THE EFFICIENCY OF MULTIMODAL INTERACTION: A CASE STUDY

P. R. Cohen, † M. Johnston, D. McGee, S. L. Oviatt, J. Clow, I. Smith
Center for Human-Computer Communication
Oregon Graduate Institute of Science and Technology
pcohen@cse.ogi.edu

ABSTRACT

This paper reports on a case study comparison of a direct-manipulation-based graphical user interface (GUI) with the QuickSet pen/voice multimodal interface for supporting the task of military force "laydown." In this task, a user places military units and "control measures," such as various types of lines, obstacles, objectives, etc., on a map. A military expert designed his own scenario and entered it via both interfaces. Usage of QuickSet led to a speed improvement of 3.2 to 8.7-fold, depending on the kind of object being created. These results suggest that there may be substantial efficiency advantages to using multimodal interaction over GUIs for map-based tasks.

1. INTRODUCTION

Before spoken interaction can pervade human-computer interaction, situations and dimensions need to be identified in which it is superior to graphical user interfaces (GUIs). Many studies have attempted to investigate the claim that spoken language interfaces will be more efficient than other interface technologies. However, generally speaking, the results have been equivocal. In early "wizard-of-Oz" (WOZ) simulations, researchers have found a *potential* 2-3 fold speed advantage of speech over typing [1-4]. Early studies of speech systems report efficiency gains in the neighborhood of 20% - 40% on a variety of hands-busy tasks [5-7] as compared with keyboard input. However, many studies also report that once the time taken for error correction is included, the expected advantages of speech can evaporate [3, 8]. For example, in a comparison of speech, keyboard, and scroll bars, Rudnicky [9] found that speech was preferred by users despite the fact that spoken control of lists was slower than use of a scroller, once error correction time was included. A recent study comparing spoken interaction with other input modalities found that although speech is preferred, a 94% recognition rate would be required for a speech interface to achieve *equivalent* performance to various manual input modes [10]. Such word recognition rates have been attained by recent high-performance spoken dialogue systems (e.g., the ATIS systems, such as [11, 12]), but those systems have not been systematically compared with graphical user interfaces. It is fair to say that, despite its obvious advantages for hands/eyes-busy and telephone-based tasks, research has still not identified circumstances in which spoken interaction is superior to the ubiquitous GUI, when both are possible. Likewise, the substantial speed improvements suggested by early WOZ simulations not been attained.

In a recent series of high-fidelity WOZ simulations, it has been demonstrated that multimodal communication involving speech and pen-based gesture offers potential task performance and user preference advantages over speech-only interfaces in map-based tasks [13]. If speech in fact offers advantages over GUI-based interfaces, it would then be expected that multimodal interaction should lead to still greater benefits. However, the existence and magnitude of any such performance advantages with implemented systems have yet to be documented.

This paper reports on a case study comparison of a direct-manipulation-based [14] graphical user interface with the QuickSet pen/voice multimodal interface [15, 16] for supporting the task of military force "laydown." In this task, a user places icons representing military units, such as the 82nd Airborne Division, and "control measures," such as various types of lines, obstacles, objectives, etc., on a map. A "backend" application subsystem takes the user specifications and attempts to decompose the higher echelon units into their constituents, positioning them onto the map subject to the control measures and features of the terrain. In the next section, we describe this system and its graphical user interface.

1.1 ExInit

Ascent Technologies, ATI Incorporated, MRJ Corporation, and the Oregon Graduate Institute have developed a new Exercise Initialization tool for the Department of Defense called ExInit. The job of this system is to create the force laydown and initial mission assignments for very large-scale simulated scenarios. Whereas previous manual methods for initializing scenarios required many person-years of effort, such a scenario recently took a single ExInit user 63 hours, most of which was computation.

ExInit provides a GUI based on the Microsoft Windows suite of interface tools, including a "browser", drop-down scrolling lists, buttons, etc. The user would employ the unit browser to explore the echelon hierarchy until the appropriate echelon is "opened," and the desired unit is located. The user then would select that unit, and drag it onto the map in order to position it on the terrain. The system then asks for confirmation of the unit's placement. Once confirmed, ExInit decomposes the unit to the requested level of the hierarchy.

To create a linear or area control measure, the user would "pull down" a list of all control measure types, scroll if necessary, and

---

† First author: Center for Human-Computer Communication, Department of Computer Science, Oregon Graduate Institute of Science & Technology, P.O. Box 91000, Portland, OR (pcohen@cse.ogi.edu; http:www.cse.ogi. edu/CHCC/)

select the desired type. Then the user would click on a button to start entering points, select the desired locations, and finally click the button to exit the point creation mode. The user is asked to confirm that the selected points are correct, after which the system connects them and creates a control measure object of the appropriate type. Many military systems, such as ModSAF [17] a military simulator, incorporate similar user interface tools for accomplishing the force laydown task.

1.2 QuickSet

QuickSet is a handheld, multimodal (pen/voice) interface for map-based tasks. With this system, a user can create entities on a map by simultaneously and continuously speaking and drawing [15, 16]. A major design goal for QuickSet is to provide the same user input capabilities for handheld, desktop, and wall-sized terminal hardware. We believe that only voice and gesture-based interaction comfortably span this range. QuickSet provides *both* of these modalities because it has been demonstrated that there exist substantive language, task performance, and user preference advantages for multimodal interaction over speech-only and gesture-only interaction with map-based tasks [13, 18]. Specifically, for these tasks, multimodal input results in 36% fewer task performance errors, 35% fewer spoken disfluencies, 10% faster task performance, and 23% fewer words, as compared to a speech-only interaction. Multimodal pen/voice interaction is known to be advantageous for small devices, for mobile users who may encounter different circumstances, for error avoidance and correction, and for robustness [19]. Furthermore, our earlier empirical research [13, 18] has identified numerous advantages of a multimodal pen/voice interface for map-based tasks, such as simulation setup.

The QuickSet interface presents a geo-referenced map, such that entities displayed on the map are registered to their positions on the actual terrain, and thereby to their positions on each of the various user interfaces connected to the facilitator. The map interface provides the usual pan and zoom capabilities, multiple overlays, icons, etc. Two levels of map are shown at once, with a small rectangle shown on a miniature version of the larger scale map indicating that portion of it being shown on the main map interface.

Employing pen, speech, or more frequently, multimodal input, the user can annotate the map, creating points, lines, and areas of various types. The user can also create entities, give them behavior, and watch a simulation unfold. When the pen is placed on the screen, the speech recognizer is activated, thereby allowing users to speak and gesture simultaneously. Speech and gesture are recognized in parallel, with the speech interpreted by a natural language parser. The meaning representations derived from speech and gesture are each represented as feature structures [20], with the final multimodal interpretation arrived at through a unification process [21] subject to empirically-derived temporal constraints [13]. At the user's choice, the system offers two modes of confirmation – allowing the user to confirm the recognized speech, or to confirm the system's entire multimodal interpretation [22]. The latter is advantageous because it allows the system to use each mode to compensate for errors in the other.

The system's interpretive processes, as well as the target application subsystems, operate in parallel and are coordinated by a facilitator agent in the Open Agent Architecture [23] (see Figure 1). ExInit's servers (e.g., unit deployment) are connected to the CORBA bridge agent shown in Figure 1. Thus, multimodal input to QuickSet can directly cause operations by the ExInit deployment servers, bypassing the ExInit GUI.

Figure 2 shows an image of the QuickSet user interface as it is being used for force laydown. For the this task, the user either selects a spot on the map and speaks the name of a unit to be placed there (e.g, "mechanized company"), or draws a control measure while speaking its name (e.g., "phase line green"). QuickSet creates the appropriate military icon on its map, and sends commands directly to ExInit. To illustrate the use of QuickSet for ExInit, consider the example of Figure 2, in which, a user has said: "Multiple boundaries," followed in rapid succession by a series of multimodal utterances such as "Battalion <draws line>," and "Company <draws line>." The "multiple" utterance tells QuickSet that subsequent input is to be interpreted as a boundary line, if possible. Multimodal input that

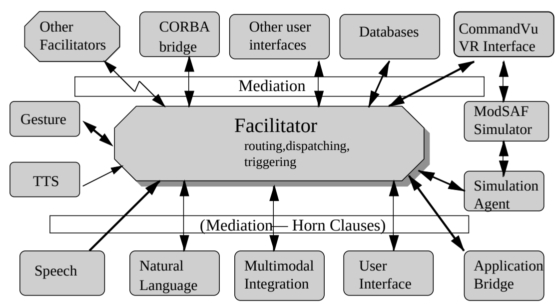

Figure 1: The agents are connected to a facilitator that routes queries to agents capable of resolving them.

both names an echelon and draws a line are then interpreted as boundaries of the appropriate echelon, and are echoed on the map appropriately. Numerous features describing engineering works, such as a fortified line, a berm, minefields, etc. have also been added to the map using speech and gesture. The user has created a number of armored companies facing 45 degrees in defensive posture; he is now beginning to add armored companies facing 225 degrees, etc.

QuickSet can employ multiple speech recognizers, including IBM's Voice Type 3.0 and Voice Type Application Factory (VTAF), as well as Microsoft's Whisper or any SAPI-compliant recognizer. For this case study, IBM's Voice Type Application Factory was used with a bigram grammar and 629 word vocabulary. VTAF produces a single recognition hypothesis.

2. PROCEDURE

The user was a retired Major in the US Marine Corps, author of numerous text books on military planning, command-and-control, and tactics. The subject was given 30 minutes to learn the ExInit GUI, and the same amount of time to learn QuickSet. He had used neither system before. The subject created a scenario of his own choosing first on paper, then with the Exinit GUI, and finally with QuickSet. The systems were run on a Pentium Pro (200MHz) computer with an 10" diagonal Wacom

PL300V, integrated color flat-panel display/digitizer with stylus input.

The scenario consisted of creating 15 control measures, and 6 units. The mean time needed to create each was calculated. The time to create an entity began at the time of the first movement towards a menu or object (for the GUI), or the time when the microphone was turned on by placing the pen on the map (for QuickSet). Creation time ended when the system asked for confirmation or disconfirmation of its impending action. With both systems, the user could enter a “mode” in which he was creating a particular kind of entity (e.g., a mechanized company). The time taken to enter the mode was amortized over the number of entities created in that mode. Separate entity creation calculations were made for units and control measures because the GUI employs a different user interface tool for each of them. Creation times include correction of all user and system errors for both QuickSet and the GUI.

3. RESULTS

Multimodal interaction resulted in an 8.7-fold speed increase in creating units compared to the GUI, and a 3.2-fold increase in creating control measures (see Table I). Much of the substantial

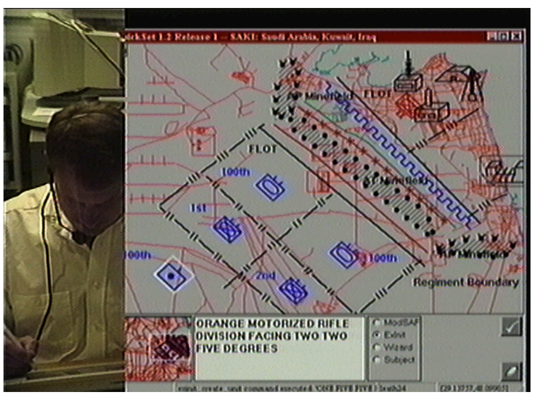

Figure 3: QuickSet being used for force laydown

speed differential can be traced to the need to browse the echelons of the US military, and scroll long lists of units using the GUI (e.g., 126 units are in the list of US Army companies), followed by a separate dragging operation to position the selected unit. In contrast, a QuickSet user can specify the type of entity (without scrolling) in parallel with specifying its location.

The QuickSet multimodal success rate (i.e., the percentage of commands that resulted in the correct item being presented for confirmation by the user) was 68%. This relatively low recognition rate occurred because of a single control measure (an “antitank ditch”) that the user wanted to create 6 times, requiring 15 attempts. Poor speech recognition of “anti” resulted in half the errors, and poor gesture recognition of very short lines as points accounted for the other half of the errors.[1] Note that the user was able to make 5 attempts with QuickSet in less time than it took to create that control measure using the GUI. The multimodal success rate for the other control measures and units was 100%.

ENTITY TYPE	QuickSet	ExInit GUI
Unit (n = 6)	3.0 secs.	26.0 secs.
Control measure (n = 15)	6.5 secs.	20.5 secs.

Table I. Mean times to create entities with QuickSet and ExInit’s graphical user interface

Not only was multimodal interaction substantially more efficient, it was strongly preferred. Comments from the user include: "The speech and gestures required were very natural and intuitive, as was the combination between them. The click-and-drag program, by comparison, was more difficult to work with. The requirement to go through various menus to emplace a unit or control measure was unwieldy."

4. DISCUSSION AND CONCLUSIONS

This case study suggests there may be substantial speed and efficiency advantages of multimodal interaction over direct manipulation-based graphical user interfaces for map-based tasks. Unlike prior research in which expected speed advantages were washed out by error correction, the strong advantages of multimodal interaction hold in spite of a 68% multimodal success rate, including the required error correction. In the future, a more comprehensive study will compare ExInit with a new version of QuickSet, which is known to be substantially more capable than the one used here (for example, the gesture recognition error rate has been reduced by 55%.).

ACKNOWLEDGEMENTS

This work was supported in part by the Information Technology and Information Systems offices of the Defense Advanced Research Projects Agency under contract number DABT63-95-C-007, in part by ONR grant number N00014-95-1-1164, and has been done in collaboration with the US Navy’s SPAWAR Systems Center, ATI Incorporated, Ascent Technologies, and MRJ Corp. Many thanks to Jay Pittman for the ExInit integration, to Liang Chen for graphics and military symbology, and to our test subject.

REFERENCES

1. A. Chapanis, R. B. Ochsman, R. N. Parrish, and G. D. Weeks, “Studies in interactive communication: I. The effects of four communication modes on the behavior of teams during cooperative problem solving,” *Human Factors*, vol. 14, pp. 487-509, 1972.

2. A. Chapanis, R. N. Parrish, R. B. Ochsman, and G. D. Weeks, “Studies in interactive communication: II. The

[1] If one simply ignored the antitank ditch creation in both conditions, the speed increase would have been 3.75 fold.

effects of four communication modes on the linguistic performance of teams during cooperative problem solving," *Human Factors*, vol. 19, pp. 101-125, 1977.

3. P. R. Cohen and S. L. Oviatt, "The Role of Voice Input for Human-Machine Communication," *Proceedings of the National Academy of Sciences*, *92(22)*, pp. 9921-9927 1995.

4. M. Helander, T. S. Moody, and M. G. Joost, "Systems Design for Automated Speech Recognition," in *Handbook of Human-Computer Interaction*, M. Helander, Ed. New York: North-Holland, 1990.

5. J. P. Marshall, "A manufacturing application of voice recognition for assembly of aircraft wire harnesses," in *Proceedings of Speech Tech/Voice Systems Worldwide*. New York, 1992.

6. G. L. Martin, "The utility of speech input in user-computer interfaces," *International Journal of Man-machine Studies*, vol. 30, pp. 355-375, 1989.

7. D. Visick, P. Johnson, and J. Long, "The use of simple speech recognisers in industrial applications," in *Proceedings of INTERACT'84, First IFIP conference on Human-Computer Interaction*. London: International Federation of Information Processing Societies, 1984.

8. A. G. Hauptmann and A. I. Rudnicky, "A Comparison of speech and typed input," in *Proceedings of the Speech and Natural Language Workshop*. San Mateo, California, 1990, pp. 219-224.

9. A. I. Rudnicky, "Mode Preference in a simple data-retrieval task," in *DARPA Human Language Technology Workshop*. Princeton, New Jersey, 1993.

10. B. A. Mellor, C. Baber, and C. Tunley, "Evaluating Automatic Speech Recognition as a Component of a Multi-Input Human-Computer Interface," in *Proceedings of the International Conference on Spoken Language Processing*, 1996.

11. H. Murveit, J. Butzberger, and M. Weintraub, "Speech Recognition in SRI's Resource Management and ATIS Systems," in *Proceedings of the 4th DARPA Workshop on Speech and Natural Language*. Asilomar, California, 1991.

12. V. Zue, J. Glass, D. Goddeau, D. Goodine, L. Hirschman, M. Phillips, J. Polifroni, and S. Seneff, "The MIT ATIS System: February 1992 Progress Report," in *Fifth DARPA Workshop on Speech and Natural Language*. San Mateo, Calif., 1992.

13. S. L. Oviatt, "Multimodal interactive maps: Designing for human performance," *Human Computer Interaction*, vol. 12, pp. 93-129, 1997.

14. B. Shneiderman, "Direct Manipulation: A Step Beyond Programming Languages," *IEEE Computer*, vol. 16, pp. 57-69, 1983.

15. P. R. Cohen, M. Johnston, D. McGee, S. Oviatt, J. Pittman, I. Smith, L. Chen, and J. Clow, "QuickSet: Multimodal interaction for distributed applications," in *Proceeedings of the Fifth ACM International Multmedia Conference*, E. Glinert, Ed. New York: ACM Press, 1997, pp. 31-40.

16. P. R. Cohen, M. Johnston, D. McGee, S. Oviatt, J. Pittman, I. Smith, L. Chen, and J. Clow, "Multimodal interaction for distributed interactive simulation," in *Readings in Intelligent User Interfaces*, M. Maybury, and W. Wahlster, Ed. San Franciso, Calif.: Morgan Kaufmann Publishers, 1998 Original publication appeared in *the Proceedings of the Ninth Conference on Innovative Applications of Artificial Intelligence, AAAI Press, Menlo Park, Calif., 1997, 978-985*.

17. A. J. Courtemanche and A. Ceranowicz, "ModSAF Development Status," in *Proceedings of the Fifth Conference on Computer Generated Forces and Behavioral Representation*. Orlando: University of Central Florida, 1995, pp. 3-13.

18. S. L. Oviatt, "Ten Myths of Multimodal Interaction," *Communications of the ACM*, in press.

19. S. L. Oviatt, "Pen/Voice: Complementary Multimodal Communication," in *Proceedings of Speech Tech'92*. New York, 1992, pp. 238-241.

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

21. M. Johnston, P. R. Cohen, D. McGee, S. L. Oviatt, J. A. Pittman, I. Smith., "Unification-based multimodal integration.," presented at Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics, 1997.

22. D. McGee, P. R. Cohen, and S. L. Oviatt, "Confirmation in Multimodal Systems," in *Proceedings of Coling-ACL '98*. Montreal: Association for Computational Linguistics, 1998.

23. P. R. Cohen, A. Cheyer, M. Q. Wang, and S. C. Baeg, "An Open Agent Architecture," in *Working notes of the AAAI Spring Symposium Series on Software Agents*. Stanford, Calif.: AAAI Press, 1994c.