

Unification-based Multimodal Integration

Michael Johnston, Philip R. Cohen, David McGee,
Sharon L. Oviatt, James A. Pittman, Ira Smith

Center for Human Computer Communication
Department of Computer Science and Engineering
Oregon Graduate Institute, PO BOX 91000, Portland, OR 97291, USA.
{johnston,pcohen,dmcgee,oviatt,jay,ira}@cse.ogi.edu

Abstract

Recent empirical research has shown conclusive advantages of multimodal interaction over speech-only interaction for map-based tasks. This paper describes a multimodal language processing architecture which supports interfaces allowing simultaneous input from speech and gesture recognition. Integration of spoken and gestural input is driven by unification of typed feature structures representing the semantic contributions of the different modes. This integration method allows the component modalities to mutually compensate for each others' errors. It is implemented in Quick-Set, a multimodal (pen/voice) system that enables users to set up and control distributed interactive simulations.

1 Introduction

By providing a number of channels through which information may pass between user and computer, multimodal interfaces promise to significantly increase the bandwidth and fluidity of the interface between humans and machines. In this work, we are concerned with the addition of multimodal input to the interface. In particular, we focus on interfaces which support simultaneous input from speech and pen, utilizing speech recognition and recognition of gestures and drawings made with a pen on a complex visual display, such as a map.

Our focus on multimodal interfaces is motivated, in part, by the trend toward portable computing devices for which complex graphical user interfaces are infeasible. For such devices, speech and gesture will be the primary means of user input. Recent empirical results (Oviatt 1996) demonstrate clear task performance and user preference advantages for multimodal interfaces over speech only interfaces, in par-

ticular for spatial tasks such as those involving maps. Specifically, in a within-subject experiment during which the same users performed the same tasks in various conditions using only speech, only pen, or both speech and pen-based input, users' multimodal input to maps resulted in 10% faster task completion time, 23% fewer words, 35% fewer spoken disfluencies, and 36% fewer task errors compared to unimodal spoken input. Of the user errors, 48% involved location errors on the map—errors that were nearly eliminated by the simple ability to use pen-based input. Finally, 100% of users indicated a preference for multimodal interaction over speech-only interaction with maps. These results indicate that for map-based tasks, users would both perform better and be more satisfied when using a multimodal interface. As an illustrative example, in the distributed simulation application we describe in this paper, one user task is to add a “phase line” to a map. In the existing unimodal interface for this application (CommandTalk, Moore 1997), this is accomplished with a spoken utterance such as ‘CREATE A LINE FROM COORDINATES NINE FOUR THREE NINE THREE ONE TO NINE EIGHT NINE NINE FIVE ZERO AND CALL IT PHASE LINE GREEN’. In contrast the same task can be accomplished by saying ‘PHASE LINE GREEN’ and simultaneously drawing the gesture in Figure 1.

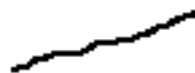


Figure 1: Line gesture

The multimodal command involves speech recognition of only a three word phrase, while the equivalent unimodal speech command involves recognition of a complex twenty four word expression. Furthermore, using unimodal speech to indicate more com-

plex spatial features such as routes and areas is practically infeasible if accuracy of shape is important.

Another significant advantage of multimodal over unimodal speech is that it allows the user to switch modes when environmental noise or security concerns make speech an unacceptable input medium, or for avoiding and repairing recognition errors (Oviatt and Van Gent 1996). Multimodality also offers the potential for input modes to mutually compensate for each others' errors. We will demonstrate how, in our system, multimodal integration allows speech input to compensate for errors in gesture recognition and vice versa.

Systems capable of integration of speech and gesture have existed since the early 80's. One of the first such systems was the "Put-That-There" system (Bolt 1980). However, in the sixteen years since then, research on multimodal integration has not yielded a reusable scalable architecture for the construction of multimodal systems that integrate gesture and voice. There are four major limiting factors in previous approaches to multimodal integration:

- (i) The majority of approaches limit the bandwidth of the gestural mode to simple deictic pointing gestures made with a mouse (Neal and Shapiro 1991, Cohen 1991, Cohen 1992, Brison and Vigouroux (ms.), Wauchope 1994) or with the hand (Koons et al 1993¹).
- (ii) Most previous approaches have been primarily speech-driven², treating gesture as a secondary dependent mode (Neal and Shapiro 1991, Cohen 1991, Cohen 1992, Brison and Vigouroux (ms.), Koons et al 1993, Wauchope 1994). In these systems, integration of gesture is triggered by the appearance of expressions in the speech stream whose reference needs to be resolved, such as definite and deictic noun phrases (e.g. 'this one', 'the red cube').
- (iii) None of the existing approaches provide a well-understood generally applicable common meaning representation for the different modes, or,
- (iv) A general and formally-well defined mechanism for multimodal integration.

¹Koons et al 1993 describe two different systems. The first uses input from hand gestures and eye gaze in order to aid in determining the reference of noun phrases in the speech stream. The second allows users to manipulate objects in a blocks world using iconic and pantomimic gestures in addition to deictic gestures.

²More precisely, they are 'verbal language'-driven. Either spoken or typed linguistic expressions are the driving force of interpretation.

We present an approach to multimodal integration which overcomes these limiting factors. A wide base of continuous gestural input is supported and integration may be driven by either mode. Typed feature structures (Carpenter 1992) are used to provide a clearly defined and well understood common meaning representation for the modes, and multimodal integration is accomplished through unification.

2 Quickset: A Multimodal Interface for Distributed Interactive Simulation

The initial application of our multimodal interface architecture has been in the development of the QuickSet system, an interface for setting up and interacting with distributed interactive simulations. QuickSet provides a portal into LeatherNet³, a simulation system used for the training of US Marine Corps platoon leaders. LeatherNet simulates training exercises using the ModSAF simulator (Courtemanche and Ceranowicz 1995) and supports 3D visualization of the simulated exercises using CommandVu (Clarkson and Yi 1996). SRI International's CommandTalk provides a unimodal spoken interface to LeatherNet (Moore et al 1997).

QuickSet is a distributed system consisting of a collection of agents that communicate through the Open Agent Architecture⁴ (Cohen et al 1994). It runs on both desktop and hand-held PCs under Windows 95, communicating over wired and wireless LANs (respectively), or modem links. The wireless hand-held unit is a 3-lb Fujitsu Stylistic 1000 (Figure 2). We have also developed a Java-based QuickSet agent that provides a portal to the simulation over the World Wide Web. The QuickSet user interface displays a map of the terrain on which the simulated military exercise is to take place (Figure 2). The user can gesture and draw directly on the map with the pen and simultaneously issue spoken commands. Units and objectives can be laid down on the map by speaking their name and gesturing on the desired location. The map can also be annotated with line features such as barbed wire and fortified lines, and area features such as minefields and landing zones. These are created by drawing the appropriate spatial feature on the map and speak-

³LeatherNet is currently being developed by the Naval Command, Control and Ocean Surveillance Center (NCCOSC) Research, Development, Test and Evaluation Division (NRaD) in coordination with a number of contractors.

⁴Open Agent Architecture is a trademark of SRI International.



Figure 2: The QuickSet user interface

ing its name. Units, objectives, and lines can also be generated using unimodal gestures by drawing their map symbols in the desired location. Orders can be assigned to units, for example, in Figure 2 an M1A1 platoon on the bottom left has been assigned a route to follow. This order is created multimodally by drawing the curved route and saying ‘WHISKEY FOUR SIX FOLLOW THIS ROUTE’. As entities are created and assigned orders they are displayed on the UI and automatically instantiated in a simulation database maintained by the ModSAF simulator.

Speech recognition operates in either a click-to-speak mode, in which the microphone is activated when the pen is placed on the screen, or open microphone mode. The speech recognition agent is built using a continuous speaker-independent recognizer commercially available from IBM.

When the user draws or gestures on the map, the resulting electronic ‘ink’ is passed to a gesture recognition agent, which utilizes both a neural network and a set of hidden Markov models. The ink is size-normalized, centered in a 2D image, and fed into the neural network as pixels, as well as being smoothed, resampled, converted to deltas, and fed to the HMM recognizer. The gesture recognizer currently recog-

nizes a total of twenty six different gestures, some of which are illustrated in Figure 3. They include various military map symbols such as platoon, mortar, and fortified line, editing gestures such as deletion, and spatial features such as routes and areas.

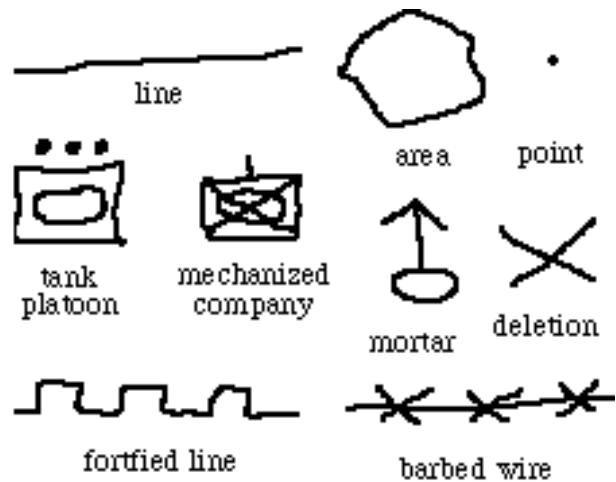


Figure 3: Example symbols and gestures

As with all recognition technologies, gesture recognition may result in errors. One of the factors

contributing to this is that routes and areas do not have signature shapes that can be used to identify them and are frequently confused (Figure 4).

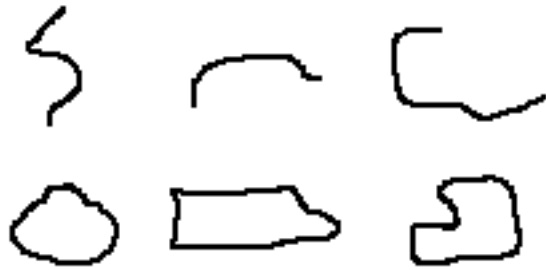


Figure 4: Pen drawings of routes and areas

Another contributing factor is that users' pen input is often sloppy (Figure 5) and map symbols can be confused among themselves and with route and area gestures.

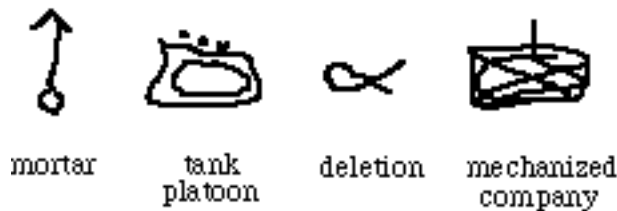


Figure 5: Typical pen input from real users

Given the potential for error, the gesture recognizer issues not just a single interpretation, but a series of potential interpretations ranked with respect to probability. The correct interpretation is frequently determined as a result of multimodal integration, as illustrated below⁵.

3 A Unification-based Architecture for Multimodal Integration

One of the most significant challenges facing the development of effective multimodal interfaces concerns the integration of input from different modes. Input signals from each of the modes can be assigned meanings. The problem is to work out how to combine the meanings contributed by each of the modes in order to determine what the user actually intends to communicate.

To model this integration, we utilize a unification operation over typed feature structures (Carpenter 1990, 1992, Pollard and Sag 1987, Calder 1987, King

⁵See Wahlster 1991 for discussion of the role of dialog in resolving ambiguous gestures.

1989, Moshier 1988). Unification is an operation that determines the consistency of two pieces of partial information, and if they are consistent combines them into a single result. As such, it is ideally suited to the task at hand, in which we want to determine whether a given piece of gestural input is compatible with a given piece of spoken input, and if they are compatible, to combine the two inputs into a single result that can be interpreted by the system.

The use of feature structures as a semantic representation framework facilitates the specification of partial meanings. Spoken or gestural input which partially specifies a command can be represented as an underspecified feature structure in which certain features are not instantiated. The adoption of typed feature structures facilitates the statement of constraints on integration. For example, if a given speech input can be integrated with a line gesture, it can be assigned a feature structure with an underspecified location feature whose value is required to be of type *line*.

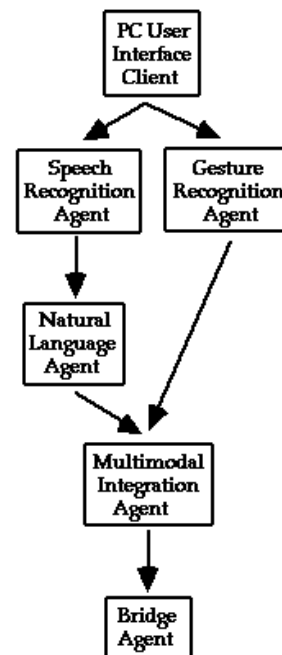


Figure 6: Multimodal integration architecture

Figure 6 presents the main agents involved in the QuickSet system. Spoken and gestural input originates in the user interface client agent and it is passed on to the speech recognition and gesture recognition agents respectively. The natural language agent uses a parser implemented in Prolog to parse strings that originate from the speech recognition agent and assign typed feature structures to

them. The potential interpretations of gesture from the gesture recognition agent are also represented as typed feature structures. The multimodal integration agent determines and ranks potential unifications of spoken and gestural input and issues complete commands to the bridge agent. The bridge agent accepts commands in the form of typed feature structures and translates them into commands for whichever applications the system is providing an interface to.

For example, if the user utters ‘M1A1 PLATOON’, the name of a particular type of tank platoon, the natural language agent assigns this phrase the feature structure in Figure 7. The type of each feature structure is indicated in italics at its bottom right or left corner.

$$create_unit \left[\begin{array}{l} \text{object :} \\ \text{location :} \end{array} \left[\begin{array}{l} \text{type : m1a1} \\ \text{echelon : platoon} \\ \text{]}_{unit} \\ \text{]}_{point} \end{array} \right]$$

Figure 7: Feature structure for ‘M1A1 PLATOON’

Since QuickSet is a task-based system directed toward setting up a scenario for simulation, this phrase is interpreted as a partially specified unit creation command. Before it can be executed, it needs a location feature indicating where to create the unit, which is provided by the user’s gesturing on the screen. The user’s ink is likely to be assigned a number of interpretations, for example, both a point interpretation and a line interpretation, which the gesture recognition agent assigns typed feature structures (see Figures 8 and 9). Interpretations of gestures as location features are assigned a general *command* type which unifies with all of commands taken by the system.

$$command \left[\begin{array}{l} \text{location :} \end{array} \left[\begin{array}{l} \text{xcoord : 95305} \\ \text{xcoord : 94365} \end{array} \right]_{point} \right]$$

Figure 8: Point interpretation of gesture

$$command \left[\begin{array}{l} \text{location :} \end{array} \left[\begin{array}{l} \text{coordlist :} \\ \text{[(95301, 94360),} \\ \text{(95305, 94365),} \\ \text{(95310, 94380)]} \end{array} \right]_{line} \right]$$

Figure 9: Line interpretation of gesture

The task of the integrator agent is to field incoming typed feature structures representing interpretations of speech and of gesture, identify the best potential interpretation, multimodal or unimodal, and

issue a typed feature structure representing the preferred interpretation to the bridge agent, which will execute the command. This involves parsing of the speech and gesture streams in order to determine potential multimodal integrations. Two factors guide this: tagging of speech and gesture as either complete or partial and examination of time stamps associated with speech and gesture.

Speech or gesture input is marked as complete if it provides a full command specification and therefore does not need to be integrated with another mode. Speech or gesture marked as partial needs to be integrated with another mode in order to derive an executable command.

Empirical study of the nature of multimodal interaction has shown that speech typically follows gesture within a window of a three to four seconds while gesture following speech is very uncommon (Oviatt et al 97). Therefore, in our multimodal architecture, the integrator temporally licenses integration of speech and gesture if their time intervals overlap, or if the onset of the speech signal is within a brief time window following the end of gesture. Speech and gesture are integrated appropriately even if the integrator agent receives them in a different order from their actual order of occurrence. If speech is temporally compatible with gesture, in this respect, then the integrator takes the sets of interpretations for both speech and gesture, and for each pairing in the product set attempts to unify the two feature structures. The probability of each multimodal interpretation in the resulting set licensed by unification is determined by multiplying the probabilities assigned to the speech and gesture interpretations.

In the example case above, both speech and gesture have only partial interpretations, one for speech, and two for gesture. Since the speech interpretation (Figure 7) requires its location feature to be of type *point*, only unification with the point interpretation of the gesture will succeed and be passed on as a valid multimodal interpretation (Figure 10).

$$create_unit \left[\begin{array}{l} \text{object :} \\ \text{location :} \end{array} \left[\begin{array}{l} \text{type : m1a1} \\ \text{echelon : platoon} \\ \text{]}_{unit} \\ \text{ [xcoord : 95305} \\ \text{ xcoord : 94365} \end{array} \right]_{point} \right]$$

Figure 10: Multimodal interpretation

The ambiguity of interpretation of the gesture was resolved by integration with speech which in this case required a location feature of type *point*. If the spoken command had instead been ‘BARBED

WIRE’ it would have been assigned the feature structure in Figure 11. This structure would only unify with the line interpretation of gesture resulting in the interpretation in Figure 12.

$$\text{create_line} \left[\begin{array}{l} \text{object :} \\ \text{location :} \end{array} \left[\begin{array}{l} \text{style : barbed_wire} \\ \text{color : red} \\ \text{ } \\ \text{ } \end{array} \right]_{\text{line_obj}} \right]_{\text{line}}$$

Figure 11: Feature structure for ‘BARBED WIRE’

$$\text{create_line} \left[\begin{array}{l} \text{object :} \\ \text{location :} \end{array} \left[\begin{array}{l} \text{style : barbed_wire} \\ \text{color : red} \\ \text{coordlist :} \\ \text{[(95301, 94360),} \\ \text{(95305, 94365),} \\ \text{(95310, 94380)]} \end{array} \right]_{\text{line_obj}} \right]_{\text{line}}$$

Figure 12: Multimodal line creation

Similarly, if the spoken command described an area, for example an ‘ANTI TANK MINEFIELD’, it would only unify with an interpretation of gesture as an area designation. In each case the unification-based integration strategy compensates for errors in gesture recognition through type constraints on the values of features.

Gesture also compensates for errors in speech recognition. In the open microphone mode, where the user does not have to gesture in order to speak, spurious speech recognition errors are more common than with click-to-speak, but are frequently rejected by the system because of the absence of a compatible gesture for integration. For example, if the system spuriously recognizes ‘M1A1 PLATOON’, but there is no overlapping or immediately preceding gesture to provide the location, the speech will be ignored. The architecture also supports selection among n-best speech recognition results on the basis of the preferred gesture recognition. In the future, n-best recognition results will be available from the recognizer, and we will further examine the potential for gesture to help select among speech recognition alternatives.

Since speech may follow gesture, and since even simultaneously produced speech and gesture are processed sequentially, the integrator cannot execute what appears to be a complete unimodal command on receiving it, in case it is immediately followed by input from the other mode suggesting a multimodal interpretation. If a given speech or gesture input has a set of interpretations including both partial

and complete interpretations, the integrator agent waits for an incoming signal from the other mode. If no signal is forthcoming from the other mode within the time window, or if interpretations from the other mode do not integrate with any interpretations in the set, then the best of the complete unimodal interpretations from the original set is sent to the bridge agent.

For example, the gesture in Figure 13 is used for unimodal specification of the location of a fortified line. If recognition is successful the gesture agent would assign the gesture an interpretation like that in Figure 14.



Figure 13: Fortified line gesture

$$\text{create_line} \left[\begin{array}{l} \text{object :} \\ \text{location :} \end{array} \left[\begin{array}{l} \text{style : fortified_line} \\ \text{color : blue} \\ \text{coordlist :} \\ \text{[(93000, 94360),} \\ \text{(93025, 94365),} \\ \text{...} \\ \text{(93112, 94362)]} \end{array} \right]_{\text{line_obj}} \right]_{\text{line}}$$

Figure 14: Unimodal fortified line feature structure

However, it might also receive an additional potential interpretation as a location feature of a more general line type (Figure 15).

$$\text{command} \left[\begin{array}{l} \text{location :} \end{array} \left[\begin{array}{l} \text{coordlist :} \\ \text{[(93000, 94360),} \\ \text{(93025, 94365),} \\ \text{...} \\ \text{(93112, 94362)]} \end{array} \right]_{\text{line}} \right]$$

Figure 15: Line feature structure

On receiving this set of interpretations, the integrator cannot immediately execute the complete interpretation to create a fortified line, even if it is assigned the highest probability by the recognizer, since speech contradicting this may immediately follow. For example, if overlapping with or just after the gesture, the user said ‘BARBED WIRE’ then the line feature interpretation would be preferred. If speech does not follow within the three to four second window, or following speech does not integrate with the gesture, then the unimodal interpretation

is chosen. This approach embodies a preference for multimodal interpretations over unimodal ones, motivated by the possibility of unintended complete unimodal interpretations of gestures. After more detailed empirical investigation, this will be refined so that the possibility of integration weighs in favor of the multimodal interpretation, but it can still be beaten by a unimodal gestural interpretation with a significantly higher probability.

4 Conclusion

We have presented an architecture for multimodal interfaces in which integration of speech and gesture is mediated and constrained by a unification operation over typed feature structures. Our approach supports a full spectrum of gestural input, not just deixis. It also can be driven by either mode and enables a wide and flexible range of interactions. Complete commands can originate in a single mode yielding unimodal spoken and gestural commands, or in a combination of modes yielding multimodal commands, in which speech and gesture are able to contribute either the predicate or the arguments of the command. This architecture allows the modes to synergistically mutual compensate for each others' errors. We have informally observed that integration with speech does succeed in resolving ambiguous gestures. In the majority of cases, gestures will have multiple interpretations, but this is rarely apparent to the user, because the erroneous interpretations of gesture are screened out by the unification process. We have also observed that in the open microphone mode multimodality allows erroneous speech recognition results to be screened out. For the application tasks described here, we have observed a reduction in the length and complexity of spoken input, compared to the unimodal spoken interface to LeatherNet, informally reconfirming the empirical results of Oviatt et al 1997. For this family of applications at least, it appears to be the case that as part of a multimodal architecture, current speech recognition technology is sufficiently robust to support easy-to-use interfaces.

Vo and Wood 1996 present an approach to multimodal integration similar in spirit to that presented here in that it accepts a variety of gestures and is not solely speech-driven. However, we believe that unification of typed feature structures provides a more general, formally well-understood, and reusable mechanism for multimodal integration than the frame merging strategy that they describe. Cheyer and Julia (1995) sketch a system based on Oviatt's (1996) results but describe neither the integration strategy nor multimodal compensation.

QuickSet has undergone a form of pro-active evaluation in that its design is informed by detailed predictive modeling of how users interact multimodally and it incorporates the results of existing empirical studies of multimodal interaction (Oviatt 1996, Oviatt et al 1997). It has also undergone participatory design and user testing with the US Marine Corps at their training base at 29 Palms, California, with the US Army at the Royal Dragon exercise at Fort Bragg, North Carolina, and as part of the Command Center of the Future at NRaD.

Our initial application of this architecture has been to map-based tasks such as distributed simulation. It supports a fully-implemented usable system in which hundreds of different kinds of entities can be created and manipulated. We believe that the unification-based method described here will readily scale to larger tasks and is sufficiently general to support a wide variety of other application areas, including graphically-based information systems and editing of textual and graphical content. The architecture has already been successfully re-deployed in the construction of multimodal interface to health care information.

We are actively pursuing incorporation of statistically-derived heuristics and a more sophisticated dialogue model into the integration architecture. We are also developing a capability for automatic logging of spoken and gestural input in order to collect more fine-grained empirical data on the nature of multimodal interaction.

5 Acknowledgments

This work is supported in part by the Information Technology and Information Systems offices of DARPA 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 NCCOSC RDT&E Division (NRaD), Ascent Technologies, Mitre Corp., MRJ Corp., and SRI International.

References

- Bolt, R. A., 1980. "Put-That-There": Voice and gesture at the graphics interface. *Computer Graphics*, 14.3:262-270.
- Brison, E., and N. Vigouroux. (unpublished ms.). Multimodal references: A generic fusion process. URIT-URA CNRS. Universit Paul Sabatier, Toulouse, France.
- Calder, J. 1987. Typed unification for natural language processing. In E. Klein and J. van Benthem,

- editors, *Categories, Polymorphisms, and Unification*, pages 65–72. Centre for Cognitive Science, University of Edinburgh, Edinburgh.
- Carpenter, R. 1990. Typed feature structures: Inheritance, (In)equality, and Extensionality. In W. Daelemans and G. Gazdar, editors, *Proceedings of the ITK Workshop: Inheritance in Natural Language Processing*, pages 9–18, Tilburg. Institute for Language Technology and Artificial Intelligence, Tilburg University, Tilburg.
- Carpenter, R. 1992. *The logic of typed feature structures*. Cambridge University Press, Cambridge, England.
- Cheyer, A., and L. Julia. 1995. Multimodal maps: An agent-based approach. In *International Conference on Cooperative Multimodal Communication (CMC/95)*, pages 24–26, May 1995. Eindhoven, The Netherlands.
- Clarkson, J. D., and J. Yi. 1996. LeatherNet: A synthetic forces tactical training system for the USMC commander. In *Proceedings of the Sixth Conference on Computer Generated Forces and Behavioral Representation*, pages 275–281. Institute for simulation and training. Technical Report IST-TR-96-18.
- Cohen, P. R. 1991. Integrated interfaces for decision support with simulation. In B. Nelson, W. D. Kelton, and G. M. Clark, editors, *Proceedings of the Winter Simulation Conference*, pages 1066–1072. ACM, New York.
- Cohen, P. R. 1992. The role of natural language in a multimodal interface. In *Proceedings of UIST'92*, pages 143–149. ACM Press, New York.
- Cohen, P. R., A. Cheyer, M. Wang, and S. C. Baeg. 1994. An open agent architecture. In *Working Notes of the AAAI Spring Symposium on Software Agents (March 21-22, Stanford University, Stanford, California)*, pages 1–8.
- Courtemanche, A. J., and A. Ceranowicz. 1995. ModSAF development status. In *Proceedings of the Fifth Conference on Computer Generated Forces and Behavioral Representation*, pages 3–13, May 9–11, Orlando, Florida. University of Central Florida, Florida.
- King, P. 1989. *A logical formalism for head-driven phrase structure grammar*. Ph.D. Thesis, University of Manchester, Manchester, England.
- Koons, D. B., C. J. Sparrell, and K. R. Thorisson. 1993. Integrating simultaneous input from speech, gaze, and hand gestures. In M. T. Maybury, editor, *Intelligent Multimedia Interfaces*, pages 257–276. AAAI Press/ MIT Press, Cambridge, Massachusetts.
- Moore, R. C., J. Dowding, H. Bratt, J. M. Gawron, Y. Gorf, and A. Cheyer. 1997. CommandTalk: A Spoken-Language Interface for Battlefield Simulations. In *Proceedings of Fifth Conference on Applied Natural Language Processing*, pages 1–7, Washington, D.C. Association for Computational Linguistics, Morristown, New Jersey.
- Moshier, D. 1988. *Extensions to unification grammar for the description of programming languages*. Ph.D. Thesis, University of Michigan, Ann Arbor, Michigan.
- Neal, J. G., and S. C. Shapiro. 1991. Intelligent multi-media interface technology. In J. W. Sullivan and S. W. Tyler, editors, *Intelligent User Interfaces*, pages 45–68. ACM Press, Frontier Series, Addison Wesley Publishing Co., New York, New York.
- Oviatt, S. L. 1996. Multimodal interfaces for dynamic interactive maps. In *Proceedings of Conference on Human Factors in Computing Systems: CHI '96*, pages 95–102, Vancouver, Canada. ACM Press, New York.
- Oviatt, S. L., A. DeAngeli, and K. Kuhn. 1997. Integration and synchronization of input modes during multimodal human-computer interaction. In *Proceedings of the Conference on Human Factors in Computing Systems: CHI '97*, pages 415–422, Atlanta, Georgia. ACM Press, New York.
- Oviatt, S. L., and R. van Gent. 1996. Error resolution during multimodal human-computer interaction. In *Proceedings of International Conference on Spoken Language Processing*, vol 1, pages 204–207, Philadelphia, Pennsylvania.
- Pollard, C. J., and I. A. Sag. 1987. *Information-based syntax and semantics: Volume I, Fundamentals.*, Volume 13 of CSLI Lecture Notes. Center for the Study of Language and Information, Stanford University, Stanford, California.
- Vo, M. T., and C. Wood. 1996. Building an application framework for speech and pen input integration in multimodal learning interfaces. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, Atlanta, GA.
- Wahlster, W. 1991. User and discourse models for multimodal communication. In J. Sullivan and S. Tyler, editors, *Intelligent User Interfaces*, ACM Press, Addison Wesley Publishing Co., New York, New York.
- Wauchope, K. 1994. *Eucalyptus: Integrating natural language input with a graphical user interface*. Naval Research Laboratory, Report NRL/FR/5510-94-9711.