its new location. Finally, units can be removed, for instance if they have withdrawn, by pointing at the unit and saying DELETE. You can then throw the post-it in the trash.

What can you say?

While drawing units or after placing them on the map, you can specify whether they are friendly or enemy. You can also specify their callsign. For example, while drawing a unit, you can say the following:

- *ENEMY*
- *FIRST* (which also indicates that the unit is friendly)
- *FRIENDLY UNIT CALLED LIMA*

The system will respond with a confirmation once the unit has been placed on the map. A visual depiction of the system's understanding of the unit you have placed will be projected onto the map. An audible computer-generated phrase will also be spoken. For example, if you draw a mechanized company and say '*FRIENDLY UNIT CALLED LIMA,*' the system will say 'FRIENDLY MECHANIZED CALLED LIMA HAS BEEN

SIGHTED AT,' the grid coordinates of the location you placed the unit. Simultaneously, Rasa will project a rendering of the unit on the paper map.

What can you draw?

Today Rasa has a limited understanding of drawn unit symbols. Only rectangular symbols are supported. Unit symbols are basic and when drawn the system will not understand attributions, such as labels, callsigns, etc. The new diamond standard for enemy units will be available in a future release. Symbols are limited to basic symbology for the following unit types at all relevant echelons:

- Artillery
- Attack helicopter
- Aviation
- Engineer
- Helicopter
- Infantry
- Maintenance
- Mechanized
- Reconnaissance (LAR is not supported)

Again, draw only the basic unit symbol. If you wish to indicate that, for example, you are placing the FIRST recon platoon, draw a basic recon platoon and say FIRST. Or put the draw the symbol, put the unit on the map, confirm it, then point at it and say first.

When you draw on either a post-it note or point at the map, the system will respond with a scratching sound. If you don't hear the scratch, the map is not receiving your input. Try pressing harder.

When placing units on the map, be sure that the post-it notes DO NOT overlap. Currently Rasa only functions if it can distinguish between the units with a slight separation.

Mistakes

Because Rasa is trying to understand human language, sometimes it makes mistakes. This is why we have both visual and audible confirmations of Rasa's understanding prior to submitting the results to the remote database. If Rasa misunderstands you, you can try again. If it doesn't understand your unit drawing, it may not respond at all. In this case, you can go ahead and put the unit on the map, then use speech to describe the symbol.

If the system fails utterly, the experimenter will inform you that it is currently down. Continue your map updates until the system is once again online, you will then be asked to update the digital information by pointing at each of the new units and describing the symbol, and moving units as described earlier. In the case of these moves, you will point at the projection of the old location and then at the current location of the post-it.

What else?

With Rasa, planning and tactical symbology can be added to the map's overlay and automatically captured electronically, such as boundary lines, axes of advance, phase lines, objectives. Cultural features can be added as well, e.g. churches, mosques, historical sites, etc.

However, these features of Rasa will not be exploited in this experiment. Feel free to add these symbols to the map, then cancel any erroneous command that Rasa generates. For example, if you add a NO FIRE AREA (NFA) to the map and Rasa recognizes that as a unit symbol, simply point at the confirmation and say cancel.

Appendix D. Scenario

Rasa is an advanced concept prototype that allows an officer to update digital systems with tactical and planning information, such as enemy unit position by using paper tools (maps and post-its) that are common to the task today.

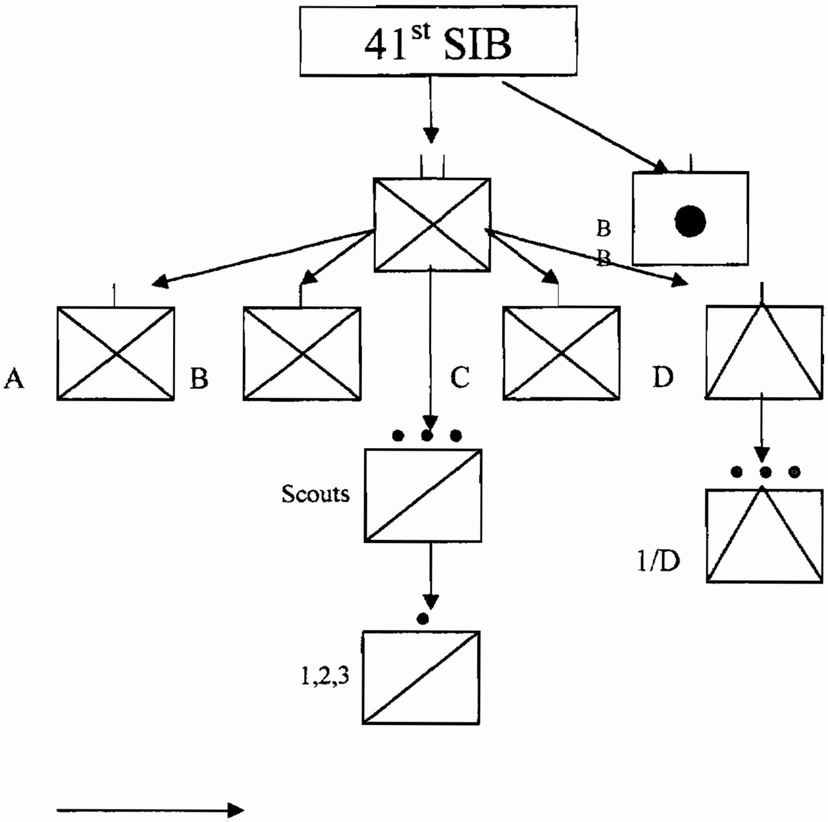
When standard unit symbology is drawn on a Post-it note and subsequently placed on a paper map, Rasa adds the unit to the digital system. Rasa will understand that you have indicated a unit movement, when you move a Post-it note. To modify or query particular attributes of each unit, such as battle damage assessment (BDA) that may not be visible on the unit's symbology, you can point at the Post-it notes and speak to Rasa.

In the next 10 minutes, you will be instructed in how to use Rasa, what unit symbols and spoken language it understands.

During the experiment, you will act as map plotter. Your job, limited by this experiment, is to ensure that the common tactical picture (CTP) is complete and up-to-date. We will simulate a scenario that you will examine for 5 minutes before the simulation begins. When it begins, you will us Rasa to track unit positions for both enemy and friendly units.

Scenario Description

During this experiment, you will to act as the operations officer of the 41st Separate Infantry Brigade. You're force breakdown is described in the following diagram. Close air support (CAS) is unavailable at this time.



The experiment will be a simulation of artificial events that take place on or near the Peoples Republic of Orange (a.k.a. Camp Pendleton). The simulation will not proceed in real time. As time moves on the simulated clock, you will receive reports relevant for that time period in the simulation. The reports and mission prior to the beginning of the simulation are described in the following table.

Activity Report

Time	Report	Mission
D-10	Rebel paramilitaries become active in the Peoples Republic of Orange. Approximately 100 U.S. citizens are present in the city of Combat Town (Objective A). U.S. Embassy for the Peoples Republic of Orange is located in Combat Town. CNN reports riots in both Alisonville (Objective B, a.k.a. R131 MOUT) and Combat Town. Terrorist activities have begun.	ARG/MEU An amphibious ready group and your MEU are ordered off the coast of Orange beyond line of sight (BLOS). A carrier battle group (CVBG) is ordered to provide support. You are to be prepared (B/P) to conduct NEO operations and secure the embassy.

Master Scenario Events List

Wall Time	Sim. Time	Report	
0	D-7	You arrive off the coast of Green as ordered. Enemy infantry platoon sighted in the vicinity of Alisonville, grid 6485.	*Training* 1 update
0	D-3	USAF pilot down vicinity of Alisonville. Enemy infantry platoon sighted in the vicinity of Combat Town, grid 681878. Enemy force present north of AI.	*Training* 1 update
0	D-2	Insert recon squads 1-3/1-162 IN at NAIs, grids 502958, 608863, 667894.	*Training* 3 updates
0	D-1	Pilot sighted by recon Squad 2/1-162 IN vicinity of Alisonville, grid 623846.	*Training* (1 create)
0	D-Day	***Mission:*** Your commander intends to evacuate U.S. citizens from both Obj. A & B, and secure the embassy. A/1-162 IN has his main effort. They will make landing at Blue Beach and proceed immediately to seize Obj. A, secure the embassy and all AmCits in vicinity of Obj. A. 1/D/1-162 IN platoon in support of the main effort will precede it and provide a screening action along phase line blue, fixing the enemy force in the vic. of Obj. A. B/1-162 IN is a supporting effort. Prior to the main effort, it will conduct an Air Assault raid on Objective B, fixing and destroying the enemy there, and evacuating the downed pilot and any U.S. non-combatants. C/1-162 IN is in reserve. D/1-162 IN(-) is in support of the main effort.	N/A
0	0200	1/D/1-162 IN platoon ashore by LCAC, grid 617768.	*Training* 1 create
0	0215	1/D/1-162 IN platoon reports minefield at landing grids 623773 to 631763.	N/A
0	0300	1/D/1-162 IN platoon reaches phase line blue, grid 6886, begins screen action.	*Training* 1 move

0	0330	B/1-162 IN begins TRAP execution.	N/A
	0530	B/1-162 IN disembarked securing zone, grid 6284. E L-hour, grid 612770.	2 creates
	0545	D/1-162 IN(-) completes mine penetration and has advanced to 654810.	1 move
	0547	A/1-162 IN L-hour, grid 624756.	1 create
	0550	B/1-162 IN has engaged the enemy at 630868.	1 move
	0635	1/1-162 IN recon squad reports enemy mechanized battalion approaching from the west, advancing along improved surface road, grid 488943.	1 create
		FRAGO: Your commander has just issued a FRAGO. His intent is to prevent the eastbound forces from reinforcing, insuring sufficient time for AmCit/pilot evacuation and withdrawal. C/1-162 IN, your reserve infantry company, will prevent northwest encroachment, moving to a defensive position south of Jardine Canyon. 1/1-162 IN recon squad will observe the eastbound enemy until reinforcements arrive from C/1-162 IN, at which point they will evade and linkup with C/1-162 IN 1/D/1-162 IN platoon, A and B/1-162 IN will continue their current missions.	N/A
	0700	C/1-162 IN airmobile to grid 596928.	1 create
	0704	B Btry airmobile to grid 595825.	1 create
	0705	Enemy mechanized battalion has advanced to position 540940.	1 move
	0715	B/1-162 IN reports enemy in retreat, heading north-northeast at 640893. Pilot recovered. Estimating 90 minutes until AmCits prepared for evacuation.	1 move
FAIL	**FAIL**	**FAIL**	
	0730	A/1-162 IN has reached Vandegrift Blvd., grid 655790, and is heading north.	1 move
	0735	1/D/1-162 IN platoon engages small unit enemy force, current grid.	N/A
	0738	A/1-162 IN encounters hidden enemy sniper force in fortified position at abandoned air station vic. 663833. Begins clearing operation.	1 move (1 create)

	0740	B Btry. begins suppressing enemy mechanized battalion advance. C/1-162 IN has established blocking position, grid 565928.	1 move
	0745	A/1-162 IN has cleared sniper force and is advancing north to engage and clear enemy unit, current grid 654862.	1 move
	0746	1/1-162 IN recon squad is at grid 485925.	1 move
	0748	Enemy mechanized battalion advanced slowed, current grid 551935.	1 move
ON	**ON**	**ON**	
	0750	B/1-162 IN retrieving NEOs and pilot from Obj. B.	N/A
	0801	A/1-162 IN engaged and clearing enemy units at Obj. A, grid 6889.	1 move
	0802	C/1-162 IN has engaged the enemy at its blocking pos., grid 548938.	1 move
	0810	Pilot and NEOs from Obj. B transported safely to carrier.	N/A
	0815	B/1-162 IN has withdrawn.	1 delete
	0817	A/1-162 IN reports entire enemy force vic. Obj. A captured or killed. All NEOs found dead by enemy fire. Withdrawing.	1 delete
	0819	C/1-162 IN reports enemy mechanized battalion badly damaged. Estimate BDA at 50 percent.	1 update
		ENDEX	

Appendix E. Post-test questionnaire

In each category, please check the answer that is most suitable based on your brief experience with Rasa.

Performance

Responsiveness

I was always waiting too long for Rasa to respond	I had occasionally had to wait too long for Rasa	Rasa responded to my commands within tolerance	Rasa responded to my commands better than expected	Responded to my commands immediately
○	○	○	○	○

Cost

Rasa made my job impossible	Rasa impeded my performance significantly	Rasa impeded my performance somewhat	Rasa impeded my performance slightly	Rasa did not impede my performance
○	○	○	○	○

Ease of Use

Not nearly as easy to use as paper	Not as easy to use as paper	As easy to use as paper	Easier to use than paper	Much easier to use than paper
○	○	○	○	○

I have not used C^4I digital systems	Not as easy to use as C^4I digital systems	As easy to use as C^4I digital systems	Easier to use than C^4I digital systems	Much easier to use than C^4I digital systems
○	○	○	○	○

Preference

I would prefer a paper map, rather than Rasa	I would not prefer a system like Rasa over paper alone	I don't have a preference between a paper map and Rasa	I might prefer Rasa over paper alone, if improvements were made	I would prefer using Rasa to a paper map alone
○	○	○	○	○

Errors

In this category the questions relate to the errors that the system made in its attempt to understand you, not in any errors that you may have made.

Quantity

The amount of errors made Rasa unusable	There were more errors than I would have liked	There were an acceptable number of errors	There were only a few errors	Rasa made almost no errors or none
◯	◯	◯	◯	◯

Correction

There were no errors	It was extremely difficult to correct errors Rasa made	It was difficult to correct errors in Rasa	It was moderately difficult to correct Rasa's errors	It was easy to correct errors Rasa made
◯	◯	◯	◯	◯

Training

This question is concerned with how much training you believe that officers in today's military would need in order to use a system like Rasa themselves.

Several days	A day or less	An hour or less	Several minutes	No training required
◯	◯	◯	◯	◯

Personnel

What amount of technical non-officer staff would be required to support a system like Rasa?

Technical personnel will be required to operate	Technical personnel will need to be on site	Technical personnel will need to be on call	Occasional technical personnel required	No technical personnel required
◯	◯	◯	◯	◯

Failure

Continuation

Work could not continue when system failed	Work was stopped for more than ten minutes when system failed	Work was stopped for several minutes when system failed	Work stopped briefly when system failed	No stoppage of work occurred when system failed
○	○	○	○	○

Recovery

It was impossible to recover after a system failure	It was extremely difficult to recover after a system failure	It was difficult to recover the map after a system failure	It was moderately difficult to recover after system failure	It was easy to recover after system failure
○	○	○	○	○

You will now be asked a series of questions that we would like for you to answer in as much detail as possible.

Benefit

What benefits to using Rasa do you see for someone performing this task?

What benefits to do you see for integrating this CTP information with other digital systems?

If paper CTPs were digitally capable how would you take advantage of that? What other capabilities would you suggest be added?

Performance

Responsiveness

If you judged that the performance of Rasa wasn't immediate, do you have suggestions on how this could be corrected?

What would be an acceptable performance delay? What is within tolerance?

Does Rasa's implicit confirmation strategy (where its understanding of what you said/drew is affirmed by continuing to the next command) eliminate some concerns about system performance?

Bother

If Rasa was "in your way," how could this be corrected?

Ease of Use

What could make the system more usable?

Why is it that current C^2/C^4I are often not used in current COCs? List your reasons in order of importance.

Preference

If you would prefer using paper rather than Rasa, why? Is there some aspect of paper CTPs that Rasa doesn't support? How could it support these?

Errors

Quantity

If the number of errors made the system somewhat or significantly unusable, what would be an acceptable number of errors?

Correction

Further, if the correction of errors was painful, how could that be remedied?

Training

If you anticipate more than a day's training to use Rasa, why? What approach would you use to reduce the required training?

Personnel

If you expect that additional technical personnel would be required to have Rasa as a COC capability, why?

Failure

Does Rasa address the issue of power and communications failure of computer systems in a COC by providing its interface on paper? If so, why? If not, why?

Continuation

If you could not continue to update the CTP when the system failed, what were the reasons?

Recovery

If you could not update the digital CTP after the system was re-enabled, what were the reasons?

Permanence

One of the distinguishing features of Rasa is its ability to retain information in the event of failure. What information, other than unit position and overlay information, would be helpful to capture in a similar fashion: for example, BDA. What information is only essential in a transitory fashion and can be dismissed if communications or power are unavailable.

Engineering issues

Input

Assuming that noise cancellation capabilities can eliminate voice recognition errors that would be expected due to noise the field, is the type of voice recognition input you used today acceptable?

What other symbols or symbol attributes would be of immediate benefit?

Rasa is also capable of supporting adding tactical, maneuver, and other symbology, such as boundaries, restricted fire areas, supply routes, objectives, and the like. List which of these are most important to support.

Output

Will text-to-speech capability be useful or will the noisy environment hinder its usefulness even with output delivered by the headset you wore today?

Do you believe that projection could provide useful information, especially if the map is placed in a horizontal format?

Appendix F. Interview Transcripts

The following appendix contains the transcript of those comments made during the interviews by the subjects pertaining to the use of Rasa and its performance. To maintain discretion, certain comments subjects made regarding their personal history and specific military activities have been removed. To remain concise interviewers' questions and other extraneous dialogue are also not included, although phrases that prompted responses, were unclear, or added for clarification are included in [brackets].

SUBJECT 1:

FIRST BATTLE-TRACKING EXPERIENCE OF ANY KIND.

"ONE OF THE EASIEST THINGS WAS MOVING TROOPS. ONCE YOU'VE INITIALLY PLACED YOUR UNITS ON THE MAPS, MOVING THEM WAS REALLY EASY. CREATING THEM WAS PRETTY EASY [TOO]. BIGGEST PROBLEM I HAD WAS DRAWING THE SYMBOL. YOU KNOW THE STICKIE PAD WOULD MOVE AND I'D BE TRYING TO DRAW THE RECTANGLE AND IT WOULD MOVE AND I WOULD MAKE A MISTAKE AND THEN IT WOULDN'T RECOGNIZE WHAT KIND OF UNIT I WAS DRAWING, WHICH I WAS ABLE TO CORRECT EASILY BY JUST PLACING THE UNIT THERE AND THEN TELLING IT WHAT KIND IT WAS."

NOT MUCH COMPUTER EXPERIENCE AT ALL.

"[SPEECH RECOGNITION] SEEMED TO DO FINE FOR THE KINDS OF UNITS WE WERE USING."

"[TEXT-TO-SPEECH OUTPUT] WAS PRETTY CLEAR."

"I FELT PRETTY COMFORATBLE WITH IT TOWARDS THE END. AT FIRST, I WAS STRUGGLING. BECAUSE I HAVEN'T DONE MUCH MAP READING. IT TOOK ME TIME TO FIND THE COORDINATES AND BE SURE WITH THAT. IT'S NOTHING THAT I WOULDN'T HAVE EXPERIENCED IF IT HAD JUST BEEN PAPER. IT WASN'T ANY MORE DIFFICULT. I THOUGHT IT WENT REALLY WELL. I'D BE COMFORTABLE WITH DOING THAT."

"THAT WAS A GREAT EXERCISE FOR ME."

SUBJECT 2:

MAP PLOTTER FOR 3 YEARS AND 15 YEARS PRIOR IN LIGHT INFANTRY

NO C4I, FAIRLY COMPUTER ILLITERATE

"THE BENEFITS THAT I SEE FOR THIS SYSTEM, FOR WHAT WE ARE DOING IN OUR JOB RIGHT NOW IS THE FACT THAT IT'S EASIER TO KEEP TRACK OF WHAT'S GOING ON. YOU'VE GOT TWO WAYS TO TRACK IT. RIGHT NOW, PUTTING ON THE PAPER ICONS AND HAVING THE SYSTEM TO USE AT THE SAME TIME. IT GIVES YOU TWO THINGS TO LOOK FOR. LIKE WHEN THE SYSTEM WHEN DOWN, TO PUT IN THE SYSTEM THE

ICONS I'D POSTED BEFORE WAS RELATIVELY EASY, BECAUSE THERE WERE ALREADY THE ICONS PAPER AND THE COMPUTER-GENERATED WHERE THEY BELONGED AND IT WAS EASY TO SEE JUST BY GLANCING AT THE MAP WHERE YOU NEEDED TO MAKE THE CHANGES AT."

"I HAD NO TROUBLE WITH IT AND I'M JUST EX-INFANTRY. AND IT'S REALLY NO TROUBLE. IT'S RATHER FUN ACTUALLY."

"IT GIVES YOU TWO THINGS TO TRACK. IT'S A LOT EASIER. DOING IT MANUALLY YOU TEND TO HAVE TO USE MORE MANPOWER TO PUT IT UP THERE. FOR INSTANCE THIS WAS POSTING THE UNITS YOU HAVE RIGHT AT THE EDGE OF THE BOARD."

"CURRENTLY YOU HAVE TO GO TO ANOTHER CHART IN ORDER TO POST BDA. THIS IS ALL UP FRONT, SO IT MIGHT ALLEVIATE SOME OF THE CHARTS WE HAVE NOW. SHRINK THEM DOWN. THE CHARTS ARE UP THERE NOW SO THAT THE COMMANDER CAN GLANCE AND TELL WHAT'S GOING ON. WITH THE COMPUTER SYSTEM HE CAN GLANCE AND TELL WHAT'S GOING ON THERE WITHOUT US MANUALLY DUPLICATING WHAT IT'S DOING, BUT IF IT GOES DOWN YOU STILL HAVE TO HAVE THAT. THE THING IS WE COULD USE IT ON A SMALLER BOARD AND ONE PERSON COULD DO THAT WITHOUT JUMPING UP THERE AND MOVING THIS BOARD, ETC. SO, IF WE NEEDED IT WE COULD STILL HAVE IT. BUT FOR HIS ON-HANDS LOOK UP, SEE IT. IT WOULD BE THERE."

"IT SEEMED TO RECOGNIZE EVERYTHING THAT I NEEDED TO PUT INTO IT. MAYBE A LITTLE BIT MORE RECOGNITION ON THE ICONS. YOU'D HAVE TO INCREASE THAT BECAUSE THERE ARE SO MANY DIFFERENT ICONS. BUT, FOR THE ICONS WE WERE USING. IMMEDIATE RECOGNITION. NO PROBLEM."

"[PEOPLE DON'T USE THE DIGITAL SYSTEMS DUE TO] LACK OF KNOWLEDGE AND TRAINING INTENSITY REQUIRED. THESE SYSTEMS TEND TO BE TRAINING INTENSIVE."

"WE HAVE TO USE BOTH [PAPER AND DIGITAL SYSTEMS]. IN OUR SITUATION, WE HAVE TO. BECAUSE WE MOVE, YOU CAN'T KEEP [THE COMPUTER SYSTEMS] GOING. I SUPPOSE YOU COULD WITH A LAPTOP, BUT IT'S JUST NOT FEASIBLE. WE NEED TO BE ABLE TO PICK IT UP AND RUN."

"YOU'VE GOT TO HAVE THAT BACKUP."

"THE NICE THING ABOUT THAT IS, THAT I SEE, WHEN IT REALLY GETS ROLLING AND THE BATTLE IS HOT AND HEAVY, YOU NEED TO GET THAT ON THE MAP IMMEDIATELY. THAT WAY YOU CAN DO IT. WE CAN JUST GO UP THERE AND PUT IT ONTO THE MAP [WITH SPEECH]. THEN WE CAN TAKE CARE OF MAKING SURE THE POST-ITS GET MADE FOR A FAILURE, TO POST THOSE. SPEED IS DEFINITELY AN ISSUE."

"I THINK THE ACCURACY ISSUE NEEDS TO BE WORKED ON, AS FAR AS 6 AND 8 DIGIT GRID COORDINATES."

"THE NICE THING ABOUT RASA TOO IS THAT IT IS TELLING ME THE GRID COORDINATES WHEN I TOUCH IT UP THERE. I KNOW THE GRID COORDINATES I NEED WHEN I TOUCH IT AND IF I AM OFF THEN IT IS SO SIMPLE TO CHANGE IT'S NOT EVEN FUNNY."

"TIME IS A BIG FACTOR WITH WHAT WE DO."

"VERY HAPPY [WITH VOICE RECOGNITION]. MY VOICE DIDN'T TAKE TOO WELL TO IT THERE FOR A WHILE, BUT I JUST CHANGED MY PITCH A LITTLE BIT AND IT WORKED FINE. I WAS TRYING TO ENUNCIATE TOO MUCH AND IT JUST WANTS YOU TO TALK NORMAL."

"I LIKED THE [VERBAL TEXT-TO-SPEECH CONFIRMATION] VERY MUCH...WHAT I LIKED ABOUT IS THAT IT IS TELLING YOU, 'YEAH, YOU'RE DONE. YOU'VE DONE IT. THAT'S CORRECT.' FEEDING IT BACK TO YOU. IF YOU HEAR IT WRONG, OF COURSE, YOU CHANGE IT... IT REPEATS THE GRID COORDINATES."

"I THINK IT MAKES THE JOB EASIER."

"AND THE COMPUTER'S TELLING YOU IF YOU'RE RIGHT TO THE MAP, IF THE MAP'S RIGHT ON THE TOUCH SCREEN. YOU'VE GOT CONTROL POINTS ON THE OVERLAYS AND YOU CAN TELL JUST BY TOUCHING IT. IT'S GOING TO TELL YOU THE GRID, 'YEAH THAT'S RIGHT OR WHOOPS WHY IS THAT ONE WRONG.' AND YOU CAN TELL IF YOU'RE OFF. SO, YEAH, THAT IS HANDY. I'D LIKE TO SEE IT COME INTO BEING."

SUBJECT 3:

OPERATIONS NCO. OVERSEE MAP-PLOTTING ACTIVITIES S3. NOT MUCH BATTLE-TRACKING EXPERIENCE.

FIRST TIME HE HAS USED THIS TYPE OF A SYSTEM, NEVER USED A C4I DIGITAL SYSTEM BEFORE.

"IN THE EVENT THAT THE TOC WERE TO BREAK DOWN AND JUMP [MOVE FROM ONE PLACE TO ANOTHER], STICKIES GO HELTER SKELTER, THEN IT WOULD BE RELATIVELY EASY TO SET THE PROJECTOR UP, SHOOT IT UP THERE, AND FOR EVERYTHING TO BE IN THE RIGHT LOCATION. THAT WOULD PROBABLY BE THE LARGEST. [USE THE COMPUTER AS A BACKUP FOR THE STICKIES. STICKIES AS PRIMARY.]"

"[CREATE THE ICONS WITH SPEECH AND DRAWING,] I LIKED A LOT. ONE OF THE THINGS THAT I DIDN'T LIKE WAS HAVING TO DRAW THE STICKIE ON THE PAD. I LIKED IT BETTER JUST CREATING IT THERE ON THE PAD, SPEAKING IT. THAT REQUIRES WHOEVER IS OPERATING THE BOARD TO GO AND SIT DOWN ON THE PAD, DRAW IT OUT. IT WOULD BE MUCH BETTER IF YOU COULD JUST, ON A THREE BY FIVE CARD DRAW IT OUT REAL QUICK, TOUCH ON THE MAP, SAY WHAT IT IS, AND THEN STICK YOUR STICKIE UP THERE."

"IT WAS CONFUSING USING THE STICKIE AND THE GRAPHIC AT THE SAME TIME, BUT...I GUESS THE CONFUSING PART WAS SEEING THE GRAPHIC AND ALL OF THE STICKIES

IN A TIGHT SPOT. AT TIMES I COULDN'T TELL WHAT WAS THERE. THE MORE I USED THE GRAPHIC, THE MORE I LIKED IT. ABSENT ALL THOSE STICKIES ON THE BOARD, I THINK IT MIGHT BE EVEN BETTER. WHEN THE POWER GOES DOWN YOU'RE GOING TO HAVE TO HAVE AN ALTERNATE MEANS OF TRACKING. HENCE THERE IS NO WAY TO GET AWAY FROM THE STICKIES."

"I THINK MAYBE THE GRAPHIC ON THE SCREEN IS A LITTLE LARGE, AND THE POST THAT LED DOWN TO THE UNIT, AND THE DESIGNATION ON THE MAP, THAT BIG LARGE CIRCLE, THE HOT SPOT, WAS TOO LARGE. IT NEEDS TO BE NOT MUCH THAN THE SECONDARY TIP OF THAT PEN, I WOULD THINK. THE MORE CLUSTERED [UNITS] GET... IF YOU WERE TO HAVE SEVERAL OF THOSE [UNITS] CLOSE TO EACH OTHER, WHICH HOT SPOT WOULD YOU ACTUALLY BE ACTIVATING WHEN YOU TOUCH?"

"I WOULD PREFER PAPER AND DIGITAL AT THE SAME TIME. IF I HAVE THAT OPPORTUNITY. YES, I WOULD. IF FOR NO OTHER REASON THAN THE JUMPING IN THAT TOC. BECAUSE AS THINGS GET MOVED AROUND IN THE BACK OF A HUMVEE..."

"[SETTING UP A TOC CAN TAKE] UP ABOUT TWO HOURS FROM START TO FINISH AND OF COURSE THAT IS JUST THE PHYSICAL SETUP. UPDATING ALL THE MAP BOARDS, DEPENDING ON WHAT THEY HAVE GONE THROUGH CAN TAKE UP TO ANOTHER TWO, TWO AND A HALF HOURS."

"THE SAME CREW THAT WAS POSTING THEM WHEN WE JUMPED, PULLS OUT THOSE MAP BOARDS AND VERIFIES THAT EVERYTHING IS CORRECT BY MEMORY AND BY THE MESSAGE FORMS THAT WE GOT IN THE LOG BOOKS SO FAR."

"SOME OF THE PROBLEMS THAT WE RUN INTO IS THE BATTLE HAND-OFF BETWEEN THE TOC AND THE TAC, WHICH IS WHERE THE BATTALION COMMAND IS RUNNING THE BATTLE FROM HIS HUMVEE BASICALLY. AND WHATEVER HAS HAPPENED DURING THE COURSE OF THAT TOC JUMPING AND RELOCATING AND GETTING SET BACK UP AND HIM TRACKING. SOMETIMES THERE ARE SEVERAL HOURS THAT TRANSPIRE BETWEEN HIM GETTING THAT INFORMATION ON HIS BOARD BACK TO OUR PHYSICAL LOCATION AND UPDATING IT."

"IF IT WERE DOWNLOADABLE DIRECTLY TO OUR EVEN DIGITALLY OVER A [NETWORK] OF SOME TYPE. THAT WOULD DEFINITELY MAKE A MUCH TIGHTER PROCESS."

"SOME OF IT, I FELT, WAS MY OWN INADEQUACY WITH THE SYSTEM. NOT MAKING SURE THAT I IDENTIFIED THINGS IN THE RIGHT ORDER. WITH MORE TIME ON THE SYSTEM, I THINK IT WOULD HAVE BEEN A BETTER EXPERIENCE."

"[IT IMPEDED MY PERFORMANCE SOMEWHAT] IN THE NAMING OF THE UNITS ACCURATELY AND QUICKLY. I THINK THAT YOU CAN PROBABLY COME UP WITH, WITH TRAINING, IF YOU TRAIN THE OPERATORS WHO ARE USING IT, SOME KIND OF A NUMBER SYSTEM, ALPHA NUMERIC, AND IT WOULD NAME THEM QUICKLY. I HAD TROUBLES WITH THE

DELTA UNIT [AND THE BRAVO UNIT.] BUT IF I WERE TO SAY LIKE, DELTA-ONE-ONE-SIX-TWO THEN IT WOULD AUTOMATICALLY KNOW THAT WHEN I SAY DELTA I AM TALKING ABOUT THE COMPANY. OR MAYBE I WOULD GO ONE-DELTA-ONE-ONE-SIX-TWO, THEN IT KNOWS I'M TALKING ABOUT THE FIRST PLATOON OF DELTA ONE-ONE-SIX-TWO AND THEN DISPLAY IT ON THE GRAPHIC ACCORDINGLY. OF COURSE, THAT WOULD JUST BE THE TRAINING OF THE OPERATOR TO USE THE RIGHT NAMES.]"

"I LIKE THE EASE THAT YOU COULD MOVE UNITS WITH."

"[I THOUGHT RASA WAS AS EASY TO USE AS PAPER] WITH A LITTLE MORE TRAINING ON THE SYSTEM."

"I LIKED THE ABILITY [TO SPEAK THE UNIT, TOUCH THE MAP] AND NOT HAVING TO GO TO THE PAD AN DRAW IT FIRST. BUT, WE'VE GOT TO CONTINUE THE PAPER ANYWAY. GIVEN THAT IT GIVES US A RELIABLE BACKUP, IT'S EASIER. IT IS AN ADDITIONAL TASK. ONE OF THE CHALLENGES THAT I THINK WE WOULD HAVE IN THE TOC IS THAT SOMETIMES THINGS GET PRETTY LOUD AND TRYING TO KEEP IT QUIET ENOUGH TO RECOGNIZE THOSE COMMANDS WITH JUST THAT OPERATOR, THAT MIGHT BE A CHALLENGE."

"[THE NUMBER OF ERRORS] WERE TOLERABLE WITHIN THE CONSTRAINTS THAT WE HAD TODAY... BRAND NEW OPERATOR THAT HAS NEVER USED THE SYSTEM BEFORE, AND I DON'T HAVE THAT MUCH BATTLE TRACKING EXPERIENCE EITHER. [CORRECTING ERRORS] WAS TOLERABLE."

"I THINK IN A DAY AN OPERATOR COULD WALK AWAY WITH WHAT HE NEEDED TO DO WITH THE SYSTEM. MOST DEFINITELY."

"I ALMOST THINK THAT THIS DIGITAL SYSTEM LENDS ITSELF TO TRAINING THE MANUAL SYSTEM AS WELL. [RASA] IS GREAT TRAINING FOR THE MANUAL SYSTEM THAT WE USE."

"TECHNICAL PEOPLE NEED TO BE AT LEAST ON CALL, BUT THERE CONTINUALLY... NO."

"[WHEN THE SYSTEM FAILED, I WAS ABLE TO KEEP MY WORK GOING.] WITH THE SYSTEM STOPPED, COULD I CONTINUE WITH THE DIGITAL SYSTEM? NO, I COULDN'T. BUT COULD I CONTINUE THE POSTING. OH HECK YEAH. NO PROBLEM."

"LOOKING TO SEE WHERE UNITS HAD MOVED, THE STICKES, AND LOOKING TO SEE WHERE THE GRAPHIC WAS. IT WAS MODERATLY DIFFICULT [TO RECONCILE AFTER FAILURE], BUT NOTHING THAT WAS OVERWHELMING. AND I THINK I MAY HAVE MISSED ONE UNIT, AN ENEMY UNIT, AND A SECTION OF DELTA. AND I THINK THAT MIGHT BE LINKED BACK TO THE NAMING OF THOSE UNITS..."

"WHAT WE USE IN OUR TOC ARE PUSHPINS, NOT STICKIES. AND THE REASON WE DO THAT IS BECAUSE SO MANY STICKIES BECOME SO CLUTTERED THAT YOU CAN'T SEE WHAT IS GOING ON. WHATEVER SYSTEM THAT WE END UP USING NEEDS TO HAVE something small enough to be concise, but very readable. [We put a] little plastic flag [on the back of the pushpins] and its written on there and its also by color code, but it destroys the map and the overlay and they fall out relatively easily."

"The scale of the map is also a challenge. The one you had in there is a 1-32. We usually see 1-25 or 1-50. Using your system on a 1-50 might present a larger challenge. 1-25 may make it easier."

"It is [translating between map scales] is an enormous headache. What happens is... We've currently changed so that now we are under the 7th Infantry's umbrella. Well, 7th ID is telling us all that we need to be using the 1 to 25s. Well, prior to the warfigher exercise all of the operations orders were developed for warfighter several months before on 1 to 50s and consequently all the operational overlays as well. So, trying to make new graphics the day of the warfighter is virtually impossible. So, brigade tried to jump through their butt to get graphics on the 1 to 25, so that we can use them on the map that division had provided us that was one to 25. So what we did is we took all of the 1 to 25 stuff, well I can't say all, most of the 1 to 25 stuff and put it in future plans and used the 1 to 25s for the future planning cells and fought the battle on 1 to 50s."

"If we had a system that could in an instant could flip-flop the graphics from 1 to 50 and 1 to 25. Oh my god! The headaches we could save. That is a huge problem."

"The amount of man hours spent with different colored pens over an overlay is something that irritates me to no end."

"This tool in future plans would be very, very useful."

"I hope the best for this project. I would sure like to see something different than what we currently have."

"One more thing on the graphics... Just an idea, I don't even know if you guys can make it work. Instead of having to draw out the icon, you know off to the side of that board where you had the unit, if you had just a small digital representation of most of the generally used graphics, and the guys could just touch it and touch on the board, then we've saved, in some maintenance issues maybe a minute. That would make it an even better system. Then we would name them. You could just touch, touch, name. And there it is. Oh, and then also the BDA. Currently, we have to have a whole nother chart that hangs off to the side of our

OPERATIONAL MAP BOARD, THAT BREAKS DOWN WHERE EACH UNIT IS ON VARIOUS CLASSES OF SUPPLY. [THAT INFORMATION IS RADIOED IN FROM THE CTCP.] IT COULD BE INTERPRETED THAT SOME OF THIS INFORMATION IS REQUIRED FOR THE CTCP, BUT NOT FOR THE TAC. WE DON'T MONITOR THAT INFORMATION AS TIGHTLY AS THE CTCP. IF WE DON'T HAVE CURRENT STATUS, WE CALL THE CTCP. 'OKAY THIS IS THE MOST CURRENT THING ON ALPHA COMPANY.' THE PERCENTAGES ARE BROKEN DOWN IN OUR TACTICAL SOP."

"IT WAS A LEARNING EXPERIENCE AND I HOPE TO SEE IT AGAIN."

SUBJECT 4:

ASSISTANT OPERATIONS NCO WITHIN THE BATALLION TOC. 25 YEARS AS ENLISTED AND OFFICER NCO. TOTAL BATTLE TRACKING 7-8 YEARS.

COMBINATION OF UPDATING MAP BOARDS, BATTLE TRACKING, INFORMATION AND MAP UPDATE. "IF THE BATTLE CAPTAIN IS NOT PRESENT THEN I HAVE SEVERAL HATS ON."

"CURRENTLY MAP AVAILABILITY COMES AT 1 TO 50,000. TOO TINY. WE'RE JUMPING AROUND KIND OF EITHER USING STICK PINS, WHICH ARE LIMITED IN HOW YOU CAN IDENTIFY THE UNITS, COMPARED TO STICKIES. SMALLER MAP SIZE MAKES IT REAL DIFFICULT. REUSE OF MAPS. THE BIGGER THE MAP SIZE IS MORE DESIRED. EXPANDING THE MAP SYMBOLS TRAINED, MAKES IT EASIER FOR US THEN TO TRACK ON THE MAP SYSTEM ITSELF. ALSO THE READABILITY IS GREATLY IMPROVED THE BIGGER THE MAP."

"I CAN SEE [HAVING PAPER AND COMPUTING SYSTEM COMBINED], YOU'VE GOT A BACKUP. I GUESS. YOU STILL NEED IT VISIBLE IF THE DIGITAL WENT DOWN. THEN, I WAS ALREADY PLOTTED. SO, YEAH, IT WAS ALREADY THERE. ONE OF THE QUESTIONS ABOUT RECOVERY. THAT WAS SHIFTED, BECAUSE THE SYSTEM WENT DOWN AND YOU'VE ALREADY GOT IT UP THERE."

"[DURING MOVEMENT OF THE TOC,] WE HAVE A SYSTEM WHERE WE'RE TRACKING, BUT THERE IS A FORWARD TAC THAT TAKES OVER THE BATTLE TRACKING. WE'RE STILL TRYING TO DO AS MUCH AS WE CAN IN A SKELETON. CAUSE THEN YOU START PULLING THE PEOPLE INSIDE OUT TO DO THE GRUNT STUFF."

"WE RETAKE OVER THE BATTLE ONCE WE GET IN A POSITION WHERE WE CAN DO IT ONE HUNDRED PERCENT. THE IDEA IS TO TRY TO TRACK ALONG, SO THAT WE DON'T HAVE THE TRANSITION."

"[A SYSTEM THAT CAN UPDATE BOTH THE TAC AND TOC MAPS SIMULTANEOUSLY] WOULD BE A DEFINITE PLUS. CAUSE IN ONE ANNUAL OF TRAINING WE MOVED EIGHT TIMES. SO,... DEPENDING ON THE ACTION AND WHAT IS GOING ON THEN DEFINITELY YEAH.

CAUSE, IF THEY'VE GOT THE SAME SYSTEM THEY'RE UPDATING, OURS IS ALREADY UPDATING BY THE TIME THEIR GONE, BUT IT'S NOT AS IF WE CAN'T CAPTURE IT."

"THERE WAS AN ELEMENT OF SPEED. I COULD GO [TO THE MAP] AND VERBALIZE, BUT THEN I WOULD HAVE LOST [THE PERMANENT STICKIE]. EARLY ON, YOU CAN WRITE AND ITS IN. BUT,I COULD HAVE ELIMINATED [THE SKETCHING]. I WAS THINKING OF SPEED. IN FACT, I THOUGHT ABOUT DOING THAT, BUT THEN I'M GOING TO LOSE THE ICONS."

"[WHEN THE SYSTEM FAILED,] IT WAS JUST A MATTER OF SAYING IT'S DOWN, PLEASE FORGIVE US. DRIVE ON. I GUESS. SO THAT WAS... MAINTAIN PLOT. NOT VERY LONG, I GUESS."

WORRIED ABOUT HEADSETS AND RECOGNITION IN THE ACTUAL TALK. NEED FEEDBACK FROM THE NOISY ENVIRONMENT.

"I WAS CREATING. NORMALLY I WOULDN'T HAVE TO CREATE."

"COMING FROM TOC-LEFT, IF I'M CENTER ON OPS MAP, THE FIST...CLEARANCE OF FIRES, AND HE'LL YELL A GRID. SO, IF IT'S INTERFACED AND LET'S SAY HE'S MICED, 564-286. ITS PLOTTED ON HIS, MINE, AND OTHERS IMMEDIATELY. AND I'M NOT [SEARCHING FOR THE GRID]. IT'S MINIMUM 6 DIGITS, SOMETIMES 8... ONCE AGAIN THE SIZE OF THE MAP YOU'RE WORKING WITH...THE SMALLER IT IS ITS TOUGHER. THAT'S WHY IT IS EASIER TO USE PUSHPINS... THEN THERE'S THAT LIGHT. YOU COULD YELL CLEAR OR NO WITHIN... BASICALLY THEY WANT TO SHOOT A MISSION, ARE THERE FRIENDLY TROOPS WITHIN 4 OR 500 METERS OF THERE."

SUBJECT 5:

8 YEARS INTELLIGENCE SEARGENT FOR AN INFANTRY BATTALION

"GIVEN THAT IT WAS SPEEDED UP AND SOME CHANGES MADE, IT MIGHT BE EASIER TO ADD IN, AS I'M READING THE GRID COORDINATE FROM THE PAPER THAT I AM HANDED, I CAN INPUT INTO THE SYSTEM AND POST TO THE MAP AT THE SAME TIME A UNIT'S LOCATION. IT TAKES MORE TIME TO HUNT FOR THE GRID COORDIANTE AND POINT THAN IT DOES TO READ IN THE GRID COORDINATE. THAT WOULD BE MORE EFFICIENT THAN CREATE THE ICON, HOLD IT UP, FIND THE GRID, PLACE IT THERE, AND THEN ANNOUNCE TO THE SYSTEM WHAT IT WAS."

"IN WHAT YOU HAVE HERE I STILL HAVE TO CREATE THE UNIT MANUALLY ALSO. ONE THING THAT WE DO TO SPEED UP POSTING IS TO NOT CREATE THE ICONS AS WE NEED THEM. WE HAVE A LIBRARY OF ICONS. WE GOT A MECHANIZED BATTALION. MECHANIZED BATTALION [GRABS IT FROM THE LIBRARY], AND THEN YOU READ THE GRID, AND PUT IT THERE. SO THAT IS PART OF OUR PREPARATION FOR THE BATTLE IS TO MAKE SURE THAT OUR UNIT AND ACTIVITY ICONS ARE CREATED AND READY TO GO. [IF WE RUN OUT OF THEM,] THEN WE HAVE TO CREATE THEM. WE CAN CREATE

HUNDRED, THOUSANDS. BRIGADE RECREATES THEM EVERY TIME. I HAVE A CAD PROGRAM CALLED TURBOCAD THAT I CREATE MY OWN SYMBOLS IN. AND I PRINT THEM OUT ON A TRANSPARENCY FROM AN INKJET PRINTER. I COVER THEM AND I PRINT THEM. AND THEN I HAVE AN INVENTORY OF ICONS."

"THE SPEED OF THE SYSTEM WAS SUCH THAT I WAS SLOWED DOWN."

"I TOUCH THE SCREEN. I SAY UNIT HERE. AND IT SAYS, 'OK, UNIT HERE.' WHEREAS, I WOULD BE WAITING FIVE TO TEN SECONDS FOR IT TO ACKNOWLEDGE THAT AND ASK FOR A CONFIRMATION. IT SEEMED LIKE. MY EXPERIENCE, ESPECIALLY WHEN WE'RE STARTING TO SEE ENEMY POP UP HERE AND THEN IS THAT THINGS START FLYING, ESPECIALLY IN OUR BATALLION TOCS, AND IF YOU'RE HAVING TO WAIT FOR THAT SYSTEM TO UPDATE AND CONFIRM THIS, AND YOU REACH FOR THE CONFIM. YOU'RE ALREADY WAY BEHIND. AND YOU'RE GOING TO HAVE A COMMANDER OR XO STANDING THERE TAPPING HIS FEET, GOING HURRY UP."

"IF IT IS ON A LARGE SCREEN AND I CAN LOOK OFF TO THE SIDE AND SEE THAT THE UNIT IS POSTED CORRECTLY THEN ANOTHER CONFIRMATION MAY NOT BE NECESSARY."

"YOUR VOICE INPUT, I THINK, IS GOOD. I HAVEN'T DEALT WITH VOICE INPUT BEFORE. IT MADE IT HARDER TO USED. BUT PART OF MY PROBLEM IS THAT I'M NOT FAMILIAR WITH THE SYSTEM. AS I GOT COMFORTABLE WITH THE SYSTEM, THE ERRORS WOULD GO DOWN I THINK. THERE WAS, LIKE ONE ERROR, I COULDN'T CORRECT."

"[AFTER THE SYSTEM FAILURE, THE ABILITY TO GET THE THING BACK UP TO SPEED] WAS NOT A BIG PROBLEM, I JUST HAD TO LOOK UP ON THE BOARD AND FIGURE OUT WHICH UNITS HADN'T BEEN ADDED IN AND WHICH UNITS HAD BEEN MOVED AND DO THAT. AND THAT TOOK ME, I DON'T KNOW, FIVE MINUTES TO DO THAT. IN THE HEAT OF THE BATTLE THAT MIGHT BE TOO LONG."

"THE ONLY WAY THE SYSTEMS FAIL NOW IS IF YOU GET ATTACKED AND YOU HAVE TO RUN AWAY. POWER GENERATORS [FAIL] SOMEBODY TURNS ON A FLASHLIGHT OR A LAMP [BECAUSE WE ARE WORKING OFF PAPER]. COMMS ARE ALMOST ALWAYS AN ISSUE. COMMS ARE NEVER PERFECT. WHAT I SEE IS AN ISSUE WITH THE POWER GENERATION IS THAT WE'VE ALWAYS GOT THE HUMVEE SITTING THERE AND WE DON'T HAVE ARRANGEMENTS FOR TAPPING THE BATTERY POWER (THE ENGINE POWER) OF THE HUMVEES TO POWER THE ELECTRICAL SYSTEMS. WE HAVE PORTABLE GENERATORS. WE FINALLY HAVE QUIET PORTABLE GENERATORS. BUT, YES, IF THE GENERATOR GOES DOWN, WE CAN'T FIRE UP A HUMVEE AND RUN, AT LEAST THE COMPUTING STUFF, OFF OF THE POWER FROM THE HUMVEE."

"TYPING IS INEXACT. PEOPLE USUALLY TYPE ATROCIOUSLY, ESPECIALLY WHEN THEY ARE UNDER PRESSURE. SO, POINTING, CLICKING, AND PULLING DOWN AND GRABBING SOMETHING IS MORE ACCURATE."

SUBJECT DISCUSSED CURRENT S2 PRACTICE AND INTELLIGENCE REPORTING. THEN PHIL DEMONSTRATED ADDING OBSTACLES.

SUBJECT 6:

6 MONTHS AS BATTALION S2.

"THE FIRST THING I NOTICED IS IT DIDN'T SEEM TO BE CLUTTERED AND THE EASE OF BEING ABLE TO UPDATE THE INFORMATION AS IT CHANGED."

"[IT WAS RELATIVELY EASY TO UPDATE THE MAP DUE TO RASA'S] SPEED, BEING ABLE TO DO BDA BY JUST TOUCHING IT AND BEING ABLE TO UPDATE IT VS. CREATING A WHOLE NEW STICKIE NOTE OR FLAG."

"THE SYSTEM PERFORMED WELL. I SAY THAT GOING FROM A MANUAL SYSTEM, WHERE IT TOOK A WHILE TO RECIEVE INFORMATION, BEING ABLE TO PROCESS IT, AND THEN PUTTING IT ON THE BOARD. THIS RASA SYSTEM ALLOWED [ME] TO DO IT ALL AT ONCE, REDUCING THE TIME TO UPDATE THE BOARD. YES, CORRECT. [THE SYSTEM RESPONDED ALMOST IMMEDIATELY.]"

"I WOULD SAY THAT RASA MADE OUT OF 10, 3 OR 4 ERRORS: 30-40%, OF THE MISTAKES. THE REST WERE USER ERRORS."

"AT TIMES I FOUND IT DIFFICULT TO RECOGNIZE THAT AN ERROR HAD BEEN MADE. FOR EXAMPLE, I HAD PLACED A UNIT UP THERE AND IT WOULD COME BACK AND SAY A DIFFERENT UNIT, [BUT I MIGHT NOT NOTICE IT]."

"[ONCE AN ERROR WAS NOTICED, IT WAS] VERY EASY [TO CORRECT IT]. I WAS ABLE TO TOUCH IT, CORRECT IT AND MOVE ON."

"ONCE YOU RECOGNIZED THE SYSTEM HAD FAILED, YOU COULD CONTINUE ON. THE ONLY THING THAT I COULD SEE YOU WOULD BE MISSING WITH THE SYSTEM DOWN IS IF YOU DIDN'T CLEARLY MARK ON YOUR POST-ITS UNIT IDENTIFICATION I.E., WHETHER IT'S BRAVO COMPANY OR NOT. BUT OTHER THAN THAT, IT WOULDN'T BE MUCH TO RECOVER FROM A SYSTEM FAILURE."

"IT WAS VERY EASY [TO RECOVER FROM FAILURE]. YOU COULD IDENTIFY WHERE THE SYSTEM IS TELLING YOU, AND WHERE YOUR POST-ITS ARE AND MAKE THOSE CHANGES RELATIVELY EASILY. INITIALLY I HAD A HARD TIME FINDING THESE DISCREPANCIES."

HE RELATED HOW IT WOULD BE GRAND TO HAVE THE SITREPS AND OTHER JOURNAL ENTRIES AUTOMATICALLY CAPTURED FOR SEARCH.

HIS TOP THREE PICKS: 0) MULTIPLE VIEWS 1) DRILL DOWN AND OTHER QUERIES: BDA, FOOD, FUEL, ETC., 2) CONTROL MEASURES, INCLUDING LOGISTICS, 3) PRINTING, 4) AGENT PROCESSING OF ENEMY INFORMATION OR EVEN FRIENDLY SITREPS,

SEARCHING FOR THINGS LIKE DUPLICATE REPORTING FOR DECONFLICTION, AND 5) TIME-BASED ANALYSES.

"THE ERRORS WEREN'T SOMETHING THAT REALLY IMPEDED ME. IT WAS MORE JUST GETTING COMFORTABLE WITH THE SYSTEM. I RECOGNIZED THE ERRORS AS THEY OCCURRED AND DEALT WITH THEM ACCORDINGLY."

"THE DETAIL THAT I HAVE IN A MANUAL SYSTEM IS NOT AS GREAT AS THIS SYSTEM PROVIDES, SO WITH THAT WHEN A SYSTEM FAILURE DOES OCCUR, I AN NOT WITHOUT. I CAN STILL OPERATE. IT'S LIKE USING A GPS VS. A COMPASS. MY GPS GOES BAD, I STILL HAVE MY COMPASS. IT'S NOT AS ACCURATE, BUT I CAN CONTINUE ON."

"[IN THE FIELD], YOU SHOULD DEFINITELY HAVE MULTIPLE INPUTS. YOU HAD THE PAD WITH POST-ITS, THE VOICE, BEING ABLE TO TYPE IT IN, ALSO MAYBE NOT NECESSARILY USING THE POST-ITS. IF I'M UP AT THE BOARD, I WANT TO BE ABLE TO TAKE MY STYLUS AND JUST POINT AND BE ABLE TO CREATE A UNIT RIGHT THERE, WITHOUT HAVING TO GO TO MY POST-IT."

"THE AUDIO IS FINE. THE VIDEO WILL NEED TO BE MOUNTED UP HIGH, SO THAT IT CAN PROJECT DOWN, BECAUSE YOU WILL HAVE PEOPLE IN FRONT OF THE MAP BOARD."

"I LIKE THE COMMANDS YOU GIVE IT: E.G., 2ND RECON SQUAD. THOSE ARE PRETTY STANDARD. AND JUST HAVE EVERYTHING STANDARDIZED, AS IT APPEARS YOU HAVE DONE. THAT MAKES IT EASY TOO, CAUSE WE'RE ALL SPEAKING THE SAME LANGUAGE."