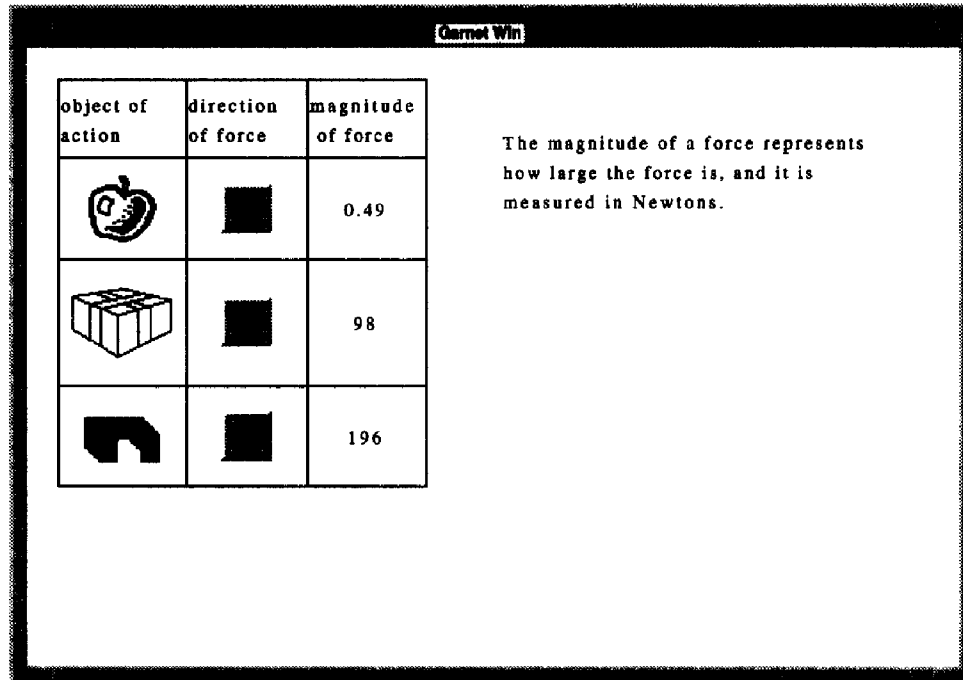


Figure 18. Final multimodal presentation: icon of correct size.



as $segment_1$. Thus, Find-proposals adds another action to this option, namely $(enlarge\ width(segment_3))$. The ranges that remain after the deletion of the ranges corresponding to $width(segment_3)$ unify. On completion of its process, the proposals generated by Find-proposals are:

- { $(reduce\ width(segment_2))$ }
- { $(enlarge\ width(segment_1))$ } $(enlarge\ width(segment_3))$ }.

For the first proposal, the table agent sets $width(column_2)$ to 50, which satisfies all the preferred constraints (viz., scn_5 , scn_6 , and scn_9). However, in this case, the required constraint scn_4 is violated. To satisfy this constraint, the table agent and the icon agent presenting the right-arrow icon engage in a negotiation process where the table agent asks the icon agent to reduce $width(segment_2)$ to fit the new column width. Upon receiving an OK-event, the plan is improved because all the required constraints are satisfied, as well as additional preferred constraints. An improved presentation of Figure 14 is shown in Figure 18, where the big right-arrow icon has been reduced. Note that in addition to the adjustment of $width(column_2)$ to satisfy additional preferred constraints, the width of each column has been adjusted to satisfy the minimum requirement for presenting a column heading (i.e., the width of the column must fit the longest word in a column heading). If the icon agent had been unable to reduce the right-arrow icon, the table agent would have dropped this proposal and recovered the previous value of $width(column_2)$. If time permitted, the table agent would

have attempted the second proposal, which also would have failed due to the unavailability of larger up-arrow icons in the icon library.

This procedure does not always produce a better plan because it may result in the violation of previously satisfied constraints. In addition to the constraints that pertain to the width of columns, there are similar constraints that affect the height of rows. When an agent enlarges or reduces a segment to satisfy a preferred width constraint, a height constraint may be violated. As seen in Section 5.1, such a situation may be encountered when the table agent asks an icon agent to enlarge or reduce the width of an icon because, in this case, both the width and the height of the icon may be increased or decreased. In our example, the box icon in the first column cannot be reduced because it is the only icon available for a box. Thus, a preferred constraint that pertains to the height of the right-arrow icon is violated after this icon is reduced. When processing a proposal, MAGPIE considers each table column and row in turn, modifying entries so that additional preferred constraints are satisfied (even if another preferred constraint is violated as a result of a modification). On completion of these modifications, the table agent evaluates the resulting plan in terms of the number of preferred constraints that are satisfied. The new plan replaces the previous plan if it satisfies more preferred constraints. This process continues until it is time to display the table.

As negotiations over a variable may introduce a new negotiation process regarding another variable, the master agent must sort out the order in which variables are considered for constraint satisfaction to avoid endless negotiations with its server agents. The considerations applied by the table agent to achieve this goal are based on the constraint that demands that the same modality be used for all the entries in a column when a table is in Format (a), where each instantiation is presented in a row (see Section 4.1). As a result of this constraint, the segments in the same column of a table are generated by the same type of agent and are therefore more likely to be of uniform size than segments generated by different types of agents. Thus, the table agent adjusts the width of each column before adjusting the height of each row. When the table agent is trying to modify the width of a column, requests from its server agents to modify the height of a row are accepted if the constraints placed on the height of the table are satisfied. In contrast, when the table agent is trying to modify the height of a row, it refuses any request from a server agent to change the width of a column that has been processed.

7. RELATED RESEARCH

Several mechanisms have been used to address specific problems in multimodal presentation planning. These mechanisms are described as follows.

Syntactic and Semantic Analysis. Graphical languages were defined by Mackinlay (1986) and by Roth and Mattis (1991) to encode the syntactic and semantic properties of graphical presentations. These languages define techniques that can be used to express different semantic relations within the information to be presented. Some perceptual tasks are accomplished more accurately by one presentation technique than by others (e.g., using different lengths to convey the value of an attribute versus using different shapes). Thus, alternative designs can be evaluated by means of criteria that rank the different techniques based on the expressiveness and effectiveness of the presentation (Mackinlay, 1986). Although syntactic and semantic analysis has proved to be useful in selecting presentation techniques, the analysis is at a low level (e.g., characteristics of attributes or binary relations). It is not sufficient for perceptual tasks that contain composite information (e.g., an illustration of cause and effect).

Hierarchical Planning. Hierarchical planning is used for modality selection in several systems that design presentations during discourse planning. A hierarchical content planner is used by COMET (Feiner & McKeown, 1990) to refine a hierarchy of Logical Forms, which are used to represent a presentation plan. Communicative acts are used to represent a presentation plan in the Map Display system (Maybury, 1993) and in WIP (André et al., 1993). Because a complex act can be decomposed into a set of sub-acts, a hierarchical planning mechanism is applied in these systems to refine the communicative acts of a presentation plan. However, there may be several acts that are suitable for achieving a goal. To cope with the selection problem, the WIP system ranks these acts using criteria that take into account their effectiveness, side effects, and cost of execution. In contrast, Maybury (1993) considered the following factors: (a) the kind of communication being conducted, (b) the number and kind of entities visible in the region, and (c) their visual properties (e.g., size, color, shading). For example, the last two factors can be used to select acts that maximize the distinction between a given entity and its background.

Feature-Based Analysis. Modalities and information types were classified by Arens, Hovy, and Vossers (1993) according to their natural features and their ability to achieve particular communicative goals. For instance, urgent information may convey a warning. Thus, this type of information should be emphasized by techniques such as highlighting and blinking. The interdependencies among these features are described by a dependency network and modality allocation rules. Based on these rules, feature-based analysis can be applied to the intended information and the communicative goals to allocate suitable modalities for a presentation. However, this type of static analysis cannot cope with restrictions on resource consumption, which would not be available until run-time.

Constraint Satisfaction. Constraints are used to describe the syntactic, semantic, spatial, and temporal relations between presentation components in several multimodal presentation systems. In the COMET system (Feiner, Litman, McKeown, & Passonneau, 1993), Allen's (1983) temporal logic is employed to solve the temporal constraints between presentation components. In the WIP system (Graf, 1992; Rist & André, 1992), an incremental constraint hierarchy solver based on the DeltaBlue algorithm (Borning, Freeman-Benson, & Wilson, 1992) is used to solve the semantic and spatial constraints associated with layout formats. To refine a presentation plan, both systems evaluate the constraints that describe the preconditions of communicative acts. Thus, they incorporate constraint satisfaction into their planning mechanism during multimodal presentation planning. Because the constraints in MAGPIE are distributed in the presentation plan hierarchy, none of our agents can access all the constraints. Hence, these algorithms cannot be used to solve our constraint satisfaction problem.

MAGPIE uses unification and local constraint propagation algorithms to solve the constraint satisfaction problem. Our approach is similar to the multiagent simulated annealing approach described by Ghedira (1994) and the heuristic repair method described by Minton, Johnston, Philips, and Laird (1990). These approaches start with a configuration containing constraint violations and incrementally repair the violations until a consistent assignment is achieved. The multiagent simulated-annealing approach and our approach take advantage of multiagent systems to deal with the dynamic constraint satisfaction problem, where constraints can be added or deleted during the reasoning process. However, due to the hierarchical structure of MAGPIE, the communication between agents is simpler than the communication in Ghedira's system, as MAGPIE's communication is restricted to an agent and its children. Further, in MAGPIE, each agent manages the satisfaction of a set of constraints; hence it can repair independently the violation of constraints that pertain to its variables (Han & Zukerman, 1996).

Finally, Mittal and Falkenhainer (1990) described a language to specify dynamic constraint satisfaction problems, where the set of variables and constraints may change as the search progresses. However, this language cannot handle constraints with different strengths, which are required by MAGPIE (see Section 5).

Existing systems use three types of planning approaches for multimodal presentation planning: (a) *top-down*, (b) *mixed top-down and bottom-up*, and (c) *cooperative*.

The top-down approach is used in COMET (Feiner & McKeown, 1990; McKeown, Feiner, Robin, Seligmann, & Tanenblatt, 1992). COMET first determines the communicative goals and the information to be presented

and then allocates a presentation modality (viz., text or graphics) based on a rhetorical schema. This modality annotation process is carried out during discourse planning; hence, feedback from the modality-specific generators is not considered by the discourse planner. In addition, all the means of integration between modalities are predefined in COMET.

The mixed top-down and bottom-up approach is used in WIP (Rist & André, 1992; Wahlster, André, Finkler, Profitlich, & Rist, 1993). WIP has distinct planning processes for textual and graphical presentations and applies a two-step process for presentation planning. First, a presentation planner uses a top-down method to expand communicative goals into a hierarchy of communicative acts. Second, the text generator and graphics generator use a bottom-up method to select communicative acts for realization according to their abilities. WIP's layout manager then automatically arranges layout components of different modalities into an efficient and expressive format by solving graphic constraints representing semantic and pragmatic relations between different discourse components (Graf, 1992). WIP is more flexible than COMET because modalities are selected on the basis of presentation plans, and negotiations between the layout manager and the presentation planner are allowed during the planning process.

Finally, the cooperative approach is found in a few recent systems. In the system described by Arens and Hovy (1994), discourse planning and presentation planning are implemented as two reactive planning processes. However, rather than working on the same plan as done in WIP, the discourse planning process generates discourse structures, and then the presentation planning process transforms them into presentation structures. The second process is carried out by applying modality allocation rules to a set of semantic models, which characterize the nature and functionality of the modalities supported by the system. This approach provides a generic interaction platform, in which knowledge required for multimodal presentation planning can be represented using a common knowledge representation and used by two reactive planning processes at different stages. This approach enhances the system's extensibility and portability because only the semantic models need to be modified when new interaction behaviors or new modalities are added to the system.

The DenK system (Bunt, Ahn, Beun, Boeghuis, & van Overveld, 1995) provides a cooperative human-computer interface in which an *electronic cooperator* and a user can (a) observe a visual representation of an application domain and (b) exchange information in natural language or by direct manipulation of the objects in the application domain. The electronic cooperator considers its private beliefs and its assumed mutual beliefs with the user to determine the content of a presentation. It communicates with the natural language processor and the Generalized Display Processor to convey the intended information, as well as to understand the user's

questions. Hence, interactions between these two processors are allowed, albeit indirectly. The cooperative architecture of the DenK system is independent from an application domain because of the separation between its content planning process (dialogue management) and presentation planning process (the natural language processor and the Generalized Display Processor). However, the addition of a new modality-specific generator to the system requires this generator to be able to apply the reasoning formalism used by the system.

A cooperative approach based on the *client-server* concept is used in a system described by Bourdot, Krus, and Gherbi (1995) and a system presented by Cheyer and Julia (1995). Bourdot et al. focused on multimodal presentations using alternative modalities. They developed a *modality server* for multimodal application clients on the X server under Unix and a *multimodal widget* to manage nonstandard events that occur in multimodal interactions. As a result, the system can process a user's voice commands, such as "Put the red door here," in conjunction with pointing to the intended position. This is enabled by the cooperation between a voice recognition system and a graphical interface. However, the manipulation of multimodal input or output depends on the semantics of a particular command provided by the graphical interface. Cheyer and Julia described a system that uses the Open Agent Architecture (Cohen, Cheyer, Wang, & Baeg, 1994) to enable the simultaneous combination of direct manipulation, gestural drawing, handwriting, and typed and spoken natural language in a travel planning domain. In this system, multimodal input is interpreted via the cooperation of multiple agents, where each agent may require supporting information from other distributed agents or from the user. A server called a *facilitator* is responsible for the analysis of a multimodal query and the delivery of tasks required by the query to the appropriate agents. Like the system described by Bourdot and colleagues, this system enables a user to ask for information by circling an item on the screen and speaking to a microphone. The agents in this system communicate what they can do to the facilitator. Then, when one agent asks for a capability, the facilitator matches this requirement with the agents offering the capability and routes the request to these agents.

Because our multiagent mechanism uses a hierarchical presentation planning process to generate presentations from a discourse structure determined by a discourse planner, the presentation structures reflect the overall structure of the discourse. In addition, the agent-based architecture used in MAGPIE enables dynamic activation or deactivation of modality-specific generators, and the blackboard enables these processes to communicate with each other with respect to resource restrictions imposed on presentations. As a result, the interaction between these agents is flexible. Compared with the system described by Arens and Hovy (1994), the modality-specific agents in MAGPIE do not have to share a common

knowledge representation. Our approach is similar to that used by Cheyer and Julia (1995). However, MAGPIE selects agents not only based on their capabilities (which is a static factor) but also on the resource restrictions imposed by the discourse structure (which is a dynamic factor).

8. CONCLUSION AND FUTURE WORK

Multimodal presentation planning must take into account both the overall discourse structure of the communication process and the requirements that existing plans place on the plan refinement process. The hierarchical presentation planning process used in our multiagent planning architecture satisfies the former requirement, and the constraint propagation and negotiation processes satisfy the latter requirement. In particular, our mechanism allows multimodal presentations to be generated cooperatively and simultaneously by independent modality-specific processes and supports flexible interactions between these processes.

The multiagent architecture and algorithms described in this article have been fully implemented in a prototype system that currently supports five modalities. Although the integration of modality-specific presentations and variation in display arrangements are restricted at this stage, our experiments with a few discourse plans and planning strategies have demonstrated that the extensibility and the flexibility offered by our approach are promising.

Proposals for future research concern a number of issues. First, we propose to enhance MAGPIE so that additional modalities (e.g., line charts) are supported and the existing agents offer more format varieties. For example, chart agents should be able to relocate the legend and labels of a chart (to save screen space) or allow icons to be used as labels, and the table agent should be able to use modalities such as vectors to present composite information, thereby reducing the number of columns required for the attributes in focus. An existing grammar and text generator (Elhadad, 1991) will be adopted to enable the text agent to generate text from our knowledge base. In addition, we intend to use constraints to represent time restrictions on multimodal presentations and to develop a mechanism for the propagation of time constraints. This will allow the system to manipulate the time available for generating a discourse component (e.g., the time available to the table agent or the chart agent to improve a presentation).

Further, the modality selection process in the current system is not flexible. We need a mechanism that selects modalities according to the information characteristics of the intended information, the capabilities of the modalities supported by the system, and the ability of the perceivers. The first two factors may be addressed by applying rules such as those described by Arens, Hovy, and Vossers (1993) to propose modalities that

are capable of presenting the intended information. To address the third factor, we propose to use a sophisticated user model, such as that in PPP (André, Müller, & Rist, 1996), which represents the interests and abilities of perceivers. A reasoning mechanism such as that described by Zukerman and McConachy (1993) can then be used in conjunction with the user model to anticipate the effect of different modalities on the understanding of perceivers, and to select a preferred modality. This mechanism may be extended to take into consideration graphical implicatures when determining the different components to be used in a presentation and their layout in the display (Marks & Reiter, 1990). In addition, if the perceiver has difficulty understanding a graphical presentation, strategies such as those described by Mittal, Roth, Moore, Mattis, and Carenini (1995) may be employed to produce an integrated presentation where the text contains information that explains a table or a chart.

Finally, the extension of the approach presented in this article to handle multimodal interactions requires the design of reactive agents that can translate a user's request into events and send these events to appropriate presentation agents. This may require the implementation of new event handlers and planning strategies to enable each modality-specific agent to handle the events generated by the reactive agents.

NOTES

Acknowledgments. The authors thank Tun Heng Chiang for his work on the implementation of the display modules, Damian Conway for his advice regarding the improvement of several tables and figures, and the three anonymous reviewers for their thoughtful comments.

Support. This research was supported in part by a research grant from the Faculty of Computing and Information Technology and by a Small grant from the Australian Research Council.

Authors' Present Addresses. Ingrid Zukerman, Department of Computer Science, Monash University, Clayton, Victoria 3168, Australia. E-mail: ingrid@cs.monash.edu.au. Yi Han, Public Telecommunication Systems, Philips Australia, Mulgrave, Victoria 3170, Australia. E-mail: hanyi@philips.oz.au.

HCI Editorial Record. First manuscript received November 1, 1995. Revision received June 16, 1996. Accepted by Sharon Oviatt and Wolfgang Wahlster. Final manuscript received November 14, 1996. —Editor

REFERENCES

- Allen, J. F. (1983). Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11), 832–843.
- Allen, J. F. (1994). *Natural language understanding*. Redwood City, CA: Benjamin-Cummings.

- André, E., Finkler, W., Graf, W., Rist, T., Schauder, A., & Wahlster, W. (1993). WIP: The automatic synthesis of multimodal presentations. In M. T. Maybury (Ed.), *Intelligent multimedia interfaces* (pp. 75–93). Menlo Park, CA: AAAI Press.
- André, E., Müller, J., & Rist, T. (1996). The PPP persona: A multipurpose animated presentation agent. *AVI'96 Proceedings—The International Workshop on Advanced Visual Interfaces*, 245–247. Gubio, Italy: ACM.
- Arens, Y., & Hovy, E. (1994). The design of a model-based multimedia interaction manager. *Artificial Intelligence Review*, 8(3), 95–188.
- Arens, Y., Hovy, E., & van Mulken, S. (1993). Structure and rules in automated multimedia presentation planning. *IJCAI-93 Proceedings—The Thirteenth International Joint Conference on Artificial Intelligence*, 1253–1259. Chambery, France: Morgan Kaufmann Publishers.
- Arens, Y., Hovy, E., & Vossers, M. (1993). On the knowledge underlying multimedia presentations. In M. T. Maybury (Ed.), *Intelligent multimedia interfaces* (pp. 280–305). Menlo Park, CA: AAAI Press.
- Borning, A., Freeman-Benson, B., & Wilson, M. (1992). Constraint hierarchies. *Lisp and Symbolic Computation*, 5(3), 223–270.
- Bourdot, P., Krus, M., & Gherbi, R. (1995). Management of non-standard devices for multimodal user interfaces under UNIX/X11. *CMC95 Proceedings—The International Conference on Cooperative Multimodal Communication*, 49–61. Eindhoven, The Netherlands.
- Bunt, H., Ahn, R., Beun, R. J., Boeghuis, T., & van Overveld, K. (1995). Cooperative multimodal communication in the DenK project. *CMC95 Proceedings—The International Conference on Cooperative Multimodal Communication*, 79–102. Eindhoven, The Netherlands.
- Cheyer, A., & Julia, L. (1995). Multimodal maps: An agent-based approach. *CMC95 Proceedings—The International Conference on Cooperative Multimodal Communication*, 103–113. Eindhoven, The Netherlands.
- Cohen, P. R., Cheyer, A., Wang, M., & Baeg, S. C. (1994). An open agent architecture. *Proceedings of the AAAI Spring Symposium on Software Agents*, 1–8. Stanford, CA: AAAI Press.
- Elhadad, M. (1991). *FUF user manual—version 5.0* (Technical Report CUCS-038-91). New York: Columbia University.
- Engelmore, R. S., & Morgan, A. J. (1988). *Blackboard systems*. New York: Addison-Wesley.
- Feiner, S. K., Litman, D. J., McKeown, K. R., & Passonneau, R. J. (1993). Towards coordinated temporal multimedia presentations. In M. T. Maybury (Ed.), *Intelligent multimedia interfaces* (pp. 139–147). Menlo Park, CA: AAAI Press.
- Feiner, S. K., & McKeown, K. R. (1990). Coordinating text and graphics in explanation generation. *AAAI-90 Proceedings—The Eighth National Conference on Artificial Intelligence*, 442–449. Boston: AAAI Press.
- Finin, T., Fritzson, R., McKay, D., & McEntire, R. (1994). KQML as an agent communication language. *CIKM'94 Proceedings—The Third International Conference on Information and Knowledge Management*, 1–8. New York: ACM.
- Ghedira, K. (1994). Dynamic partial constraint satisfaction by a multi-agent-simulated annealing approach. *ECAI-94 Workshop on Constraint Satisfaction Issues Raised by Practical Applications*. Amsterdam, The Netherlands.
- Graf, W. (1992). Constraint-based graphical layout of multimodal presentations. *AVI'92 Proceedings—The International Workshop on Advanced Visual Interfaces*, 365–385. Singapore: World Scientific Press.

- Han, Y. (1996). *Cooperative agents for multimodal presentation planning*. Unpublished doctoral dissertation, Monash University, Victoria, Australia.
- Han, Y., & Zukerman, I. (1995). A cooperative approach for multimodal presentation planning. *CMC95 Proceedings—The International Conference on Cooperative Multimodal Communication*, 145–159. Eindhoven: The Netherlands.
- Han, Y., & Zukerman, I. (1996). Constraint propagation in a cooperative approach for multimodal presentation planning. *ECAI-96 Proceedings—The Twelfth European Conference on Artificial Intelligence*, 256–260. Budapest, Hungary: Wiley.
- Holmes, N. (1984). *Designer's guide to creating charts & diagrams*. New York: Watson-Guptill.
- Mackinlay, J. D. (1986). Automating the design of graphical presentation of relational information. *ACM Transaction on Graphics*, 5(2), 110–141.
- Marks, J., & Reiter, E. (1990). Avoiding unwanted conversational implicatures in text and graphics. *AAAI-90 Proceedings—The Eighth National Conference on Artificial Intelligence*, 450–456. Boston: AAAI Press.
- Maybury, M. T. (1993). Planning multimedia explanations using communicative acts. In M. T. Maybury (Ed.), *Intelligent multimedia interfaces* (pp. 59–74). Menlo Park, CA: AAAI Press.
- McKeown, K. R., Feiner, S. K., Robin, J., Seligmann, D. D., & Tanenblatt, M. (1992). Generating cross-references for multimedia explanation. *AAAI-92 Proceedings—The Tenth National Conference on Artificial Intelligence*, 9–16. San Jose, CA: AAAI Press.
- Minton, S., Johnston, M., Philips, A., & Laird, P. (1990). Solving large-scale constraint satisfaction and scheduling problems using a heuristic repair method. *AAAI-90 Proceedings—The Eighth National Conference on Artificial Intelligence*, 17–24. Boston: AAAI Press.
- Mittal, S., & Falkenhainer, B. (1990). Dynamic constraint satisfaction problems. *AAAI-90 Proceedings—The Eighth National Conference on Artificial Intelligence*, 25–32. Boston: AAAI Press.
- Mittal, V. O., Roth, S., Moore, J. D., Mattis, J., & Carenini, G. (1995). Generating explanatory captions for information graphics. *IJCAI-95 Proceedings—The Fourteenth International Joint Conference on Artificial Intelligence*, 1276–1283. Montreal, Canada: Morgan Kaufmann Publishers.
- Rist, T., & André, E. (1992). Incorporating graphics design and realization into the multimodal presentation system WIP. *AVI'92 Proceedings—The International Workshop on Advanced Visual Interfaces*, 1–14. Singapore: World Scientific Press.
- Roth, S. F., & Mattis, J. (1991). Automating the presentation of information. *Proceedings of the IEEE Conference on AI Applications*, 90–97. Miami Beach, FL: IEEE.
- Wahlster, W., André, E., Finkler, W., Profitlich, H., & Rist, T. (1993). Plan-based integration of natural language and graphics generation. *Artificial Intelligence*, 63(1–2), 387–427.
- Zukerman, I., & McConachy, R. (1993). Generating concise discourse that addresses a user's inferences. *IJCAI-93 Proceedings—The Thirteenth International Joint Conference on Artificial Intelligence*, 1202–1207. Chambéry, France: Morgan Kaufmann Publishers.

On Representing Salience and Reference in Multimodal Human-Computer Interaction

From: AAAI Technical Report WS-98-09. Compilation copyright © 1998, AAAI (www.aaai.org). All rights reserved.

Andrew Kehler¹, Jean-Claude Martin², Adam Cheyer¹, Luc Julia¹, Jerry R. Hobbs¹
and John Bear¹

¹ SRI International, 333 Ravenswood Avenue, Menlo Park, CA 94025 USA

² LIMSI-CNRS, BP 133, 91403 Orsay Cedex, France

Abstract

We discuss ongoing work investigating how humans interact with multimodal systems, focusing on how successful reference to objects and events is accomplished. We describe an implemented multimodal travel guide application being employed in a set of Wizard of Oz experiments from which data about user interactions is gathered. We offer a preliminary analysis of the data which suggests that, as is evident in Huls et al.'s (1995) more extensive study, the interpretation of referring expressions can be accounted for by a rather simple set of rules which do not make reference to the type of referring expression used. As this result is perhaps unexpected in light of past linguistic research on reference, we suspect that this is not a general result, but instead a product of the simplicity of the tasks around which these multimodal systems have been developed. Thus, more complex systems capable of evoking richer sets of human language and gestural communication need to be developed before conclusions can be drawn about unified representations for salience and reference in multimodal settings.

Introduction

Multimodal systems are particularly appropriate for applications in which users interact with a terrain model that is rich in topographical and other types of information, containing many levels of detail. Applications in this class span the spectrum from travel guide systems containing static, two-dimensional models of the terrain (e.g., a map-based system), to crisis management applications containing highly complex, dynamic, three-dimensional models (e.g., a forest fire fighting system). We are currently investigating how humans interact with multimodal systems in such settings, focusing on how reference to objects and events is accomplished as a user communicates by gesturing with a pen (by drawing arrows, lines, circles, and so forth), speaking natural language, and handwriting with a pen.

In this report, we begin to address the question of how knowledge and heuristics guiding reference resolution are to be represented. Is it possible to have a unified representation for salience that is applicable across multimodal systems, or do new tasks require

new representations? Can constraints imposed by the task be modularized in the theory, or are they inherently strewn within the basic mechanisms? Can linguistic theories of reference, which typically treat gestural and spoken deixis as a peripheral phenomenon, be naturally extended to the multimodal case, in which such deixis is the norm?

A Fully Automated Multimodal Map Application

The basis for our initial study is an implemented prototype multimodal travel guide application (Cheyer & Julia 1995) that was inspired by a multimodal Wizard of Oz simulation (Oviatt 1996). The system provides an interactive interface on which the user may draw, write, or speak. The system makes available information about hotels, restaurants, and tourist sites that have been retrieved by distributed software agents from commercial Internet World Wide Web sites.

The types of user interactions and multimodal issues handled can be illustrated by a brief scenario featuring working examples. Suppose Mary is planning a business trip to Toronto, but would like to schedule some activities for the weekend. She turns on her laptop PC, executes a map application, and selects Toronto.

To determine the most appropriate interpretation for the incoming streams of multimodal input, our approach employs an agent-based framework to coordinate competition and cooperation among distributed information sources, working in parallel to resolve the ambiguities arising at every level of the interpretation process. With respect to interpreting anaphora, such as in the command "Show photo of hotel", separate information sources may contribute to the resolution:

- Context by object type: The natural language component can return a list of hotels talked about.
- Deictic: Pointing, circling, or arrow gestures might indicate the referent, which may occur before, during, or after an accompanying verbal command.
- Visual context: The user interface agent might determine that only one hotel is currently visible.

M: [Speaking] Where is downtown?
Map scrolls to appropriate area.

M: [Speaking and drawing region]
 Show me all hotels near here.
Icons representing hotels appear.

M: [Writes on a hotel] Info?
A textual description appears.

M: [Speaking] I only want hotels with a pool.
Some hotels disappear.

M: [Draws a crossout on a hotel near a highway]
Hotel disappears.

M: [Speaking and circling]
 Show me a photo of this hotel.
Photo appears.

M: [Points to another hotel]
Photo appears.

M: [Speaking] Price of the other hotel?
Price appears for previous hotel.

M: [Speaking and drawing an arrow] Scroll down.
Display adjusted.

M: [Speaking and drawing an arrow toward a hotel]
 What is the distance from here to China Town?
A line and number representing distance displayed.

- Database queries: Information from a database agent can be combined with results from other resolution strategies, such as location information for the hotel asked about.
- Discourse analysis: The discourse history provides information for interpreting phrases such as “No, the other one.”

The map application is implemented within a multi-agent framework called the Open Agent Architecture (OAA).³ The OAA provides a general-purpose infrastructure for constructing systems composed of multiple software agents written in different programming languages and running on different platforms. Similar in spirit to distributed object frameworks such as OMG’s CORBA or Microsoft’s DCOM, agent interactions are more flexible and adaptable than the tightly bound object method calls provided by these architectures, and are able to exploit parallelism and dynamic execution of complex goals. Instead of preprogrammed single method calls to known object services, an agent can express its requests in terms of a high-level logical description of what it wants done, along with optional constraints specifying how the task should be performed. This specification request is processed by one or more Facilitator agents, which plan, execute and monitor the coordination of the subtasks required to accomplish the end goal (Cohen *et al.* 1994).

³Open Agent Architecture and OAA are trademarks of SRI International. Other brand names and product names herein are trademarks and registered trademarks of their respective holders.

Application functionality in the map application is thus separated from modality of user interaction. The system is composed of 10 or more distributed agents that handle database access, speech recognition (Nuance Communications Toolkit or IBM’s Voice-Type), handwriting (by CIC) and gesture (in-house algorithms) recognition, and natural language interpretation. These agents compete and cooperate to interpret the streams of input media being generated by the user. More detailed information regarding agent interactions for the multimodal map application and the strategies used for modality merging can be found in Cheyer and Julia (1995) and Julia and Cheyer (1997).

Data Collection

Despite the coverage of the system’s current anaphora resolution capabilities, we are interested in collecting naturally-occurring data which may include phenomena not handled by our system. We therefore designed a Wizard of Oz (WOZ) experiment around the travel guide application. In WOZ experiments, users believe they are interacting directly with an implemented system, but in actuality a human “wizard” intercepts the user’s commands and causes the system to produce the appropriate output. The subject interface and wizard interface are depicted in Figure 1.

Experiment Description Subjects were asked to plan activities during and after a hypothetical business trip to Toronto. They planned places to stay, sights to see, and places to dine using speech, writing, and pen-based gestures. The task consisted of four subtasks. To provide experience using each modality in isolation, during the first two tasks subjects planned half days using speech only and pen only respectively. In the third task, subject planned two half-days using any combination of these modalities they wished. Finally, the subjects completed a direction giving task, begun by picking up a phone placed nearby. On the other end was an experimenter who told the subject that he wants to meet for dinner, providing the name of the hotel at which he is staying and the restaurant at which they are to meet. The subject then interacted with the system to determine directions to give to the experimenter. For all tasks, the subjects were given only superficial instruction on the capabilities of the system. The tasks together took an average of 40 minutes. At the end of a session, the subjects were given surveys to determine whether they understood the task and the modalities available to them, and to probe their thoughts on the quality of the system.

The interactions were recorded using video, audio, and computer storage. The video displays a side-by-

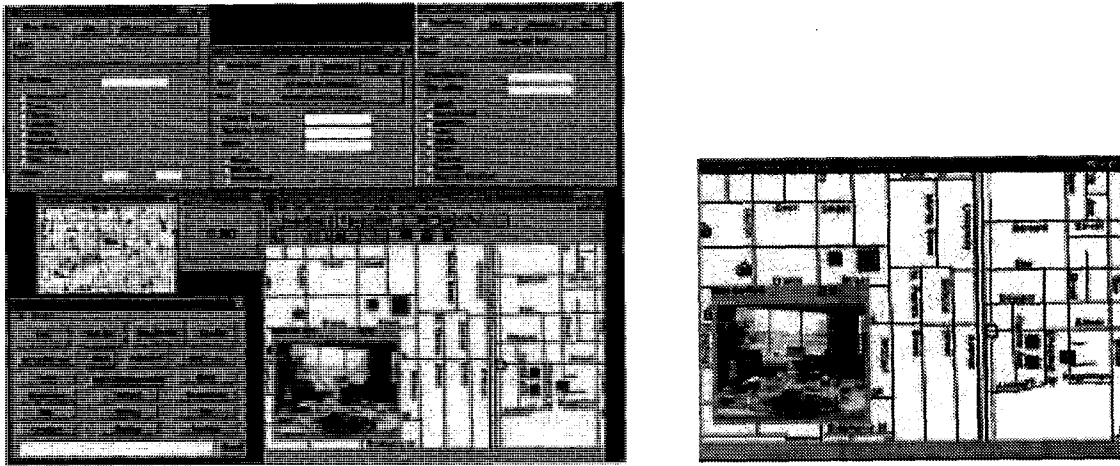


Figure 1: The Wizard Interface (left) and the Subject Interface (right)

side view with the subject on one side and the map interface on the other. The video and audio records are used for transcription, and the computer storage for reenacting scenarios for evaluation.

Coevolution of Multimodal and Wizard-of-Oz Systems In our quest for unconstrained, naturally-occurring data, we sought to place as few assumptions on the user interactions as possible. Unfortunately, WOZ experiments using simulated systems often necessitate such assumptions, so that facilities allowing the wizard to respond quickly and accurately can be encoded. We have improved upon this paradigm by having the wizard use our implemented and highly capable multimodal system to produce the answers to the user.

As described by Cheyer et al. (1998), our multimodal map application already possessed two qualities that allowed it to be used as part of a WOZ experiment. First, the system allows multiple users to share a common workspace in which the input and results of one user may be seen by all members of the session. This enables the Wizard to see the subject's requests and remotely control the display. Second, the user interface can be configured on a per-user basis to include more or fewer graphical user interface (GUI) controls. Thus, the Wizard can use all GUI command options, and also work on the map by using pen and voice. Conversely, the subject is presented with a map-only display. To extend the fully automated map application to be suitable for conducting WOZ simulations, we added only three features: a mode to disable the automatic interpretation of input from the subject, domain-independent logging and playback functions, and an agent-based mechanism for sending WOZ-specific in-

structions (e.g., Please be more specific.) to the user with text-to-speech and graphics.

The result is a hybrid WOZ experiment: While a naive user is free to write, draw, or speak to a map application without constraints imposed by specific recognition technologies, the hidden Wizard must respond as quickly and accurately as possible by using any available means. In certain situations, a scrollbar or dialog box might provide the fastest response, whereas in others, some combination of pen and voice may be the most efficient way of accomplishing the task. In a single experiment, we simultaneously collect data input from both an unconstrained new user (unknowingly) operating a simulated system – providing answers about how pen and voice are combined in the most natural way possible – and from an expert user (under duress) making full use of our best automated system, which clarifies how well the real system performs and lets us make comparisons between the roles of a standard GUI and a multimodal interface. We expect that this data will prove invaluable from an experimental standpoint, and since all interactions are logged electronically, both sets of data can be applied to evaluating and improving the automated processing.

Performing such experiments and evaluations in a framework in which a WOZ simulation and its corresponding fully functional end-user system are tightly intertwined produces a bootstrap effect: as the automated system is improved to better handle the corpus of subject interactions, the Wizard's task is made easier and more efficient for future WOZ experiments. The methodology promotes an incremental way of designing an application, testing the design through semi-automated user studies, gradually developing the automated processing to implement appropriate behavior

for input collected from subjects, and then testing the finished product while simultaneously designing and collecting data on future functionality – all within one unified implementation. The system can also be used without a Wizard, to log data about how real users make use of the finished product.

Data Analysis

At the time of this writing, 17 subjects out of a planned 25 have completed the tasks. We are currently in the process of transcribing and analyzing this data, and so we limit our discussion to a subset of 10 of the sessions. Our conclusions must therefore remain preliminary.

Our analysis of the data covers a broad range of factors concerning modality use. In addition to classical metrics used for analyzing multimodal corpora (monomodal features, temporal relationship between speech and gesture), we are analyzing the commands using a typology based on types of cooperation: specialization, equivalence, redundancy, complementarity, concurrency, and transfer (Martin 1997; Martin, Julia, & Cheyer 1998). Our focus here, however, concerns the use of referring expressions, and we therefore restrict our analysis to this issue.

Models of linguistic reference generally consist of two components. The first is the evolving representation of the discourse state, or “discourse model”, which usually includes a representation of the salience of previously introduced entities and events. For instance, entities introduced from an expression occupying subject position are generally considered as being more salient for future reference than those introduced from the direct object or other positions. The second component is a representation of the properties of referring expressions which dictates how they should be interpreted with respect to the discourse model (Prince 1981; Gundel, Hedberg, & Zacharski 1993). For instance, pronouns have been claimed to refer to entities that are highly salient or ‘in focus’, whereas full definite noun phrases need not refer to salient entities, or even ones that have been mentioned at all. Similarly, the choice among different deictic expressions (i.e., ‘this’ vs. ‘that’) is presumably guided by factors relating to the relative places at which their antecedents reside within the discourse model. Within this picture, the representation of discourse state and the interpretation of referring expressions against it are kept distinct; furthermore, they are considered independent of the task underlying the interaction.

An alternative embodied in some multimodal systems, including ours, could be termed the ‘decision list’ approach. Here, heuristics are encoded as a decision list (i.e., a list of if-then rules applied sequen-

tially) which do not necessarily enforce a strict separation between the representation of multimodally-integrated salience factors and the identities and properties of particular referring expressions. Furthermore, these rules might even query the nature of the task being performed or the type of command being issued, if task analyses would suggest that such differences be accounted for (Oviatt, DeAngeli, & Kuhn 1997).

A unified, modularized theory of reference which is applicable across multimodal applications is presumably preferable to a decision list approach. Huls et al. (1995) in fact take this position and propose such a mechanism. They describe data arising from sessions in which subjects interacted with a system using a keyboard to type natural language expressions and a mouse to simulate pointing gestures. To model discourse state, they utilize Alshawi’s (1987) framework, in which *context factors* (CFs) are assigned significance weights and a decay function according to which the weights decrease over time. Significance weights and decay functions are represented together via a list of the form $[w_1, \dots, w_n, 0]$, in which w_1 is an initial significance weight which is then decayed in accordance with the remainder of the list. The *salience value* (SV) of an entity *inst* is calculated as a simple sum of the significance weights $W(CF_i)$:

$$SV(inst) = \sum_{i=1}^n W(CF_i^{inst})$$

Four “linguistic CFs” and three “perceptual CFs” were encoded. Linguistic CFs include weights for being in a major constituent position ([3,2,1,0]), the subject position ([2,1,0], in addition to the major constituent weight), a nested position ([1,0]), and expressing a relation ([3,2,1,0]). Perceptual CFs include whether the object is visible ([1, ..., 1, 0]), selected ([2, ..., 2, 0]), and indicated by a simultaneous pointing gesture ([30, 1, 0]). The weights and decay functions were determined by trial and error.

To interpret a referring expression, the system chooses the most salient entity that meets all type constraints imposed by the command and by the expression itself (e.g., the referent of “the file” in “close the file” must be something that is a file and can be closed). This strategy was used regardless of the type of referring expression. Huls et al. tested their framework on 125 commands containing referring expressions, and compared it against two baselines: (i) taking the most recent compatible reference, and a pencil-and-paper simulation of a focus-based algorithm derived from Grosz and Sidner (1986). They found that all 125 referring expressions were correctly resolved with their approach, 124 were resolved correctly with the Grosz

and Sidner simulation, and 119 were resolved correctly with the simple recency-based strategy.

The fact that all of the methods do very well, including a rather naive recency-based strategy, indicates a lack of difficulty in the problem. Particularly noteworthy in light of linguistic theories of reference is that this success was achieved with resolution strategies that were not tied to choice of referring expression. That is, well-known differences between the conditions in which forms such as “it”, “this”, “that”, “here”, and “there” are used apparently played no role in interpretation.

We were thus inclined to take a look at the reference behavior shown in our corpus. Table 1 summarizes the distribution of referring expressions within information-seeking commands for our 10 subjects. (Commands to manipulate the environment, such as to scroll the screen or close a window, were not included.) On the vertical axis are the types of referential form used. The symbol ϕ denotes “empty” referring expressions corresponding to phonetically unrealized arguments to commands (e.g., the command “Information”, when information is requested for a selected hotel). Full NPs are noun phrases for which interpretation does not require reference to context (e.g., “The Royal Ontario Museum”), whereas definite NPs are reduced noun phrases that do (e.g., “the museum”).

On the horizontal axis are categories indicating the information status of referents. We first distinguish between cases in which an object was gestured to (e.g., by pointing or circling) at the time the command was issued, and cases in which there was no such gesture. “Unselected” refers to a (visible) object that is not selected. “Selected Immediate” includes objects that were selected and mentioned in the previous command, whereas “Selected Not Immediate” refers to objects that have remained selected despite intervening commands that have not made reference to it (e.g., due to intervening commands to show the calendar or scroll the screen). There was also one outlying case, in which the user said “Are there any Spanish restaurants here”, in which “here” referred to the area represented by the entire map.

These data show a divergence between the distribution of referring expressions and the heuristics one might use to resolve them. On one hand, there are distributional differences in even our admittedly limited amount of data that accord roughly with expectations. For instance, unselected entities, which are presumably not highly salient, were never referred to with pronominal forms without an accompanying gesture. Instead, nonpronominal noun phrases were used (20 full NPs and 2 definite NPs), and in all cases the content of the noun phrase constrained reference to one possible

antecedent (e.g., “the museum” when only one museum was visible). Also, the antecedents of empty referring expressions were almost always highly-focused (selected, immediate) objects when no accompanying gesture was used, and “it” always referred to a selected, immediate antecedent. Finally, in accordance with their generally deictic use, “this NPs” (e.g., “this museum”) and “this” were usually accompanied by a simultaneous gesture. “Here” was only used when accompanied by such a gesture, whereas “there” was used for all types of selected referents.

Certain other facets of the distribution are more contrary to expectation. For instance, in 36 cases a full NP was used to refer to a selected, immediate object which, as such, was a candidate for a reduced referential expression. In four of these cases, the user also gestured to the antecedent, resulting in an unusually high degree of redundancy. We suspect that such usage may result from a bias some users have regarding the ability of computer systems to interpret natural language.

Despite the distributional differences among the referential forms, a simple algorithm can be articulated which handles all of the data without making reference to the type of referential expression used nor its distributional properties. First, the algorithm narrows the search given any type constraints imposed by the *content* (vs. the *type*) of the referring expression, as when full and definite NPs are used. As indicated earlier, in these cases the constraints narrowed the search to the correct referent. The remaining cases are captured with two simple rules: if there was a simultaneous gesture to an object, then that object is the referent; otherwise the referent is the currently selected object.

While our preliminary findings accord with Huls et al., we have articulated our rules in decision list form rather than a salience ordering scheme. In fact, at least part of the Huls et al. analysis appears to be of the decision list variety, albeit cast in a salience ordering format. For instance, they found, as did we, that all referring expressions articulated with simultaneous gesturing to an object refer to that object. While they encode this preference with a very large weight (30), this value is chosen only to make certain that no other antecedent can surpass it.

To conclude, the question of whether a unified view of salience and reference for multimodal systems can be provided remains open. It appears that the nature of the tasks used in our experiments and by Huls et al. makes for a relatively easy resolution task. This could be due to two reasons: either reference is generally so constrained in multimodal interactions that the distinctions made by different referring expressions

| Form | No Gesture | | | Simultaneous Gesture | | | Total |
|-------------|------------|--------------------|------------------------|----------------------|--------------------|------------------------|-------|
| | Unselected | Selected Immediate | Selected Not Immediate | Unselected | Selected Immediate | Selected Not Immediate | |
| Full NP | 20 | 32 | 5 | 10 | 4 | 0 | 71 |
| Definite NP | 2 | 1 | 1 | 0 | 0 | 0 | 4 |
| “here” | 0 | 0 | 0 | 5 | 3 | 0 | 8 |
| “there” | 0 | 7 | 3 | 0 | 3 | 1 | 14 |
| “this” NP | 0 | 0 | 0 | 2 | 10 | 0 | 12 |
| “that” NP | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| “this” | 0 | 4 | 0 | 8 | 5 | 0 | 17 |
| “they” | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| “it” | 0 | 6 | 0 | 0 | 2 | 0 | 8 |
| ϕ | 0 | 22 | 2 | 13 | 1 | 0 | 38 |
| TOTAL | 22 | 74 | 11 | 38 | 28 | 1 | 174 |

Table 1: Distribution of Referring Expressions

become unimportant for understanding, or the systems that have been developed have not been complex enough to evoke the full power of human language and gestural communication. We expect that in fact the latter is the case, and are currently designing systems in more complicated domains to test this hypothesis.

Conclusions and Future Work

We have described an implemented multimodal travel guide application being used in a WOZ setting to gather data on how successful reference is accomplished. We presented a preliminary analysis of data which suggests that, as is evident in Huls et al.’s (1995) more extensive study, the interpretation of referring expressions can be accounted for by a set of rules which do not make reference to the type of expression used. This is contrary to previous research on linguistic reference, in which the differences between such forms have been demonstrated to be crucial for understanding.

We suspect that this not a general result, but instead a product of the simplicity of the tasks around which these multimodal systems have been developed. We are currently planning the development of a crisis management scenario which would involve expert or trainee fire-fighters directing resources to objectives while using a multimodal computerized terrain model. This model will be three-dimensional and dynamic, in contrast to the two-dimensional, static map application. We expect that the complexity of the task will evoke much richer interactions, and thus may serve to clarify the use of reference in these settings.

Acknowledgements

This work was supported by National Science Foundation Grant IIS-9619126, “Multimodal Access to Spatial Data”, funded within the Speech, Text, Image, and MULTimedia Advanced Technology Effort (STIMULATE).

References

- Alshawi, H. 1987. *Memory and Context for Language Interpretation*. Cambridge University Press.
- Cheyen, A., and Julia, L. 1995. Multimodal maps: An agent-based approach. In *Proceedings of CMC95*. 103–113.
- Cheyen, A.; Julia, L.; and Martin, J.-C. 1998. A unified framework for constructing multimodal experiments and applications. In *Proceedings of CMC98*, 63–69.
- Cohen, P.; Cheyer, A.; Wang, M.; and Baeg, S. 1994. An open agent architecture. In *AAAI Spring Symposium*. 1–8.
- Grosz, B., and Sidner, C. 1986. Attention, intentions, and the structure of discourse. *Computational Linguistics* 12(3):175–204.
- Gundel, J. K.; Hedberg, N.; and Zacharski, R. 1993. Cognitive status and the form of referring expressions in discourse. *Language* 69(2):274–307.
- Huls, C.; Bos, E.; and Classen, W. 1995. Automatic referent resolution of deictic and anaphoric expressions. *Computational Linguistics* 21(1):59–79.
- Julia, L., and Cheyer, A. 1997. Speech: a privileged modality. In *Proceedings of EUROSPEECH’97*. 103–113.

Martin, J.-C. 1997. Towards intelligent cooperation between modalities. The example of a system enabling multimodal interaction with a map. In *Proceedings of the IJCAI-97 Workshop on Intelligent Multimodal Systems*. 63-69.

Martin, J.-C.; Julia, L.; and Cheyer, A. 1998. A theoretical framework for multimodal user studies. In *Proceedings of CMC98*, 104-110.

Oviatt, S. 1996. Multimodal interfaces for dynamic interactive maps. In *Proceedings of CHI96*. 95-105.

Oviatt, S.; DeAngeli, A.; and Kuhn, K. 1997. Integration and synchronization of input modes during multimodal human-computer interaction. In *Proceedings of CHI97*. 415-422.

Prince, E. 1981. Toward a taxonomy of given-new information. In Cole, P., ed., *Radical Pragmatics*. New York, New York: Academic Press. 223-255.

QuickSet: Multimodal Interaction for Distributed Applications

Philip R. Cohen, Michael Johnston, David McGee, Sharon Oviatt,

Jay Pittman, Ira Smith, Liang Chen and Josh Clow

Center for Human Computer Communication

Oregon Graduate Institute of Science and Technology

P.O.Box 91000

Portland, OR 97291-1000 USA

Tel: 1-503-690-1326

E-mail: pcohen@cse.ogi.edu

<http://www.cse.ogi.edu/CHCC>

ABSTRACT

This paper presents an emerging application of multimodal interface research to distributed applications. We have developed the QuickSet prototype, a pen/voice system running on a hand-held PC, communicating via wireless LAN through an agent architecture to a number of systems, including NRaD's¹ LeatherNet system, a distributed interactive training simulator built for the US Marine Corps. The paper describes the overall system architecture, a novel multimodal integration strategy offering mutual compensation among modalities, and provides examples of multimodal simulation setup. Finally, we discuss our applications experience and evaluation.

KEYWORDS: multimodal interfaces, agent architecture, gesture recognition, speech recognition, natural language processing, distributed interactive simulation.

1. INTRODUCTION

A new generation of multimodal systems is emerging in which the user will be able to employ natural communication modalities, including voice, hand and pen-based gesture, eye-tracking, body-movement, etc. [Koons et al., 1993; Oviatt, 1992, 1996; Waibel et al., 1995] in addition to the usual graphical user interface technologies. In order to make progress on building such systems, a principled method of modality integration, and a general architecture to support it is needed. Such a framework should provide sufficient flexibility to enable rapid experimentation with different modality integration architectures and applications. This experimentation will allow researchers to discover how each communication modality can best contribute its strengths yet compensate for the weaknesses of the others.

Fortunately, a new generation of distributed system frameworks is now becoming standardized, including the CORBA and DCOM frameworks for distributed object systems. At a higher level, multiagent architectures are being developed that allow integration and interoperation of semi-autonomous knowledge-based components or "agents". The advantages of these architectural frameworks are modularity, distribution, and asynchrony — a subsystem can request that a certain functionality be provided without knowing who will provide it, where it resides, how to invoke it, or how long to wait for it. In virtue of these qualities, these frameworks provide a convenient platform for experimenting with new architectures and applications.

In this paper, we describe QuickSet, a collaborative, multimodal system that employs such a distributed, multiagent architecture to integrate not only the various user interface components, but also a collection of distributed applications. QuickSet provides a new *unification-based* mechanism for fusing partial meaning representation fragments derived from the input modalities. In so doing, it selects the best *joint* interpretation among the alternatives presented by the underlying spoken language and gestural modalities. Unification also supports multimodal discourse. The system is scaleable from handheld to wall-sized interfaces, and interoperates across a number of platforms (PC's to UNIX workstations). Finally, QuickSet has been applied to a collaborative military training system, in which it is used to control a simulator and a 3-D virtual terrain visualization system.

This paper describes the "look and feel" of the multimodal interaction with a variety of back-end applications, and discusses the unification-based architecture that makes this new class of interface possible. Finally, the paper discusses the application of the technology for the Department of Defense.

¹ NRaD = US Navy Command and Control Ocean Systems Center Research Development Test and Evaluation (San Diego).

2. QUICKSET

QuickSet is a collaborative, handheld, multimodal system for interacting with distributed applications. In virtue of its modular, agent-based design, QuickSet has been applied to a number of applications in a relatively short period of time, including:

- *Simulation Set-up and Control* — Quickset is used to control LeatherNet [Clarkson and Yi, 1996], a system employed in training platoon leaders and company commanders at the USMC base at Twentynine Palms, California. LeatherNet simulations are created using the ModSAF simulator [Courtnanche and Ceranowicz, 1995] and can be visualized in a wall-sized virtual reality CAVE environment [Cruz-Neira et al., 1993; Zyda et al., 1992] called CommandVu. A QuickSet user can create entities, give them missions, and control the virtual reality environment from the handheld PC. QuickSet communicates over a wireless LAN via the Open Agent Architecture (OAA) [Cohen et al., 1994] to ModSAF, and to CommandVu, each of which have been made into agents in the architecture.
- *Force Laydown* — QuickSet is being used in a second effort called ExInit (Exercise Initialization), that enables users to create large-scale (division- and brigade- sized) exercises. Here, QuickSet interoperates via the agent architecture with a collection of CORBA servers.
- *Medical Informatics* — A version of QuickSet is used in selecting healthcare in Portland, Oregon. In this application, QuickSet retrieves data from a database of 2000 records about doctors, specialties, and clinics.

Next, we turn to the primary application of QuickSet technology.

3. NEW INTERFACES FOR DISTRIBUTED SIMULATION

Begun as SIMNET in the 1980's [Thorpe, 1987], distributed, interactive simulation (DIS) training environments attempt to provide a high degree of fidelity in simulating combat equipment, movement, atmospheric effects, etc. One of the U.S. Government's goals, which has partially motivated the present research, is to develop technologies that can aid in substantially reducing the time and effort needed to create large-scale scenarios. A recently achieved milestone is the ability to create and simulate a large-scale exercise, in which there may be on the order of 60,000 entities (e.g., a vehicle or a person).

QuickSet addresses two phases of user interaction with these simulations: creating and positioning the entities, and supplying their initial behavior. In the first phase, a user "lays down" or places forces on the terrain, which need to be positioned in realistic ways, given the terrain, mission, available equipment, etc. In addition to force laydown the user needs to supply them with behavior, which may involve complex maneuvering, communication, etc.

Our contribution to this overall effort is to rethink the nature of the user interaction. As with most modern simulators, DISs are controlled via graphical user interfaces (GUIs). However, GUI-based interaction is rapidly losing its benefits, especially when large numbers of entities need to be created and controlled, often resulting in enormous menu trees. At the same time, for reasons of mobility and affordability, there is a strong user desire to be able to create simulations on small devices (e.g., PDA's). This impending collision of trends for smaller

screen size and for more entities requires a different paradigm for human-computer interaction with simulators.

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 [Oviatt, 1996; Oviatt, in press].² 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 [Oviatt, 1992; Oviatt 1995].

In summary, a multimodal voice/gesture interface complements, but also promises to address the limitations of, current GUI technologies for controlling simulators. In addition, it has been shown to have numerous advantages over voice-only interaction for map-based tasks. These findings had a direct bearing on the interface design and architecture of QuickSet.

4. SYSTEM ARCHITECTURE

In order to build QuickSet, distributed agent technologies based on the Open Agent Architecture³ were employed because of its flexible asynchronous capabilities, its ability to run the same set of software components in a variety of hardware configurations, ranging from standalone on the handheld PC to distributed operation across numerous computers, and its easy connection to legacy applications. Additionally, the architecture supports user mobility in that less computationally-intensive agents (e.g., the map interface) can run on the handheld PC, while more computationally-intensive processes (e.g., natural language processing) can operate elsewhere on the network. The agents may be written in any programming language (here, Quintus Prolog, Visual C++, Visual Basic, and Java), as long as they communicate via an interagent communication language. The configuration of agents used in the QuickSet system is illustrated in Figure 1. A brief description of each agent follows.

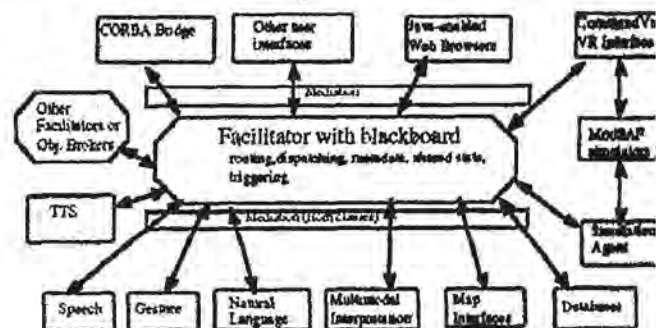


Figure 1: The facilitator, channeling queries to capable agents.

² Our prior research [Cohen et al., 1989; Cohen, 1992] has demonstrated the advantages of a multimodal interface offering natural language and direct manipulation for controlling simulators and reviewing their results.
³ Open Agent Architecture is a trademark of SRI International.

5. EXAMPLES

5.1 Leathernet

Holding QuickSet, the user views a map from the ModSAF simulation. With speech and pen, she then adds entities into the ModSAF simulation. For example, to create a unit in QuickSet, the user would hold the pen at the desired location and utter: "red T72 platoon" resulting in a new platoon of the specified type being created. The user then adds a barbed-wire fence to the simulation by drawing a line at the desired location while uttering "barbed wire." A fortified line can be added multimodally, by drawing a simple line and speaking its label, or unimodally, by drawing its military symbology. A minefield of an amorphous shape is drawn and is labeled verbally. Finally an MIA1 platoon is created as above. Then the user can assign a task to the platoon by saying "MIA1 platoon follow this route" while drawing the route with the pen.

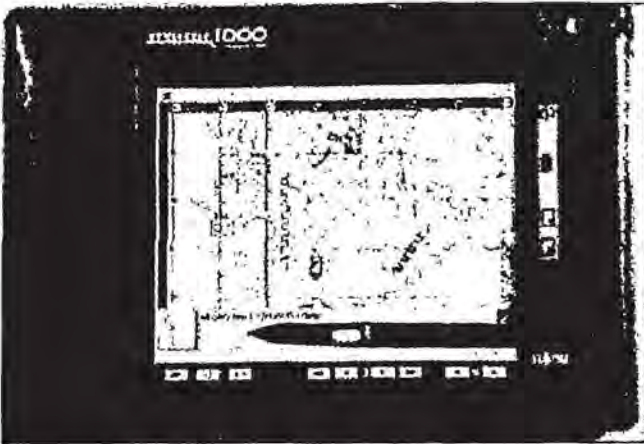


Figure 4: QuickSet running on a wireless handheld PC. The user has created numerous units, fortifications and objectives.

The results of these commands are visible on the QuickSet screen, as seen in Figure 4, as well as on the ModSAF simulation, which has been executing the user's QuickSet commands in the virtual world (Figure 5).



Figure 5: Controlling the CommandVu 3-D visualization via QuickSet interaction. QuickSet tablets are on the desks.

Two specific aspects of QuickSet to be discussed below are its usage as a collaborative system, and its ability to control a virtual reality environment.

5.1.1 Collaboration.

In virtue of the facilitated agent architecture, when two or more user interfaces connected to the same network of facilitators subscribe to and/or produce common messages, they (and their users) become part of a collaboration. The agent architecture offers a framework for heterogeneous collaboration, in that users can have very different interfaces, operating on different types of hardware platforms, and yet be part of a collaboration. For instance, by subscribing to the entity-location database messages, multiple QuickSet user interfaces can be notified of changes in the locations of entities, and can then render them in whatever form is suitable, including 2-D map-based, web-based, and 3-D virtual reality displays. Likewise, users can interact with different interfaces (e.g., placing entities on the 2-D map or 3-D VR) and thereby affect the views seen by other users. To allow for tighter synchronicity, the current implementation also allows users to decide to couple their interface to those of the other users connected to a given network of facilitators. Then, when one interface pans and zooms, the other coupled ones do as well. Furthermore, coupled interfaces subscribe to the "ink" messages, meaning one user's ink appears on the others' screens, immediately providing a shared drawing system. On the other hand, collaborative systems also require facilities to prevent users from interfering with one another. QuickSet incorporates authentication of messages in order that one user's speech is not accidentally integrated with another's gesture.

In the future, we will provide a subgrouping mechanism for users, such that there can be multiple collaborating groups using the same facilitator, thereby allowing users to be able to choose to join collaborations of specific subgroups. Also to be developed is a method for handling conflicting actions during a collaboration.

5.1.2 Multimodal Control of Virtual Travel

Most terrain visualization systems allow only for flight control, either through a joystick (or equivalent), via keyboard commands, or via mouse movement. Unfortunately, to make effective use of such interfaces, people need to be pilots, or at least know where they are going. Believing this to be unnecessarily restrictive, our virtual reality set-up follows the approach recommended by Baker and Wickens [unpublished ms], Brooks [1996], and Stoakley et al., [1995] in offering two "linked" displays — a 2-D "birds-eye" map-based display (QuickSet), and the 3-D CommandVu visualization. In addition to the existing 3-D controls, the user can issue spoken or multimodal commands via the handheld PC to be executed by CommandVu. Sample commands are:

"CommandVu, heads up display on,"

"take me to objective alpha"

"fly me to this platoon <gesture on QuickSet map>" (see Figure 4).

"fly me along this route <draws route on QuickSet map> at fifty meters"

Spoken interaction with virtual worlds offers distinct advantages over direct manipulation, in that users are able to describe entities and locations that are not in view, can be teleported to those out-of-view locations and entities, and can

QuickSet interface: On the handheld PC is a geo-referenced map of some region,⁴ 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 simulation. 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 the portion of it 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 the simulation unfold from the handheld. When the pen is placed on the screen, the speech recognizer is activated, thereby allowing users to speak and gesture simultaneously. The interface offers controls for various parameters of speech recognition, for loading different maps, for entering into collaborations with other users, for connecting to different facilitators, and for discovering other agents who are connected to the facilitator. The QuickSet system also offers a novel map-labeling algorithm that attempts to minimize the overlap of map labels as the user creates more complex scenarios, and as the entities move (cf. [Christensen et al., 1996]).

Speech recognition agent: The speech recognition agent used in QuickSet is built on IBM's VoiceType Application Factory and VoiceType 3.0, recognizers, as well as Microsoft Whisper speech recognizer.

Gesture recognition agent: QuickSet's pen-based gesture recognizer consists of both a neural network [Pittman, 1991, Manke et al., 1994] and a set of hidden Markov models. The digital ink is size-normalized, centered in a 2D image, and fed into the neural network as pixels. The ink is also smoothed, resampled, converted to deltas, and given as input to the HMM recognizer. The system currently recognizes 68 pen-gestures, including various military map symbols (platoon, mortar, fortified line, etc.), editing gestures (deletion, grouping), route indications, area indications, taps, etc. The probability estimates from the two recognizers are combined to yield probabilities for each of the possible interpretations. The inclusion of route and area indications creates a special problem for the recognizers, since route and area indications may have a variety of shapes. This problem is further compounded by the fact that the recognizer needs to be robust in the face of sloppy writing. More typically, sloppy forms of various map symbols, such as those illustrated in Figure 3, will often take the same shape as some route and area indications. A solution for this problem can be found by combining the outputs from the gesture recognizer with the outputs from the speech recognizer, as is described in the following section.

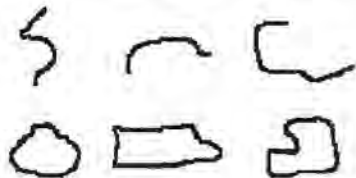


Figure 2 Pen drawings of routes and areas. Routes and areas do not have signature shapes that can be used to identify them.

⁴ QuickSet can employ either UTM or Latitude/Longitude coordinate systems.

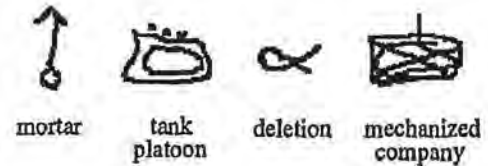


Figure 3: Typical pen input from real users. The recognizer must be robust in the face of sloppy input.

Natural language agent: The natural language agent currently employs a definite clause grammar and produces typed feature structures as a representation of the utterance meaning. Currently, for the force laydown and mission assignment tasks, the language consists of noun phrases that label entities, as well as a variety of imperative constructs for supplying behavior.

Text-to-Speech agent: Microsoft's text-to-speech system has been incorporated as an agent, residing on each individual PC.

Multimodal integration agent: The task of the integrator agent is to field incoming typed feature structures representing individual interpretations of speech and of gesture, and identify the best potential unified interpretation, multimodal or unimodal. In order for speech and gesture to be incorporated into a multimodal interpretation, they need to be both semantically and temporally compatible. The output of this agent is a typed feature structure representing the preferred interpretation, which is ultimately routed to the bridge agent for execution. A more detailed description of multimodal interpretation is in Section 6.

Simulation agent: The simulation agent, developed primarily by SRI International [Moore et al., 1997], but modified by us for multimodal interaction, serves as the communication channel between the OAA-brokered agents and the ModSAF simulation system. This agent offers an API for ModSAF that other agents can use.

Web display agent: The Web display agent can be used to create entities, points, lines, and areas, and posts queries for updates to the state of the simulation via Java code that interacts with the blackboard and facilitator. The queries are routed to the running ModSAF simulation, and the available entities can be viewed over a WWW connection.

CommandVu agent: Since the CommandVu virtual reality system is an agent, the same multimodal interface on the handheld PC can be used to create entities and to fly the user through the 3-D terrain.

Application bridge agent: The bridge agent generalizes the underlying applications' API to typed feature structures, thereby providing an interface to the various applications such as ModSAF, CommandVu, and Exinit. This allows for a domain-independent integration architecture in which constraints on multimodal interpretation are stated in terms of higher-level constructs such as typed feature structures, greatly facilitating reuse.

CORBA bridge agent: This agent converts OAA messages to CORBA IDL (Interface Definition Language) for the Exercise Initialization project.

To see how QuickSet is used, we present the following examples.

ask questions about entities in the scene. We are currently engaged in research to allow the user to gesture directly into the 3-D scene while speaking, a capability that will make these more sophisticated interactions possible.

5.2 Exercise Initialization: ExInit

QuickSet has been incorporated into the DoD's new Exercise Initialization tool, whose job is to create the force laydown and initial mission assignments for very large-scale simulated scenarios. Whereas previous manual methods for initializing scenarios resulted in a large number of people spending more than a year in order to create a division-sized scenario, a 60,000+ entity scenario recently took a single ExInit user 63 hours, most of which was computation.

ExInit is distinctive in its use of CORBA technologies as the interoperation framework, and its use of inexpensive off-the-shelf personal computers. ExInit's CORBA servers (written or integrated by MRJ Corp. and Ascent Technologies) include a relational database (Microsoft Access or Oracle), a geographical information system (CARIS), a "deployment" server that knows how to decompose a high-level unit into smaller ones and position them in realistic ways with respect to the terrain, a graphical user interface, and QuickSet for voice/gesture interaction.

In order for the QuickSet interface to work as part of the larger ExInit system, a CORBA bridge agent was written for the OAA, which communicated via IDL to the CORBA side, and via the interagent communication language to the OAA agents. Thus, to the CORBA servers, QuickSet is viewed as a Voice/Gesture server, whereas to the QuickSet agents, ExInit is simply another application agent. Users can interact with the QuickSet map interface (which offers a fluid multimodal interface), and view ExInit as a "back-end" application similar to ModSAF. A diagram of the QuickSet-ExInit architecture can be found in Figure 6. Shown there as well is a connection to DARPA's Advanced Logistics Program demonstration system for which QuickSet is the user interface.

To illustrate the use of QuickSet for ExInit, consider the example of Figure 7, in which, a user has said: "Multiple boundaries," followed in rapid succession by a series of multimodal utterances such as "Battalion <draws line>," "Company <draws line>," etc. The first utterance tells ExInit that subsequent input is to be interpreted as a boundary line, if possible. When the user then names an echelon and draws a line, the multimodal input is interpreted as a boundary of the appropriate echelon.

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. Then the user creates a number of armored companies facing 45 degrees in defensive posture; he is now beginning to add armored companies facing 225 degrees, etc. Once the user is finished positioning the entities, he can ask for them to be deployed to a lower-level (e.g., platoon).

An informal user test was recently run in which an experienced ExInit user (who had created the 60,000 entity scenario) designed his own test scenario involving the creation of 8 units and 15 control measures (e.g., the lines and areas shown in Figure 7). The user first entered the scenario via the ExInit graphical user interface, a standard Microsoft Windows mouse-menu-based GUI. Then, after a relatively short training session with QuickSet, he created the same scenario using speech and gesture. Interaction via QuickSet resulted in a two-fold to seven-fold speedup, depending on the size of the units involved (companies or battalions). Although a more comprehensive user test remains to be conducted, this early data point indicates the productivity gains that can potentially be derived from using multimodal interaction.

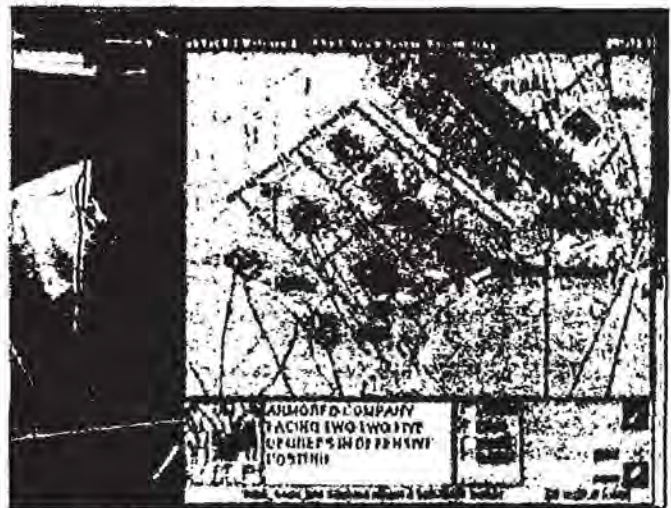


Figure 7: QuickSet used for ExInit — large-scale exercise initialization

5.3 Multimodal Interaction with Medical Information: MIMI

The last example a QuickSet-based application is MIMI, which allows users to find appropriate health care in Portland, Oregon. Working with the Oregon Health Sciences University, a prototype was developed that allows users to inquire using speech and gesture about available health care providers. For example, a user might say "show me all psychiatrists in this neighborhood <circling gesture on map>". The system translates the multimodal input into a query to a database of doctor records. The query results in a series of icons being displayed on the map. Each of these icons contains one or more health care providers meeting the appropriate criterion. Figure 8 show the map-based interaction supported by MIMI.

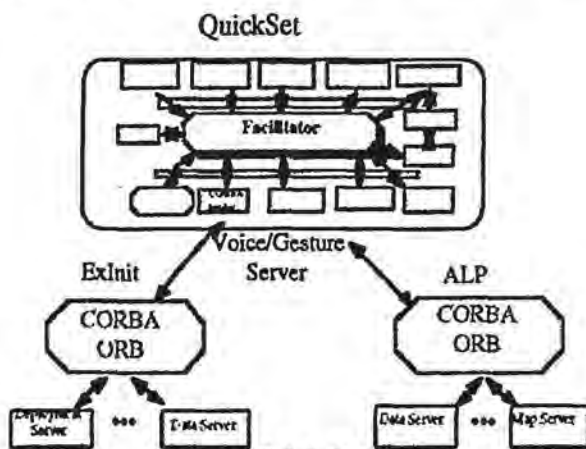


Figure 6: ExInit Architecture



Figure 8: Multimodal Interaction with Medical Information

Users can ask to see details of the providers and clinics, ask follow-up questions, and inquire about transportation to those sites.

In summary, QuickSet provides a multimodal interface to a number of distributed applications, including simulation, force laydown, virtual reality, and medical informatics. The heart of the system is its ability to integrate continuous spoken language and continuous gesture. Section 6 discusses the unification-based architecture that supports this multimodal integration.

6. MULTIMODAL INTEGRATION

Given the advantages of multimodal interaction, the problem of integrating multiple communication modalities is key to future human-computer interfaces. However, in the sixteen years since the "Put-That-There" system [Bolt 1980], research on multimodal integration has yet to yield a reusable scaleable architecture for the construction of multimodal systems that integrate gesture and voice. As we reported in Johnston et al. [1997], we see four major limiting factors in previous approaches to multimodal integration:

- The majority of approaches only consider simple deictic pointing gestures made with a mouse [Brisson and Vigouroux (ms.); Cohen 1992; Neal and Shapiro 1991; Wauchope 1994] or with the hand [Bolt, 1980; Koons et al 1993].
- Most previous approaches have been primarily language-driven, treating gesture as a secondary dependent mode [Neal and Shapiro 1991, Cohen 1992; Brisson and Vigouroux (ms.), Koons et al 1993, Wauchope 1994]. In these approaches, 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. 'the platoon facing east,' 'this one', etc.).
- None of the existing approaches provide a well-understood and generally applicable common meaning representation for the different modes.
- None of the existing approaches provide a general and formally-well defined mechanism for multimodal integration.

6.1 Multimodal Architecture Requirements

In order to create such a mechanism we need:

- Parallel recognizers and "understanders" that produce a set of time-stamped meaning fragments for each continuous input stream
- A common framework within which to represent those meaning fragments
- A time-sensitive grouping process that decides *which* meaning fragments from each modality stream should be combined. For example, should the gesture in a sequence of <speech, gesture, speech> be interpreted with the preceding speech, the following speech, or by itself?
- Meaning "fusion" operations that combine semantically compatible meaning fragments. The modality combination operation needs to allow any meaningful part to be expressed in any of the available modalities
- A process that chooses the best *joint* interpretation of the multimodal input. Such a process will support mutual compensation of modes — allowing, for example, speech to compensate for errors in gesture recognition, and vice-versa.
- A flexible asynchronous architecture that allows multiprocessing and can keep pace with human input.

6.2 Overview Of Quickset's Approach To Multimodal Integration

Using a distributed agent architecture, we have developed a multimodal integration process for QuickSet that meets these goals.

- The system employs continuous speech and continuous gesture recognizers running in parallel. A wide range of continuous gestural input is supported, and integration may be driven by either mode.
- Typed feature structures are used to provide a clearly defined and well understood common meaning representation for the modes.
- Multimodal integration is accomplished through unification.
- The integration is sensitive to the temporal characteristics of the input in each mode.
- The unification-based integration method allows spoken language and gesture to compensate for recognition errors in the other modality.
- The agent architecture offers a flexible asynchronous framework within which to build multimodal systems.

In the remainder of this section, we briefly present the multimodal integration method. Further information can be found in [Johnston et al., 1997].

6.3 A Temporally-Sensitive Unification-Based Architecture for Multimodal Integration

One the most significant challenges facing the development of effective multimodal interfaces concerns the integration of input from different modes. In QuickSet, inputs from each mode need to be both temporally and semantically compatible before they will be fused into an integrated meaning.

6.3.1 Temporal compatibility

In recent empirical work [Oviatt et al. 1997], it was discovered that when users speak and gesture in a sequential manner, they

gesture first, then speak within a relatively short time window; speech rarely precedes gesture. As a consequence, our multimodal interpreter prefers to integrate gesture with speech that follows within a short time interval, than with preceding speech. If speech arrives after that interval, the gesture will be interpreted unimodally. This temporally-sensitive architecture requires that there at least be time stamps for the beginning and end of each input stream. However, this strategy may be difficult to implement for a distributed environment in which speech recognition and gesture recognition might be performed by different machines on a network, requiring a synchronization of clocks. For this reason, it is preferable to have speech and gestural processing performed on the same machine.

6.3.2 Semantic compatibility through unification of typed feature structures

Semantic compatibility is captured via unification over typed feature structures [Carpenter 1990, 1992; Calder 1987]. Unification is an operation that determines the consistency of two representational structures, and if they are consistent combines them into a single result. Feature structure unification is a generalization of term unification in logic programming languages, such as Prolog (and is often implemented using term unification). Feature structure unification differs from term unification in logic programming where the features are positionally encoded in a term, in that they are explicitly labeled and unordered in a feature structure.

A feature structure consists of a collection of feature-value pairs. The value of a feature may be an atom, a variable, or another feature structure. When two features structures are unified, a composite structure containing all of the feature specifications from each component structure is formed. Any feature common to both feature structures must not clash in its 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 bound together, constraining them to always receive the same value (if unified with another appropriate feature structure). If the values are themselves feature structures, the unification operation is applied recursively. Importantly, feature structure unification can result in a directed acyclic graph structure when more than one value in the collection of feature/values pairs makes use of the same variable. Whatever value is ultimately unified with that variable thus will fill the value slot of all the corresponding features, resulting in a DAG.

Typed feature structures are an extension of the representation whereby feature structures and atoms are assigned to hierarchically ordered types. Typed feature structure unification requires pairs of feature structures or pairs of atoms which are being unified to be compatible in type. To be compatible in type, 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 or atom in the type hierarchy.

Typed feature structure unification is ideally suited to the task of multimodal integration because 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. Unification is appropriate for multimodal integration because it

can combine complementary or redundant input from both modes⁵ but rules out contradictory inputs.

6.3.3 Advantages of typed feature structure unification

We identify four advantages of using typed feature structure unification to support multimodal integration — partiality, mutual compensation, structure sharing, and multimodal discourse. These are discussed below.

Partial meaning representations. 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, but are given a certain type based on the semantics of the input. 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*, as in Figure 9 where the spoken phrase 'barbed wire' is assigned the feature structure shown.

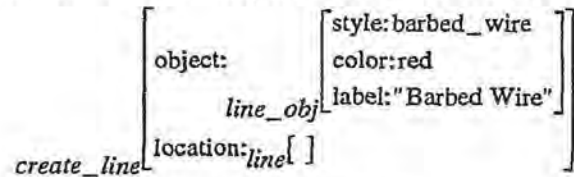


Figure 9: Feature Structure for 'barbed wire'

Since QuickSet is a task-based system directed toward setting up a scenario for simulation, this phrase is interpreted as a partially specified creation command. Before it can be executed, it needs a location feature indicating where to create the line, which is provided by the user's drawing 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 are represented as typed feature structures (see Figures 10 and 11). Interpretations of gestures as location features are assigned the more general *command* type which unifies with all of the commands supported by the system, one of which is *create_line* (see Figure 9).

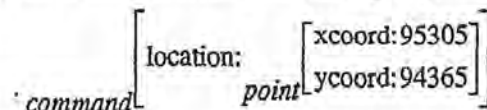


Figure 10: Point Interpretation of Gesture

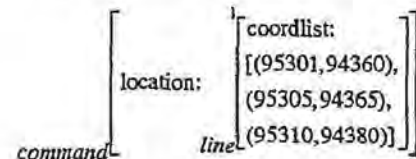
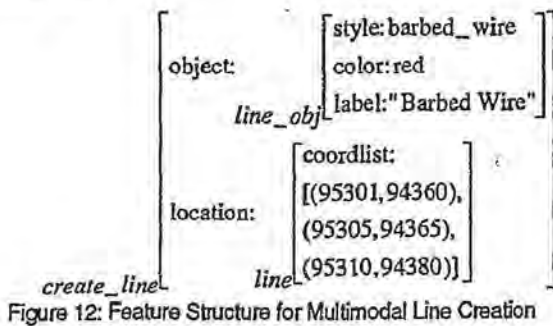


Figure 11: Line Interpretation of Gesture

Multimodal Compensation. 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 *line*, only unification with the line interpretation of the gesture will

⁵ Redundant multimodal input occurs infrequently in map-based tasks [Oviatt and Olsen, 1994; Oviatt et al. 1977].

succeed and be passed on as a valid multimodal interpretation (Figure 12).



The ambiguity of interpretation of the gesture was resolved by integration with speech, which in this case required a location feature of type *line*. If the spoken command had instead been 'M1A1 Platoon', intending to create an entity at the indicated location, it would have selected the point interpretation of the gesture in Figure 10. Similarly, if the spoken command described an area, for example a swamp, 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. As a simple example, in the open microphone mode, 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 recognizes 'M1A1 platoon', but there is no overlapping or immediately preceding gesture to provide the location, the speech will be ignored. More generally, the architecture also supports selection among the n-best speech recognition results on the basis of the preferred gesture recognition. We obtain the best joint interpretation using the maximum of the sum of the log probabilities of the spoken and gestural interpretations among the semantically and temporally compatible joint interpretations. We are currently engaged in quantifying the benefits observed by this mutually compensatory recognition process.

Structure Sharing. Another advantage of typed feature structure unification is the use of shared variables among elements of the feature structure. For example, if the user says "M1A1 platoon facing this way <draws arrow>", in the resulting feature structure, the orientation feature of the command is structured-shared with the angle of its location feature. When it is unified with an arrow gesture feature structure, the orientation feature is automatically instantiated with the angle at which the arrow was drawn.

Multimodal Discourse. The user can explicitly enter into a "mode" in which s/he is creating a specific type of entity, for example, M1A1 platoons, by simply saying "multiple M1A1 platoons." This results in a more specific feature structure that will subsequently be unified with future input (Figure 13).

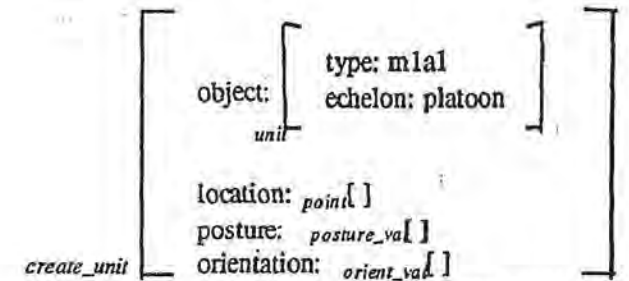


Figure 13: Feature structure for the "mode" of creating M1A1 platoons"

For example, the user could then place the pen at a desired location and say "whiskey four six," intending to create an M1A1 platoon named "W46" at that location. Any phrase resulting in a structure that unifies with the type of entity that is being created will result in the creation of that more specific type of entity. For instance, the subsequent utterances "whiskey four seven facing southeast," "whiskey four eight oriented one hundred and thirty five degrees," (see Figure 7), result in the creation of units with those names and orientations. When there is no interpretation that unifies with the one initially specified, the "mode" is ended.

In summary, we have identified four main advantages to using unification of typed feature structures as the core of a multimodal integration process: partiality, mutual compensation, structure sharing, and multimodal discourse. In virtue of these capabilities, the QuickSet system is now a usable testbed for experimenting with multimodal architectures, and for developing next-generation multimodal systems.

Vo and Wood [1996] and Waibel et al., [1995] 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. In particular, the unification approach allows for DAG interpretations and supports multimodal discourse in an elegant way. Cheyer and Julia [1995] sketch a system based on Oviatt's [1996] results and the Open Agent Architecture [Cohen et al., 1994], but describe neither the integration strategy nor multimodal compensation.

7. CONCLUDING REMARKS

QuickSet has been delivered to the US Navy and US Marine Corps. for use at Twentynine Palms, California, where it is primarily used to set up training scenarios and to control the virtual environment. The system was also used by the US Army's 82 Airborne Corps. at Ft. Bragg during the Royal Dragon Exercise. There, QuickSet was deployed in a tent, where it was subjected to noise from explosions, low-flying jet aircraft, generators, etc. Not surprisingly, it readily became apparent that spoken interaction with QuickSet would not be feasible. To support usage in such a harsh environment, a complete overlap in functionality between speech, gesture, and direct manipulation was desired. The system has been revised to accommodate these needs. As part of ExInit, QuickSet is being delivered to STRICOM, the US Army's Simulation and Training Command for use in DARPA's STOW-97 Advanced Concept Demonstration.

Regarding the multimodal interface itself, QuickSet has undergone a "proactive" interface evaluation in that high-

fidelity "wizard-of-Oz" studies were performed in advance of building the system, which predicted the utility of multimodal over unimodal speech as an input to map-based systems [Oviatt, 1996; Oviatt et al., 1997]. For example, it was discovered there that multimodal interaction would lead to simpler language than unimodal speech. Such observations have been confirmed when examining how users would create linear features with CommandTalk [Moore et al., 1997], a unimodal spoken system that also controls LeatherNet. Whereas to create a "phase line" between two three-digit $\langle x,y \rangle$ grid coordinates, a user would have to say: "create a line from nine four three nine six one to nine five seven nine six eight and call it phase line green," a QuickSet user would say "phase line green" while drawing a line. Given that numerous difficult-to-process linguistic phenomena (such as utterance disfluencies) are known to be elevated in lengthy utterances and also to be elevated when people speak locative constituents [Oviatt, 1996; Oviatt in press], multimodal interaction that permits pen input to specify locations offers the possibility of more robust recognition.

In summary, we have developed a handheld system that integrates numerous advanced technologies, including speech recognition, gesture recognition, natural language processing, multimodal integration, distributed agent technologies, and reasoning. The multimodal integration strategy allows speech and gesture to compensate for each other, yielding a more robust system. We are currently engaged in evaluation experiments to quantify the benefits of this approach. The system interoperates with existing military simulators and virtual reality environments through a distributed agent architecture. QuickSet has been deployed for the US Navy, US Marine Corps, and the US Army, and is being integrated into the DARPA STOW-97 ACTD. We are currently evaluating its performance in the field.

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, MRJ Corp. and SRI International.

REFERENCES

- Bolt, R. A. 1980. "Put-That-There": Voice and gesture at the graphics interface. *Computer Graphics*. 14.3, pp. 262-270.
- Baker, M. P., and Wickens, C. D. Human factors in virtual environments for the visual analysis of scientific data. Unpublished ms., University of Illinois.
- Brooks, Frederic. 3-D user interfaces: When results matter, Invited presentation (unpublished), UIST'96, Seattle, 1996.
- Brison, E. and Vigouroux, N. (unpublished ms.). Multimodal references: A generic fusion process. URIT-URA CNRS. Université Paul Sabatier, Toulouse, France.
- Calder, J. 1987. Typed unification for natural language processing. In E. Klein and J. van Benthem (Eds.), *Categories, Polymorphisms, and Unification*. Centre for Cognitive Science, University of Edinburgh, Edinburgh, pp. 65-72.
- Cheyser, A., and Julia L. 1995. Multimodal maps: An agent-based approach. *International Conference on Cooperative Multimodal Communication (CMC/95)*, May 1995. Eindhoven, The Netherlands. pp. 24-26.
- Carpenter, R. 1990. Typed feature structures: Inheritance, (In)equality, and Extensionality. In W. Daelemans and G. Gazdar (Eds.), *Proceedings of the ITK Workshop: Inheritance in Natural Language Processing*, Tilburg University, pp. 9-18.
- Carpenter, R. 1992. *The logic of typed feature structures*. Cambridge University Press, Cambridge.
- Christensen, J., Marks, J., and Shieber, S. Placing text labels on maps and diagrams, *Graphics Gems IV*, Heckbert, P. (ed.), Academic Press, Cambridge, Mass., 1994, 497-504.
- Clarkson, J. D., and Yi, J. LeatherNet: A synthetic forces tactical training system for the USMC commander. *Proceedings of the Sixth Conference on Computer Generated Forces and Behavioral Representation*. Orlando, Florida, 1996 275-281.
- Cohen, P. R. The Role of Natural Language in a Multimodal Interface. *Proceedings of UIST'92*, ACM Press, NY, 1992, 143-149.
- Cohen, P. R., Dalrymple, M., Moran, D. B., Pereira, F. C. N., Sullivan, J. W., Gargan, R. A., Schlossberg, J. L., Tyler, S. W., Synergetics use of direct manipulation and natural language, *Proceedings of Human Factors in Computing Systems (CHI'89)*, ACM Press, New York, 1989, 227-234.
- Cohen, P. R., Cheyer, A., Wang, M., and Baeg, S.C. An Open Agent Architecture. *Proceedings of the AAAI Spring Symposium Series on Software Agents (March 21-22, Stanford)*, Stanford Univ., CA, 1994, 1-8.
- Courtemanche, A.J. and Ceranowicz, A. ModSAF Development Status. *Proceedings of the Fifth Conference on Computer Generated Forces and Behavioral Representation*, Univ. Central Florida, Orlando, 1995, 3-13.
- Cruz-Neira, C. D. J. Sandin, T. A. DeFanti, "Surround-Screen Projection-Based Virtual Reality: The Design and Implementation of the CAVE," *Computer Graphics*, ACM SIGGRAPH, August 1993, pp. 135-142.
- Johnston, M., Cohen, P. R., McGee, D., Pittman, J., Oviatt, S. L., and Smith, I. Unification-based multimodal integration, *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL-97/EACL-97) Conference*, Madrid, Spain, July, 1997.
- Koons, D.B., C. J. Sparrell and K. R. Thorisson. 1993. Integrating simultaneous input from speech, gaze, and hand gestures. In Mark T. Maybury (ed.) *Intelligent Multimedia Interfaces*. AAAI Press/ MIT Press, Cambridge, MA, pp. 257-276.
- Manke, S., Finke, M., and Waibel, A., The use of dynamic writing information in a connectionist on-line cursive handwriting recognition system, *Advances in Neural Information processing Systems 7 (NIPS)*, 1994.
- Moore, R. C., Dowding, J, Bratt, H., Gawron, M., and Cheyer, A. CommandTalk: A spoken-language interface for battlefield simulations, *Proc. of the 3rd Applied Natural Language Conference*, Wash. DC, 1997.
- Neal, J.G. and Shapiro, S.C. Intelligent multi-media interface technology. In J.W. Sullivan and S.W. Tyler, editors, *Intelligent User Interfaces*, chapter 3, pages 45-68. ACM Press Frontier Series, Addison Wesley Publishing Co., New York, New York, 1991.
- Oviatt, S. L. Pen/Voice: Complementary multimodal communication, *Proceedings of SpeechTech'92*, New York, February, 1992, 238-241.
- Oviatt, S.L. Multimodal interfaces for dynamic interactive maps. *Proceedings of CHI'96 Human Factors in Computing Systems* ACM Press, NY, 1996, 95-102.
- Oviatt, S.L. Multimodal interactive maps: Designing for human performance, *Human Computer Interaction*, in press.

Oviatt, S. L., A. DeAngeli, and K. Kuhn. Integration and synchronization of input modes during multimodal human-computer interaction. *Proceedings of the Conference on Human Factors in Computing Systems (CHI '97)*, ACM Press, NY, 1997, 415-422.

Oviatt, S. L., and Olsen, E., Integration themes in multimodal human-computer interaction, *Proceedings of the International Conference on Spoken Language Processing*, Acoustical Society of Japan, Yokohama, Japan, 1994, 551-554.

Pittman, J. A. Recognizing handwritten text *Human Factors in Computing Systems (CHI'91)*, 1991, 271-275.

Stoakley, R., Conway, M., and Pausch, R. Virtual reality on a WIM: Interactive worlds in miniature, *Proceedings of Human Factors in Computing Systems (CHI'95)*, ACM Press, New York, 1995, 265-272.

Thorpe, J. A. The new technology of large scale simulator networking: Implications for mastering the art of warfighting. *9th Interservice Training Systems Conference*, 1987, 492-501.

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

Waibel, A., Vo, M. T., Duchnowski, P., and Manke, S. Multimodal interfaces, *Artificial Intelligence Review*, 1995.

Wauchope, K. 1994. Eucalyptus: Integrating natural language input with a graphical user interface. Naval Research Laboratory, Report NRL/FR/5510--94-9711.

Zyda, M. J., Pratt, D. R., Monahan, J. G., and Wilson, K. P., NPSNET: Constructing a 3-D virtual world, *Proceedings of the 1992 Symposium on Interactive 3-D Graphics*, March, 1992.

Permission to make digital/hard copies of all or part of this material for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication and its date appear, and notice is given that copyright is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires specific permission and/or fee.

ACM Multimedia 97 Seattle Washington USA
Copyright 1997 ACM 0-89791-991-2/97/11..\$3.50

Towards "intelligent" cooperation between modalities. The example of a system enabling multimodal interaction with a map

Jean-Claude MARTIN
LIMSI-CNRS, BP 133, 91403 Orsay Cedex, France
martin@limsi.fr

Abstract

In this paper we propose a coherent approach for studying and implementing multimodal interfaces. This approach is based on six basic "types of cooperation" between modalities: transfer, equivalence, specialization, redundancy, complementarity and concurrence. Definitions and examples of these types of cooperations are given in the paper.

We have used this approach to develop both theoretical tools (a framework, and formal notations) and software tools (a language for specifying multimodal input, and a module integrating events detected on several modalities).

These tools have been applied to the development of a prototype enabling a user to interact with a geographic map by combining speech recognition, pointing gestures with a mouse and a keyboard. We explain the underlying software architecture and give details on how the multimodal module may enable "multimodal recognition scores".

Finally, we describe what we believe "intelligent" multimodal systems should be, and how our approach based on the types of cooperation between modalities could be used in this direction.

1. [Introduction](#)
2. [Theoretical tools](#)
3. [The CARTOON prototype](#)
4. [The specification language](#)
5. [The multimodal module](#)
6. [Conclusions and perspectives](#)
7. [References](#)

1. Introduction

The development of multimodal systems addresses several issues [[Maybury 1994](#)]: content selection ("what to say"), modality allocation ("which modality to say it"), modality realization ("how to say that in that modality") and modality combination. Our work deals with the "modality combination" issue. A multimodal interface developer has to know how to combine modalities and why this combination may improve the interaction. Although several multimodal interfaces have already been developed [[CMC 1995](#); [IMMI 1995](#)], there is still a lack of coherent theoretical and software tools.

In the first part of this paper, we propose a theoretical framework for analyzing modality combinations. The second part details two software tools based on the framework: a specification language and a multimodal module using Guided Propagation Networks. Illustrative examples are taken from a prototype enabling multimodal interrogation of a geographic map developed by [[Goncalves et al. 1997](#)].

2. Theoretical tools

A system should use multimodality only if it helps in achieving usability criteria and requirement specifications such as:

- improving recognition in a noisy (audio, visual or tactile) environment,
- enabling a fast interaction,
- being intuitive or easy to learn,
- adapting to several environments, users or user's behaviors,
- enabling the user to easily link presented information to more global contextual knowledge,
- translating information from one modality to another modality...

These usability criteria may depend on the application to be developed. From a multimodal point of view, they can be seen as "goals of cooperation" between modalities. How can modalities cooperate and be combined to achieve each of these goals? We propose six basic "types of cooperation" between modalities: transfer, specialization, equivalence, redundancy, complementarity and concurrency. In this section, we define each of them and give examples on how they may help in reaching usability criteria (figure 1). In our definitions, a

modality is considered as a process receiving and producing chunks of information. More examples of types of cooperation can be found in [\[Martin et al. in press\]](#).

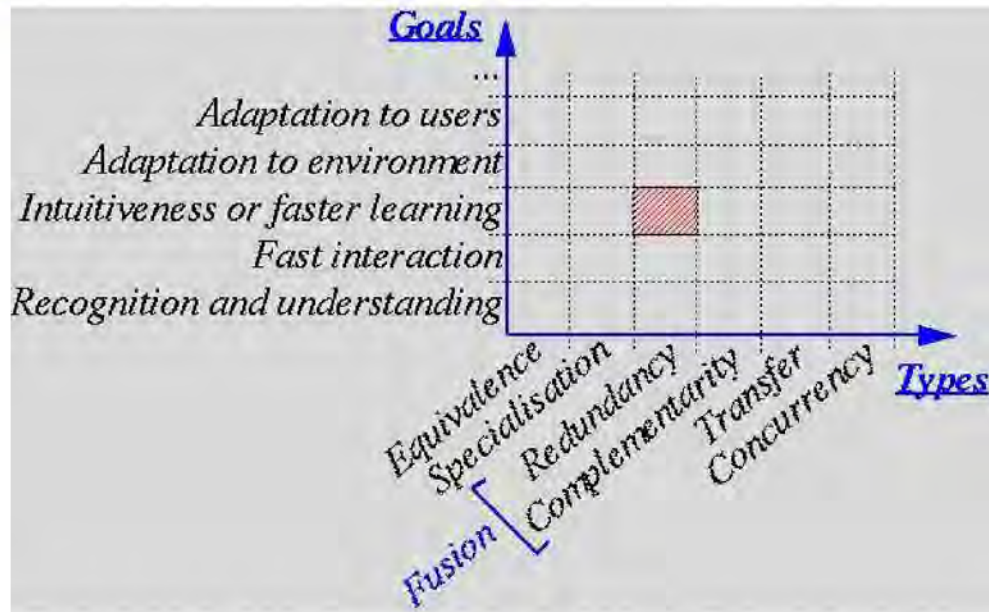


Figure 1. The framework proposed in this paper for studying and designing multimodal interfaces. Six "types of cooperation" between modalities (horizontal axis) may be involved in several "goals of cooperation (vertical axis). For instance (red box), it has been shown that with redundant displayed text and vocal output, a user learned faster how to use a graphical interface [\[Wang et al. 1993\]](#).

2.1. Equivalence

When several modalities cooperate by equivalence, this means that a chunk of information may be processed as an alternative, by either of them.

In COMIT, a multimodal interface that we have developed, the user can create a graphical interface (windows, buttons, scrollbars) interactively by combining speech, mouse and keyboard. For instance, the user may either utter or type "create a scrollbar" to create a new scrollbar.

The EDWARD system [\[Huls and Bos 1995\]](#) is applied to hierarchical file system management. It allows the user to choose at any time during the interaction the style that suits best at that moment (mouse or natural language). Experimental tests have shown that subjects tended to choose the mouse for selecting an object with a long name. Yet, when the object was difficult to locate on the screen, subjects preferred typing.

Equivalence also enables adaptation to the user by customization: the user may be allowed to select the modalities he prefers [\[Hare et al. 1995\]](#). The formation of accurate mental models of a multimodal system seems dependent upon the implementation of such options over which the user has control [\[Sims and Hedberg 1995\]](#).

Thus, equivalence means alternative. It is clear that differences between each modality, either cognitive or technical, have to be considered.

2.2 Specialization

When modalities cooperate by specialization, this means that a specific kind of information is always processed by the same modality.

Specialization is not always absolute and may be more precisely defined: one should distinguish data-relative specialization and modality-relative specialization. In several systems, sounds are somehow specialized in errors notification (forbidden commands are signaled with a beep). On the other way, it is a modality-relative specialization if sounds are not used to convey any other type of information. It is a data-relative specialization if errors only produce sounds and no graphics or text. When there is a one-to-one relation between a set of information and a modality, we will speak of an absolute specialization.

Specialization may help the user to interpret the events produced by the computer (to link them to the global contextual knowledge). This means that the choice of a given modality adds semantic information and hence helps the interpretation process.

When a modality is specialized, it should respect the specificity of this modality including the information it is good at representing. For instance, in reference interpretation, the designation gesture aims at selecting a specific area and the verbal channel provides a frame for the interpretation of the reference: categorical information, constraints on the number of objects selected [\[Bellalem and Romary 1995\]](#).

In an experimental study [Bressole et al. 1995] aiming at the understanding of cooperative cognitive strategies used by air traffic controllers, non-verbal resource are revealed to be a specific vector of communication for some types of information which are not verbally expressed such as the emergency of a situation. Intuitive specialization of a modality may go against its technical specificities. In the Wizard of Oz experiment dealing with a tourist application described in [Siroux et al. 1995], despite the low recognition rate of town names, the users did not use the tactile screen to select a town but used speech instead.

2.3. Redundancy

If several modalities cooperate by redundancy, this means that the same information is processed by these modalities.

In COMIT, if the user types "quit" on the keyboard or utters "quit", the system asks for a confirmation. But if the user both types and utters "quit", the systems interpret this redundancy to avoid a confirmation dialogue thus enabling a faster interaction by reducing the number of actions the user has to perform.

Regarding intuitiveness, redundancy has been observed in the Wizard of Oz study described in [Siroux et al. 1995]: sometimes the user selected a town both by speech and a touch on the tactile screen.

Regarding learnability of interfaces, it has been observed that a redundant multimodal output involving both visual display of a text and speech restitution of the same text enabled faster graphical interface learning [Dowell et al. 1995]. Redundancy between visual and vocal text with verbatim reinforcement was also tested in [Huls and Bos 1995] with natural language descriptions of the objects the user manipulates and the action he performs. Although speech coerced the subjects into reading the typed descriptions, the subjects made more errors and were slower than with the visual text output only.

2.4. Complementarity

When several modalities cooperate by complementarity, it means that different chunks of information are processed by each modality but have to be merged. First systems enabling the "put that there" command for the manipulation of graphical objects are described in [Carbonnel 1970 ; Bolt 1980]. In COMIT, if the user wants to create a radio button, he may type its name on the keyboard and select its position with the mouse. These two chunks of information have to be merged to create the button with the right name at the right position. This complementarity may enable a faster interaction since the two modalities can be used simultaneously and convey shorter messages which are moreover better recognized than long messages.

In [Huls and Bos 1995], experiments have shown that the use of complementarity input such as "Is this a report ?" while pointing on a file, increases with user's experience.

Complementarity may also improve interpretation, as in [Santana and Pineda 1995] where a graphical output is sufficient for an expert but need to be completed by a textual output for novice users. An important issue concerning complementarity is the criterion used to merge chunks of information in different modalities. The most classical approaches are to merge them because they are temporally coincident, temporally sequential or spatially linked. Regarding intuitiveness, complementarity behavior were observed in [Siroux et al. 1995]. Two types of behavior did feature complementarity. In the "sequential" behavior, which was rare, the user would by example utter "what are the campsites at" and then select a town with the tactile screen. In the "synergistic" behavior, the user would utter "Are there any campsites here ?" and select a town with the tactile screen while pronouncing "here". Regarding the output from the computer, it was observed in the experiment described in [Hare et al. 1995] that spatial linking of related information encourages the user's awareness of causal and cognitive links. Yet, when having to retrieve complementary chunks of information from different media, users behavior tended to be biased towards sequential search avoiding synergistic use of several modalities.

Modalities cooperating by complementarity may be specialized in different types of information. In the example of a graphical editor, the name of an object may be always specified with speech while its position is specified with the mouse. But modalities cooperating by complementarity may be also be equivalent for different types of information. As a matter of fact, the user could also select an object with the mouse and its new position with speech ("in the upper right corner"). Nevertheless, the complementary use of specialized modalities gives the advantages of specialization: speech recognition is improved since the vocabulary and syntax is simpler than a complete linguistic description.

2.5. Transfer

When several modalities cooperate by transfer, this means that a chunk of information produced by a modality is used by another modality.

Transfer is commonly used in hypermedia interfaces when a mouse click provokes the display of an image. In information retrieval applications, the user may express a request in one modality (speech) and get relevant information in another modality (video) [Foote et al. 1995]. Output information may not only be retrieved but also produced from scratch. Several systems generate graphical descriptions of a scene from a linguistic description [O Nuallain and Smith 1994]. Natural language instructions can also be used to create animated simulations of virtual human agents carrying out tasks [Webber 1995]. Similarly, the visual description of a scene can be used to generate a linguistic description [Jackendoff 1987] or a multimodal description [André and Rist 1995]. Let's say that all these previous examples involved transfer for a goal of translation.

Transfer may also be involved in other goals such as improving recognition: mouse click detection may be transferred to a speech modality in order to ease the recognition of predictable words (here, that...) as in the GERBAL system [Salisbury et al. 1990].

2.6. Concurrency

Finally, when several modalities cooperate by concurrency, it means that different chunks of information are processed by several modalities at the same time but must not be merged. This may enable a faster interaction since several modalities are used in parallel.

2.7. Formal notations

To define more precisely these types of cooperation, we propose logical formal notations. They aim at stating explicitly the parameters of each type of cooperation and the relation between these parameters which is subsumed by the type of cooperation. We consider the case of input modalities (human towards computer). These formal notations have helped us in defining a specification language for implementing multimodal interfaces (next section).

We define a modality as a process receiving and producing chunks of information. A modality M is formally defined by:

- $E(M)$ the set of chunks of information received by M
- $S(M)$ the set of chunks of information produced by M

Two modalities M_1 and M_2 cooperate by transfer when a chunk of information produced by M_1 can be used by M_2 after translation by a transfer operator tr which is a parameter of the cooperation.

$$\begin{aligned} & \textit{transfer} (M_1, M_2, tr): \\ & \quad tr(S(M_1)) \subset E(M_2) \end{aligned}$$

An input modality M cooperate by specialization with a set of input modalities M_i in the production of a set I of chunks of information if M produces I (and only I) and no modality in M_i produces I .

$$\begin{aligned} & \textit{specialisation}(M, I, \{M_i\}): \\ & \quad I = S(M) \wedge \forall M_i, I \not\subset S(M_i) \end{aligned}$$

Two input modalities M_1 and M_2 cooperate by equivalence for the production of a set I of chunks of information when each element i of I can be produced either by M_1 or M_2 . An operator eq controls which modality will be used and may take into account user's preferences, environmental features, information to be transmitted...

$$\begin{aligned} & \textit{equivalence} (M_1, M_2, I, eq): \\ & \quad \forall i \in I, \exists e_1 \in E(M_1), \exists e_2 \in E(M_2), i = eq((M_1, e_1), (M_2, e_2)) \end{aligned}$$

Two input modalities M_1 and M_2 cooperate by redundancy for the production of a set I of chunks of information when each element i of I can be produced by an operator re merging a couple (s_1, s_2) produced respectively by M_1 and M_2 . The operator re will merge (s_1, s_2) if their redundant attribute has the same value and a criterion $crit$ is true. A chunk of information has several attributes. For instance, a chunk of information sent by a speech recognizer has the following attributes: time of detection, label of recognized word, recognition score. The redundant attribute of two modalities plays a role in deciding whether two chunks of information produced by these modalities is redundant or complementary.

$$\begin{aligned} & \textit{redundancy} (M_1, M_2, I, \textit{redundant_attribute}, \textit{crit}): \\ & \quad \forall i \in I, \exists s_1 \in S(M_1), \exists s_2 \in S(M_2), \\ & \quad \textit{redundant_attribute} (s_1) = \textit{redundant_attribute} (s_2) \wedge \\ & \quad i = \textit{re}(s_1, s_2, \textit{crit}) \end{aligned}$$

Two input modalities M_1 and M_2 cooperate by complementarity for the production of a set I of chunks of information when each element i of I can be produced by an operator co merging a couple (s_1, s_2) produced respectively by M_1 and M_2 . The process co will merge (s_1, s_2) if their redundant attribute does not have the same value and a criterion $crit$ is true:

complementarity ($M_1, M_2, I, \text{redundant_attribute}, \text{crit}$):
 $\forall i \in I, \exists s_1 \in S(M_1), \exists s_2 \in S(M_2),$
 $\text{redundant_attribute}(s_1) \neq \text{redundant_attribute}(s_2) \wedge$
 $i = \text{co}(s_1, s_2, \text{crit})$

In the next sections, we introduce a specification language based on these formal notation. This language has been used for the implementation of a multimodal prototype: CARTOON.

3. The CARTOON prototype

We have implemented CARTOON (CARTography and cOOperation between modalities), a multimodal interface to a cartographic application developed by [Goncalves et al. 1997] enabling the manipulation of streets, the computation of shortest itinerary... Multimodal interrogation of maps seems to be a promising application for multimodal systems [Cheyer and Julia 1995 ; Siroux et al. 1995] as more and more tourist information is available on the Internet. Figure 2 shows a screen dump during a multimodal interaction in CARTOON. A map is displayed on the screen. The user may combine speech utterances and pointing gestures with the mouse. For instance, the user may utter (translated from French) "I want to go from here to here". Then the system computes the shortest itinerary and the streets to be taken are displayed in red. The following combinations are possible with CARTOON:

- Where is the police station ?
- Show me the hospital
- I want to go from here to the hospital
- I am in front of the police station. How can I go here ?
- What is the name of this building ?
- What is this ?
- Show me how to go from here to here

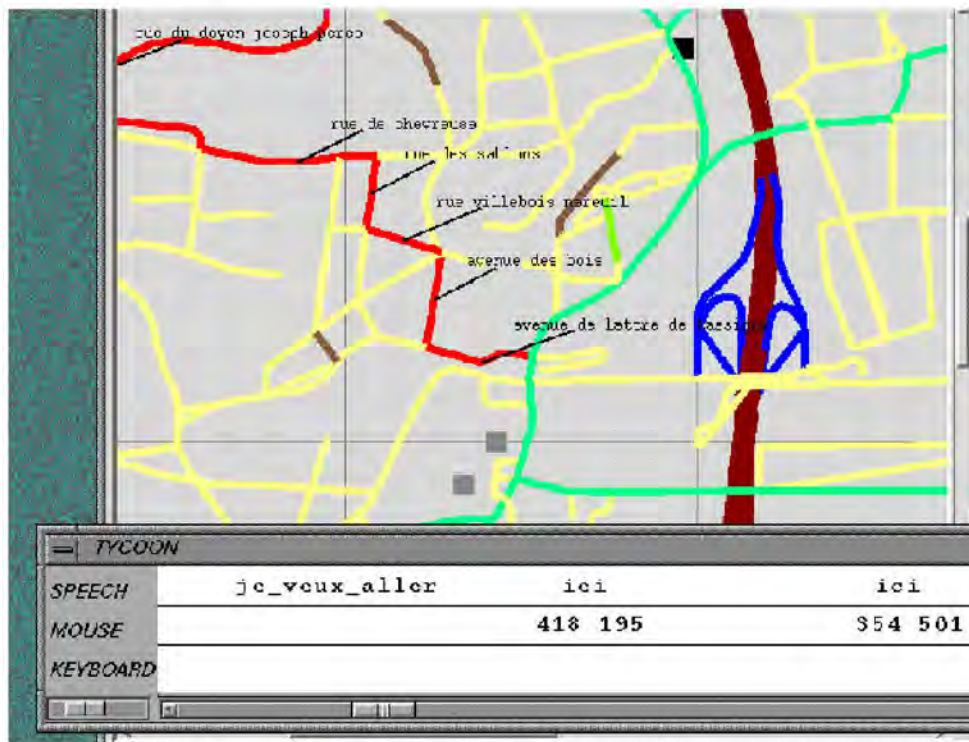


Figure 2. Example of a multimodal interaction with the CARTOON prototype. The events detected on the three modalities (speech, mouse, keyboard) are displayed in the lower window as a function of time. In this case, the detected speech events were: "I_want_to_go", "here", "here". Two mouse clicks were detected. The system integrated these events as a request and displays the shortest itinerary.

In the current version, there is no linguistic analysis preliminary to the multimodal fusion. Events produced by the speech recognition system (a Vecsys Datavox) are either words ("here") or sequences of words ("I_want_to_go"). There are 38 such possible speech events. Each speech event is characterized by: the recognized word, the time of utterance and the recognition score.

The pointing gestures events are characterized by an (x, y) position and the time of detection.

The overall hardware and software architecture is described in figure 3.

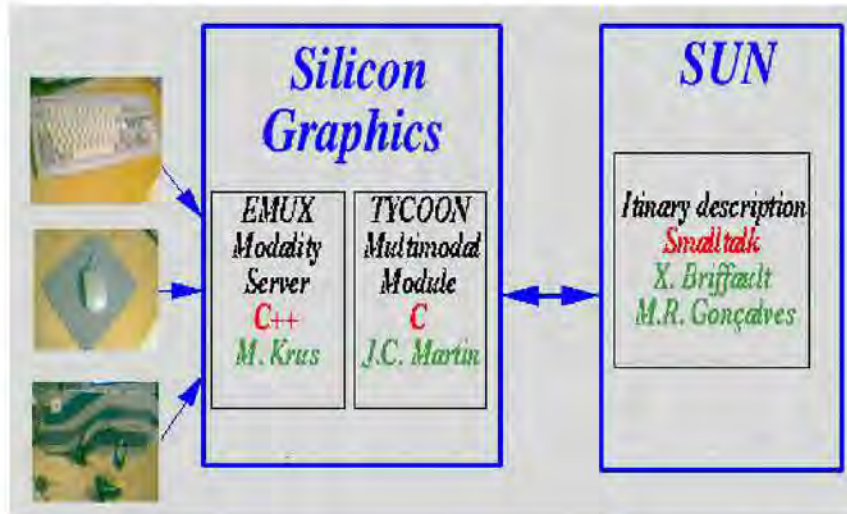


Figure 3. hardware and software architecture. Events detected on the speech, mouse and keyboard modalities (left-hand side) are time-stamped coherently by a Modality Server [Bourdot et al. 95]. The events are then integrated in our multimodal module TYCOON (in the middle) which merges them and sends messages to the cartography and itinerary application (right-hand side).

4. The specification language

The combination of modalities used in CARTOON are described in a specification language that is based on our formal notations. In this section, we explain parts of the specification file used for CARTOON.

Firstly, the modality used are specified (the objects modality is activated when one graphical object such as a building is mouse-clicked) :

```
modality Speech Keyboard Mouse Objects
```

Then, these modalities are connected to the multimodal module:

```
link      Speech      Multimodal
link      Mouse       Multimodal
link      Keyboard    Multimodal
link      Objects     Multimodal
```

The events to be detected on each modality are also specified (38 speech items):

```
event      Speech  where_is
           Speech  show_me
           Speech  I_am
           Speech  I_want_to_go
           ...
```

For each command of the cartographic application, the possible combination of modalities are specified. Here is the example of the command NameOf: A variable V3 is defined as the beginning of a sequence:

```
start_sequence Multimodal V3
```

It may be activated by one event among several (the word "name" typed on the keyboard or the speech items "what is the name of" or "what is that"):

```
equivalence Multimodal V3
           Keyboard name
           Speech  what_is_the_name_of
           Speech  what_is_that
```

This V3 variable is linked sequentially to a second variable V4:

```
complementarity_sequence Multimodal V3 V4
```

V4 may only be activated by a mouse event:

```
specialization Multimodal V4 Mouse *
```

V4 is bound to a parameter of an application module which is involved in the execution process:

```
bind_application Parameter1NameOf V4
```

V4 is the last variable of the sequence:

end_sequence Multimodal V4 NameOf

5. The multimodal module

The multimodal module used in CARTOON is based on Guided Propagation [Bérroule 1985] (figure 4). Such networks comprise elementary processing units: event-detectors and multimodal units. Event detectors (square units) selectively respond to events at the moment they occur in the environment. When activated by an event, these event-detectors send a signal to the multimodal units (circle units) to which they are connected. The connections between the units are build from the specification file described in the previous section.

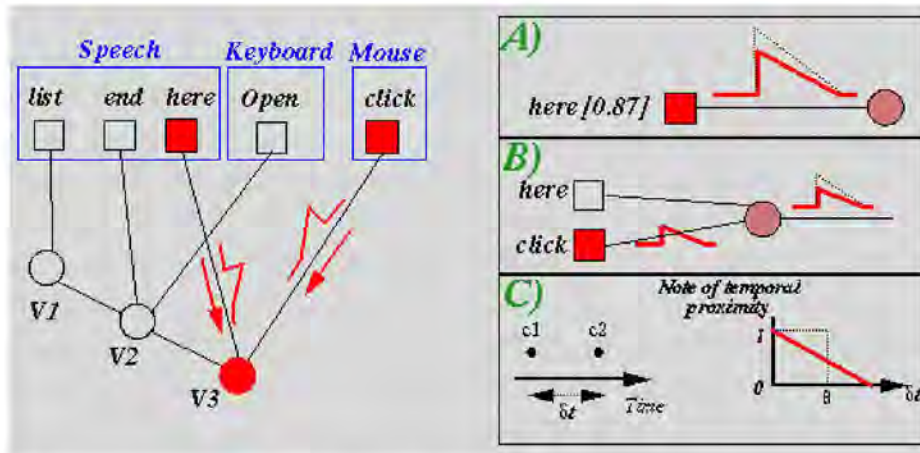


Figure 4: the multimodal module uses Guided Propagation Networks. Left-hand side: a network integrating events detected on three modalities is composed of event-detectors (square units) and multimodal units (circle units). Right-hand side: three properties of these networks enable multimodal recognition scores (see text).

The activity level of a detector at the end of a multimodal command pathway corresponds to the way an occurrence of this command matches its internal representation. This "matching score" accounts for the degree of distortions undergone by the reference multimodal command, including noisy, missing or inverse components. Initially applied to robust parsing [Westerlund et al. 1994], this feature has been adapted to multimodality [Veldman 1995]. This quantified matching score results from three properties of GPN (figure 4, right-hand side):

- A: the amplitude of the signal emitted by a speech detector is proportional to the recognition score provided by the speech recogniser
- B: a multimodal unit can be activated even if some expected events are missing (in this case, the amplitude of the signal emitted by this variable is lower than the maximum)
- C: the bigger the temporal distortion between two events, the weaker their summation (or note of temporal proximity), because of the decreasing shape of the signals.

6. Conclusion and perspectives

In this paper, we have described some theoretical and software tools that we have developed. We explained how we used them for implementing a multimodal interface to a cartography application. The main features of our work are the typology of types of cooperation that we propose and the capacity of our multimodal module to provide multimodal recognition scores.

We plan to improve the CARTOON system in the following directions:

- make user studies to test the advantages of multimodal recognition scores and to evaluate the types of cooperation that are used by the user
- develop linguistic and semantic representations (which are currently missing in our work) : we plan to connect our multimodal module to the linguistic tools developed by [Briffault et al. 1997] and test several possibilities of interaction such as early dropping of linguistic hypothesis due to multimodal results
- extend the gesture modality to circling and trajectory gestures on a tactile screen

More generally, what should be an "intelligent" multimodal system ? We propose hereafter some answers to this question. It should:

- recognize several input modalities (speech, hand and body gesture, gaze)
- generate contextual output modalities (speech, displayed text and graphics) depending on the users profile, behavior and environment
- be intuitive to use
- integrate multi-users dialogues mediated by the computer
- manipulate semantic representations
- find out dynamically the most important goal of cooperation between modalities depending on the user and environmental features

- dynamically select (these three questions have to be tackled together):
- the information to be transmitted
- the modalities to be used (and hence the media)
- the types of cooperation between modalities to be used

Acknowledgments

The author would like to thank Marie-Rose Goncalves and Xavier Briffault for the cartographic application they have developed and which is used within the CARTOON project.

References

- [André and Rist 1995] André, E. and Rist, T. Generating coherent presentations employing textual and visual material. *Artificial Intelligence Review*, 9 (2-3), 147-165.
- [Bellalem and Romary 1995] Bellalem, N. and Romary, L. Reference interpretation in a multimodal environment combining speech and gesture. In [IMMI 1995].
- [Béroule 1985] Béroule, D. (1985). A model of Adaptive Dynamic Associative Memory for speech processing. Thesis, 31 May, Univ. Orsay. 185p. In French.
- [Bolt 1980] Bolt, R.A. "Put-That-There": Voice and Gesture at The Graphics Interface. *Computer Graphics* 14 (3):262-270.
- [Bourdote et al. 1995] Bourdote, P., Krus, M., Gherbi, R. Management of non-standard devices for multimodal user interfaces under UNIX/X11. In [CMC 1995].
- [Bressole et al. 1995] Bressole, M.C, Pavard, B., Leroux, M. The role of multimodal communication in cooperation and intention recognition: the case of air traffic control. In [CMC 1995].
- [Goncalves et al. 1997] <http://www.limsi.fr/Individu/goncalve/index.html>
<http://www.limsi.fr/Individu/xavier/index.html>
- [Carbonnel et al. 1970] Carbonnel, J.R. Mixed-Initiative Man-Computer Dialogues. Bolt, Beranek and Newman (BBN) Report N 1971, Cambridge, MA.
- [Cheyer and Julia 1995] Cheyer, A. and Julia, L. Multimodal maps: an agent-based approach. In [CMC 1995].
- [CMC 1995]. Proceedings of the International Conference on Cooperative Multimodal Communication (CMC'95). Bunt, H, Beun, R.J. and Borghuis, T. (Eds.). Eindhoven, May 24-26.
- [Dowell et al. 1995] Dowell, J.; Shmueli, Y.; and Salter, I. Applying a cognitive model of the user to the design of a multimodal speech interface. In [IMMI 1995].
- [Foote et al. 1995] Foote, J.T.; Brown, M.G.; Jones, G.J.F.; Sparck Jones, K.; and Young, S.J. Video mail retrieval by voice: towards intelligent retrieval and browsing of multi-media documents. In [IMMI 95].
- [Briffault et al. 1997] <http://www.limsi.fr/Individu/xavier/index.html> <http://www.limsi.fr/Individu/vap/index.html>
- [Hare et al. 1995] Hare, M.; Doubleday, A., Bennett, I.; and Ryan, M. Intelligent presentation of information retrieved from heterogeneous multimedia databases. In [IMMI 1995].
- [Huls and Bos 1995] Huls, C. and Bos, E. Studies into full integration of language and action. In [CMC 1995].
- [IMMI 1995] Pre-Proceedings of the First International Workshop on Intelligence and Multimodality in Multimedia Interfaces: Research and Applications. Edited by John Lee. University of Edinburgh, Scotland, July 13-14.
- [Jackendoff 1987]. Jackendoff, R. On beyond zebra: the relation between linguistic and visual information. *Cognition* 26(2):89-114.
- [Martin et al. In press] Martin, J.C., Veldman, R. and Béroule, D. Developing Multimodal Interfaces : A theoretical Framework and Guided Propagation Networks. Book following the [CMC 1995] workshop. Bunt, H. (Ed.)
- [Maybury 1994] Maybury, M. Introduction. In *Intelligent multimedia interfaces*. AAAI Press. Cambridge Mass.
- [O'Nuallain and Smith 1994] O'Nuallain, S. and Smith, A.G. An investigation into the common semantics of language and vision. *Artificial Intelligence Review* 8 (2-3):113-122.

- [Salisbury et al. 1990] Salisbury M.W.; Hendrickson, J.H.; Lammers, T.L.; Fu, C.; and Moody, S.A. Talk and draw: bundling speech and graphics. *IEEE Computer.*, 23(8) 59-65.
- [Santana and Pineda 1995] Santana, S. and Pineda, L.A. Producing coordinated natural language and graphical explanations in the context of a geometric problem-solving task. In [IMMI 1995].
- [Sims and Hedberg 1995] Sims, R. and Hedberg, J. Dimensions of learner control: a reappraisal of interactive multi-media instruction. In [IMMI 1995].
- [Siroux et al. 1995] Siroux, J., Guyomard, M., Multon, F., Remondeau, C. Modeling and processing of the oral and tactile activities in the Georal tactile system. In [CMC 1995].
- [Veldman 1995] Experiments on robust parsing in a multi-modal Guided Propagation Network. ERASMUS Report. LIMSI.
- [Wang et al. 1993] Wang, E.; Shahnvaz, H.; Hedman, L.; Papadopoulos, K.; and Watkinson. A usability evaluation of text and speech redundant help messages on a reader interface. In G. Salvendy M. Smith (Eds.), *Human-Computer Interaction: Software and Hardware Interfaces*. pp 724-729.
- [Webber 1995] Webber, B. Instructing Animated Agents: Viewing Language in Behavioural Terms. In [CMC 1995].
- [Westerlund et al. 1994] Westerlund, P., Bérroule, D and Roques, M. Experiments of robust parsing using a Guided Propagation Network. *Proc. of the International Conf. on New Methods in Language Processing (NEMLAP)*, sept. 14-16, Manchester.

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.

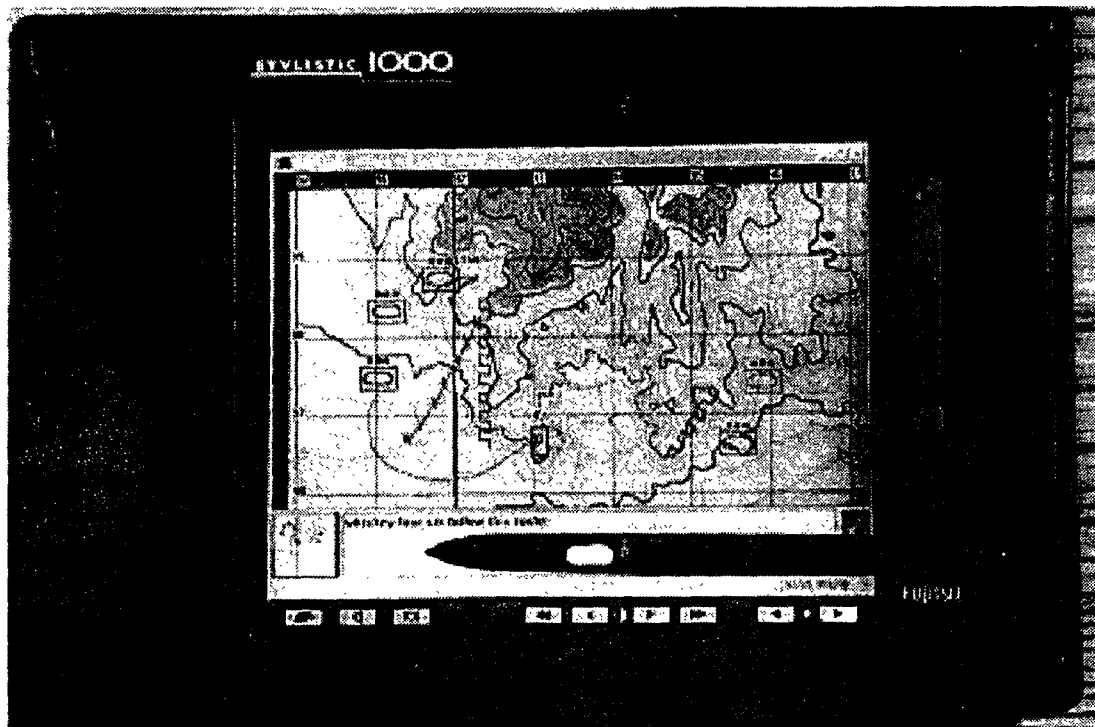


Figure 2: The QuickSet user interface

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.

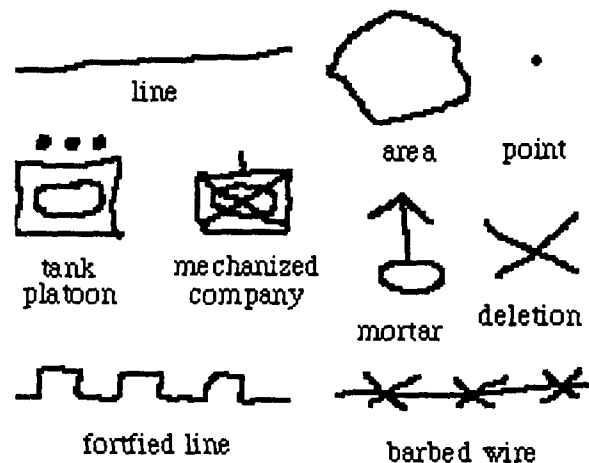


Figure 3: Example symbols and gestures

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.

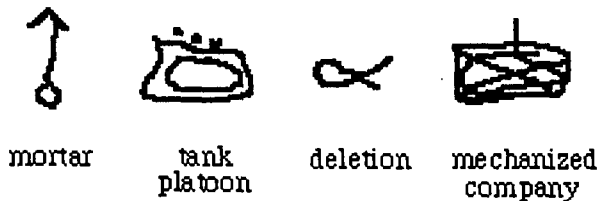


Figure 5: Typical pen input from real users

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*.

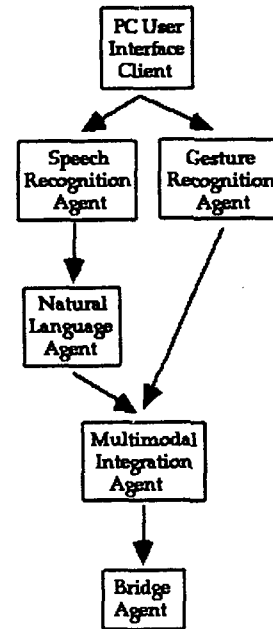


Figure 6: Multimodal integration architecture

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.

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

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.

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

Figure 8: Point interpretation of gesture

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

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).

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

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.

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

Figure 11: Feature structure for 'BARBED WIRE'

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

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 'MIA1 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

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

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).

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

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. Gorfu, 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.

DECLARATION OF HARRY BUNT CONCERNING THE INTERNATIONAL CONFERENCE ON
COOPERATIVE MULTIMODAL COMMUNICATION (CMC /95) IN EINDHOVEN, MAY 24-26, 1995 AND
THE PUBLICATION OF PAPERS PRESENTED AT THE CONFERENCE

I, Harry Bunt, declare as follows:

1. I am over the age of 18, have never been convicted of a felony or crime of moral turpitude and am legally competent to make this declaration. I have personal knowledge of the matters stated herein.
2. I am a Professor at Tilburg University in the Netherlands.
3. I have been employed by Tilburg University for over 34 years.
4. In my position, I research and teach in the area of language and artificial intelligence, including multimodal human-human and human-computer interaction and natural language parsing and generation.
5. I served as the Chairman of the First International Conference on Cooperative Multimodal Communication in Eindhoven, The Netherlands in May of 1995 (“CMC/95”). The conference was held at the Institute of Perception Research at Eindhoven.
6. CMC /95 was attended by at least 50 people. All of the attendees of CMC /95 were active participants in the area of multimodal communications.
7. I was the main organizer of CMC/95. I was personally involved in all aspects of the proceedings, from organizing, inviting authors to submit papers, overseeing the review of

papers, attending the conference, and the publication of the papers received for the conference, including the distribution of those papers to the attendees of the conference.

8. I am familiar with the paper submitted by Adam Cheyer and Luc Julia of SRI International to CMC/95, and presented at CMC/95 entitled “Multimodal Maps: An Agent-Based Approach.” (hereinafter, “Cheyer”). Cheyer is attached as Exhibit 1 to this declaration.
9. I was personally involved in receiving the Cheyer article, in overseeing the article being reviewed by the program committee, and in its publication and dissemination at the CMC /95 conference to persons interested in the field.
10. The Cheyer article was submitted as part of the proceedings and collected in a publication with other papers and published as Bunt, H. C., Beun, R. J., & Borghuis, V. A. J. (Eds.) (1995), “Proceedings of the international conference on cooperative multimodal communication CMC/95, Eindhoven, May 24-26, 1995”, Eindhoven: Samenwerkingsorgaan Brabantse Universiteiten, (hereinafter “1995 Proceedings Publication”), as shown in Exhibit 1.
11. I am familiar with the process for publication for the 1995 Proceedings Publication which included the Cheyer article, as I personally was involved with the publication and am listed as one of the editors of the 1995 Proceedings Publication. The 1995 Proceedings Publication was published by the Eindhoven University of Technology at the CMC/95 conference before the beginning of the conference. In particular, the 1995 Proceedings Publication was distributed to all attendees of the conference at the time of the

conference. Thus, the publication, including the Cheyer article, was available to all attendees of the conference no later than May 24, 1995.

12. Additionally, copies of the 1995 Proceedings Publication including the Cheyer article were available from the Eindhoven University of Technology library to interested persons in the field. From 1976 to 1983, I was employed at the Institute for Perception Research. From 1983 to the present, I have been employed by Tilburg University as a Professor. The Institute for Perception Research was a joint venture of the Eindhoven University of Technology and Philips Research. Even after 1983, I closely cooperated with the Institute for Perception Research and have personal knowledge of the library of the Institute for Perceptual Research at Eindhoven and Eindhoven University of Technology with respect to its indexing, cataloging, keeping, and public availability of the 1995 Proceedings Publication. Based on the markings on page 2 of Exhibit 1, the marking “*9690453*” indicates that the work was received, indexed, and cataloged by the Institute for Perceptual Research at Eindhoven at least by 1996 because the first two numbers indicate the year. The library at the Institute for Perceptual Research at Eindhoven was open to the public, including those of ordinary skill in the art. The library maintained a catalog of publications that allowed searching on title, author, or keyword. The 1995 Proceedings Publication entry in the catalog included the following keywords listed on page 3 of Exhibit 1: mens-machine communicatie, multimedia, user-interfaces. Based on my knowledge and records, the 1995 Proceedings Publication was available at the library from at least 1996 until the Institute for Perceptual Research was closed in 2001. Thereafter, the publication was transferred to the main library of the Eindhoven University of Technology. Attached as Exhibit 2 is a copy of the library records for the

1995 Proceeding Publication. Based on my knowledge and experience at the Institute for Perceptual Research at Eindhoven, the library records in Exhibit 2 are complete and correct. Thus, the 1995 Proceeding Publications have been continuously available from the Institute for Perceptual Research at Eindhoven and Eindhoven University of Technology from 1996 to the present. Currently the 1995 Proceedings Publication has been converted to digital form and can be found and downloaded freely over the Internet at <https://pure.tue.nl/ws/files/4264441/466003-1.pdf>.

13. CMC /95 was announced through the commonly used channels for conference announcements. Some of the most prominent researchers in the field of multimodal communication were involved in papers presented at the conference, including Norman Badler, Catherine Pelachaud, Justine Cassel, Mark Steedman, Jaques Siroux, Marc Guyomard, Ingrid Zukerman, Jean-Claude Martin, Michael McTear, Susan Luperfory, and John Lee, and represented several of the major research centers in this field, such as Webber's group at University of Pennsylvania, MIT Media Lab, Zukerman's group at Monash University in Australia, SRI International in the United States, LIMSI in Orsay, France, Informatics at University of Ulster in Northern Ireland, HCRC in Edinburgh, ATR Labs in Kyoto, NTT Labs in Kanagawa, Japan, and IBM Research Lab in Yorktown Heights, among other institutions and researchers. Prominent researchers were involved in the conference as reviewers, including Walter von Hahn, and Ray Perrault from SRI International. This meant that people working in the field were well aware of the CMC/95 conference.

14. People working in the areas of multimodal communications in 1995 would have been aware of the CMC /95 conference because the number of researchers and developers

working in the field of multimodal communication and spoken language technologies at that time was not very large. The papers submitted for the conference were authored by researchers and developers working in the field, and peer-reviewed by those in the field.

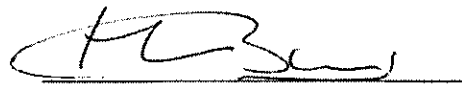
15. The Cheyer article was publicly available by May of 1995, including to researchers in the field of natural language processing and multimodal communication no later than late May of 1995.
16. In 1995, I was generally familiar with the work of SRI International, especially in the areas of natural language processing and multimodal communication.
17. In 1995, SRI International was generally known by those of skill in the art to have been involved in natural language processing and multimodal communication. It would have been common for one in the field of natural language processing and/or multimodal communication to review and reference SRI International publications, technical documents, and conference presentations as a source of information.
18. A selection of the papers of the CMC/95 proceedings were also published in 1998 in *Multimodal Human Computer Communication: Systems, Techniques, and Experiments* [selected papers from the First International Conference on Cooperative Multimodal Communication, Eindhoven, May 1995], Harry C. Bunt, Robbert-Jan Beun and Tijn Borghuis eds., in *Lecture Notes in Artificial Intelligence 1374* (Berlin: Springer, 1998), (hereinafter, "1998 Proceedings Publication") of which the same Cheyer article is at pp. 111-121. I was personally involved in the 1998 Proceedings Publication and am familiar with its contents and its publications. The contents of the 1998 Proceedings Publication

in some cases vary from the 1995 Proceedings Publication, to reflect revisions or updates to the original 1995 versions of those papers.

19. The 1998 Proceedings Publication was widely distributed to persons skilled in the field and was publicly available by the end of 1998.

All statements made herein of my own knowledge are true and all statements made on information and belief are believed to be true. I further understand that willful false statements and the like are punishable by fine or imprisonment, or both under Section 1001 of Title 18 of the United States Code. I declare under penalty of perjury of the laws of the United States that the foregoing is true and correct.

Executed on December 18, 2017 in Driebergen, the Netherlands,

A handwritten signature in black ink, appearing to read 'H. Bunt', written over a horizontal line.

Harry C. Bunt

EXHIBIT 1

Proceedings of the international conference on cooperative multimodel communication CMC/95, Eindhoven, May 24-26, 1995

Bunt, H.C.; Beun, R.J.; Borghuis, V.A.J.

Published: 01/01/1995

Document Version

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the author's version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

Citation for published version (APA):

Bunt, H. C., Beun, R. J., & Borghuis, V. A. J. (Eds.) (1995). Proceedings of the international conference on cooperative multimodel communication CMC/95, Eindhoven, May 24-26, 1995: proceedings. (CMC : cooperative multimodel communication : international conference; Vol. 1). Eindhoven: DENK: Samenwerkingsorgaan Brabantse Universiteiten.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 22. Nov. 2017

**Proceedings of the
International Conference on
Cooperative Multimodal
Communication CMC/95
Part I**

Eindhoven, May 24-26, 1995

Harry Bunt, Robbert-Jan Beun & Tijn Borghuis (eds.)

I P O



9690453

E I N D H O V E N

**ex
32/1**

**BIBLIOTHEEK
INSTITUUT VOOR PERCEPTIE
ONDERZOEK**

CIP GEGEVENS KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Harry Bunt, Robbert-Jan Beun & Tijn Borghuis

Proceedings of the International Conference on Cooperative
Multimodal Communication, Eindhoven, May 24-26, 1995

ISBN 90-9008315-4

trefw.: mens-machine communicatie, multimedia, user-interfaces

Preface

Communication is a bidirectional activity that comes naturally in multimodal form, involving both verbal and nonverbal, vocal, visual, tactile and other means of interaction. Natural communication is also cooperative, in that the participants make an effort to understand each other, and act in a way that takes each other's goals and purposes into account, for instance helping a dialogue partner to obtain relevant information.

Technical developments increasingly allow the realization of human-computer interfaces where more sophisticated forms of visual and auditory, verbal and nonverbal information are used by the computer and where the user is allowed a greater variety of forms of expression. Two crucial aspects of natural communication are, however, still conspicuously absent in existing user interfaces:

- real cooperation from the part of the computer, based on a good understanding of the user's wants;
- true multimodality in the sense of fully integrated, simultaneous use of several modalities to convey a complex message.

As a result, human-computer communication is generally felt to be only marginally cooperative, and to be unnatural and primitive, compared to natural human communication.

The present conference aims at contributing to improving the state of the art in cooperative multimodal human-computer communication, bringing together researchers involved in the design, implementation, and application of forms of cooperative human-computer communication where natural language (typed or spoken) is used in combination with other modalities, such as visual feedback and direct manipulation. The conference focuses on formal, computational, and user aspects of building cooperative multimodal dialogue systems, with the following topics being identified in the call for papers:

- cooperativity in multimodal dialogue
- natural language semantics in a multimodal context
- formal and computational models of dialogue context
- incremental knowledge representation and dialogue
- interacting with visual domain representations
- collaborative problem solving
- constraint-based approaches to animation and visual modelling
- effective use of different interactive modalities
- modelling temporal aspects of multimodal communication
- type theory and natural language interpretation

In response to the call for papers, we have received submissions from all over the world (Europe, North America, Asia, Australia), from which the programme committee has selected 17 for paper presentation and 8 for poster presentation at the conference. In addition, the conference features a presentation of the multimodal DenK-project, which has provided the inspiration for organizing this conference, and invited papers by Mark Maybury, Wolfgang Wahlster, Bonnie Webber and Kent Wittenburg.

I would like to use this occasion to thank the members of the programme committee for reviewing the submitted contributions for the conference, and the members of the organizing committee plus the staff at the Institute for Perception Research IPO, which hosts the conference, for all their efforts to make the conference run smoothly. Particular thanks are due to the Samenwerkingsorgaan Brabantse Universiteiten (the organization for cooperation between the universities in the province of Brabant, i.e. the universities of Tilburg and Eindhoven), and to the Royal Dutch Academy of Sciences (KNAW) for their financial support.

Harry Bunt
Program Committee chairman.

Program Committee

Harry Bunt (chair)

Norman Badler

Walther von Hahn

Hans Kamp

Joseph Mariani

Paul Mc Kevitt

Kees van Overveld

Donia Scott

Bonnie Webber

Jeroen Groenendijk

Dieter Huber

John Lee

Mark Maybury

Rob Nederpelt

Ray Perrault

Wolfgang Wahlster

Kent Wittenburg

Organizing Committee

Robbert-Jan Beun (chair)

Tijn Borghuis

Harry Bunt

Rob Nederpelt

Marianne Wagemans

Sponsorship

Koninklijke Nederlandse Akademie van Wetenschappen (KNAW)

Samenwerkingsorgaan Brabantse Universiteiten (SOBU)

ACL Special Interest Group in Multimedia (SIGMEDIA)

Table of Contents

PART I

Invited Papers

Toward Cooperative Multimedia Interaction (abstract) 3
Mark T. Maybury

Instructing Animated Agents: Viewing Language in Behavioral Terms 5
Bonnie Webber

Visual Language Parsing: If I Had a Hammer... 17
Kent Wittenburg

Submitted Papers

Contexts in Dialogue 37
Tijn Borghuis

**Management of Non-Standard Devices for Multimodal User Interfaces
under UNIX/X11** 49
Patrick Bourdot, Mike Krus and Rachid Gherbi

**The Role of Multimodal Communication in Cooperation and Intention
Recognition: The Case of Air Traffic Control** 63
Marie-Christine Bressolle, Bernard Pavard and Marcel Leroux

Cooperative Multimodal Communication in the DenK Project 79
Harry Bunt, René Ahn, Robbert-Jan Beun, Tijn Borghuis and Kees van Overveld

Multimodal Maps: An Agent Based Approach 103
Adam Cheyer and Luc Julia

Object Reference During Task-related Terminal Dialogues 115
Anita Cremers

**Speakers' Responses to Requests for Repetition in a
Multimedia Language Processing Environment** 129
Laurel Fais, Kyung-ho Loken-Kim and Young-Duk Park

A Cooperative Approach for Multimodal Presentation Planning 145
Yi Han and Ingrid Zukerman

PART II

| | |
|--|-----|
| Studies into Full Integration of Language and Action | 161 |
| <i>Carla Huls and Edwin Bos</i> | |
| Referent Identification Requests in Multimodal Dialogues | 175 |
| <i>Tsuneaki Kato and Yukiko I. Nakano</i> | |
| Anaphora in Multimodal Discourse | 193 |
| <i>John Lee and Keith Stenning</i> | |
| Towards Adequate Representation Technologies for Multimodal Interfaces . | 207 |
| <i>Jean Claude Martin, Remko Veldman and Dominique Béroule</i> | |
| Designing a Multimedia Interface for Operators Assembling Circuit Boards | 225 |
| <i>Fergal McCaffery, Michael McTear and Maureen Murphy</i> | |
| Synthesizing Cooperative Conversation | 237 |
| <i>Catherine Pelachaud, Justine Cassell, Norman Badler, Mark Steedman, Scott Prevost and Matthew Stone</i> | |
| Topic Management in Information Dialogues | 257 |
| <i>Mieke Rats</i> | |
| An Approach to Solving the Symbol Grounding Problem: Neural Networks for Object Naming and Retrieval | 273 |
| <i>N.J. Sales and R.G. Evans</i> | |
| Modeling and Processing of the Oral and Tactile Activities in the Georal Tactile System | 287 |
| <i>J. Siroux, M. Guyomard, F. Multon and C. Remondeau</i> | |
| Poster presentations | |
| The Generalized Display Processor: a Platform for Real-time Interactive Computer Animation | 299 |
| <i>Gino van Bergen and Kees van Overveld</i> | |
| Communication and Co-ordination Difficulties During Interactive Tele-Teaching in the Independent Problem-Solving Format | 301 |
| <i>Martin Colbert</i> | |
| Automatic Generation of Statistical Graphics | 303 |
| <i>Massimo Fasciano and Guy Lapalme</i> | |

| | |
|---|-----|
| Computer-Aided Negotiation Support in Hypermedia Multi-Agent Systems | 307 |
| <i>Igor V. Kotenko and Dmitry L. Krechman</i> | |
| Multimodal Dialogue Semantics Against a Dynamic World Model | 311 |
| <i>Susann Luperfoy and David Duff</i> | |
| Two Basic Orientations of Subject in World and in Human-Computer Communication | 315 |
| <i>Olga I. Marchenko</i> | |
| Internalized Contexts in NL Semantics | 317 |
| <i>Wlodek Zadrozny</i> | |
| Author Index | 321 |
| Subject Index | 323 |

Invited Papers

Multimodal Maps: An Agent-based Approach

Adam Cheyer and Luc Julia

SRI International
333 Ravenswood Ave
Menlo Park, CA 94025 - USA

Abstract

In this paper, we discuss how multiple input modalities may be combined to produce more natural user interfaces. To illustrate this technique, we present a prototype map-based application for a travel planning domain. The application is distinguished by a synergistic combination of handwriting, gesture and speech modalities; access to existing data sources including the World Wide Web; and a mobile handheld interface. To implement the described application, a hierarchical distributed network of heterogeneous software agents was augmented by appropriate functionality for developing synergistic multimodal applications.

Key words: Multimodal Interface, Agent Architecture, Distributed Artificial Intelligence.

1 Introduction

As computer systems become more powerful and complex, efforts to make computer interfaces more simple and natural become increasingly important. Natural interfaces should be designed to facilitate communication in ways people are already accustomed to using. Such interfaces allow users to concentrate on the tasks they are trying to accomplish, not worry about what they must do to control the interface.

In this paper, we begin by discussing what input modalities humans are comfortable using when interacting with computers, and how these modalities should best be combined in order to produce natural interfaces. In section three, we present a prototype map-based application for the travel planning domain which uses a synergistic combination of several input modalities. Section four describes the agent-based approach we used to implement the application and the work on which it is based. In section five, we summarize our conclusions and future directions.

2 Natural Input

2.1 Input Modalities

Direct manipulation interface technologies are currently the most widely used techniques for creating user interfaces. Through the use of menus and a graphical user interface, users are presented with sets of discrete actions and the objects on which to perform them. Pointing

devices such as a mouse facilitate selection of an object or action, and drag and drop techniques allow items to be moved or combined with other entities or actions.

With the addition of electronic pen devices, gestural drawings add a new dimension direct manipulation interfaces. Gestures allow users to communicate a surprisingly wide range of meaningful requests with a few simple strokes. Research has shown that multiple gestures can be combined to form dialog, with rules of temporal grouping overriding temporal sequencing [22]. Gestural commands are particularly applicable to graphical or editing type tasks.

Direct manipulation interactions possess many desirable qualities: communication is generally fast and concise; input techniques are easy to learn and remember; the user has a good idea about what can be accomplished, as the visual presentation of the available actions is generally easily accessible. However, direct manipulation suffers from limitations when trying to access or describe entities which are not or can not be visualized by the user.

Limitations of direct manipulation style interfaces can be addressed by another interface technology, that of natural language interfaces. Natural language interfaces excel in describing entities that are not currently displayed on the monitor, in specifying temporal relations between entities or actions, and in identifying members of sets. These strengths are exactly the weaknesses of direct manipulation interfaces, and concurrently, the weaknesses of natural language interfaces (ambiguity, conceptual coverage, etc.) can be overcome by the strengths of direct manipulation.

Natural language content can be entered through different input modalities, including typing, handwriting, and speech. It is important to note that, while the same textual content can be provided by the three modalities, each modality has widely varying properties.

- Spoken language is the modality used first and foremost in human-human interactive problem solving [4]. Speech is an extremely fast medium, several times faster than typing or handwriting. In addition, speech input contains content that is not present in other forms of natural language input, such as prosody, tone and characteristics of the speaker (age, sex, accent).
- Typing is the most common way of entering information into a computer, because it is reasonably fast, very accurate, and requires no computational resources.
- Handwriting has been shown to be useful for certain types of tasks, such as performing numerical calculations and manipulating names which are difficult to pronounce [18, 19]. Because of its relatively slow production rate, handwriting may induce users to produce different types of input than is generated by spoken language; abbreviations, symbols and non-grammatical patterns may be expected to be more prevalent amid written input.

2.2 Combination of Modalities

As noted in the previous section, direct manipulation and natural language seem to be very complementary modalities. It is therefore not surprising that a number of multimodal systems combine the two.

Notable among such systems is the Cohen's Shoptalk system [6], a prototype manufacturing and decision-support system that aids in tasks such as quality assurance monitoring, and production scheduling. The natural language module of Shoptalk is based on the Chat-85



Figure 1: Multimodal Application for Travel Planning

natural language system [25] and is particularly good at handling time, tense, and temporal reasoning.

A number of systems have focused on combining the speed of speech with the reference provided by direct manipulation of a mouse pointer. Such systems include the XTRA system [1], CUBRICON [15], the PAC-Amodeus model [16], and TAPAGE [9].

XTRA and CUBRICON are both systems that combine complex spoken input with mouse clicks, using several knowledge sources for reference identification. CUBRICON's domain is a map-based task, making it similar to the application developed in this paper. However, the two are different in that CUBRICON can only use direct manipulation to indicate a specific item, whereas our system produces a richer mixing of modalities by adding both gestural and written language as input modalities.

The PAC-Amodeus systems such as VoicePaint and Notebook allow the user to synergistically combine vocal or mouse-click commands when interacting with notes or graphical objects. However, due to the selected domains, the natural language input is very simple, generally of the style "Insert a note here."

TAPAGE is another system that allows true synergistic combination of spoken input with direct manipulation. Like PAC-Amodeus, TAPAGE's domain provides only simple linguistic input. However, TAPAGE uses a pen-based interface instead of a mouse, allowing gestural commands. TAPAGE, selected as a building block for our map application, will be described more in detail in section 4.2.

Other interesting work regarding the simultaneous combination of handgestures and gaze can be found in [2, 13].

3 A Multimodal Map Application

In this section, we will describe a prototype map-based application for a travel planning domain. In order to provide the most natural user interface possible, the system permits the

user to simultaneously combine direct manipulation, gestural drawings, handwritten, typed and spoken natural language. When designing the system, other criteria were considered as well:

- The user interface must be light and fast enough to run on a handheld PDA while able to access applications and data that may require a more powerful machine.
- Existing commercial or research natural language and speech recognition systems should be used.
- Through the multimodal interface, a user must be able to transparently access a wide variety of data sources, including information stored in HTML form on the World Wide Web.

As illustrated in Figure 1, the user is presented with a pen sensitive map display on which drawn gestures and written natural language statements may be combined with spoken input. As opposed to a static paper map, the location, resolution, and content presented by the map change, according to the requests of the user. Objects of interest, such as restaurants, movie theaters, hotels, tourist sites, municipal buildings, etc. are displayed as icons. The user may ask the map to perform various actions. For example :

- *distance calculation* : e.g. "How far is the hotel from Fisherman's Wharf?"
- *object location* : e.g. "Where is the nearest post office?"
- *filtering* : e.g. "Display the French restaurants within 1 mile of this hotel."
- *information retrieval* : e.g. "Show me all available information about Alcatraz."

The application also makes use of multimodal (multimedia) output as well as input: video, text, sound and voice can all be combined when presenting an answer to a query.

During input, requests can be entered using gestures (see Figure 2 for sample gestures), handwriting, voice, or a combination of pen and voice. For instance, in order to calculate the distance between two points on the map, a command may be issued using the following:

- *gesture*, by simply drawing a line between the two points of interest.
- *voice*, by speaking "What is the distance from the post office to the hotel?"
- *handwriting*, by writing "dist p.o. to hotel?"
- *synergistic combination of pen and voice*, by speaking "What is the distance from here to this hotel?" while simultaneously indicating the specified locations by pointing or circling.

Notice that in our example of synergistic combination of pen and voice, the arguments to the verb "distance" can be specified before, at the same time, or shortly after the vocalization of the request to calculate the distance. If a user's request is ambiguous or underspecified, the system will wait several seconds and then issue a prompt requesting additional information.

The user interface runs on pen-equipped PC's or a Dauphin handheld PDA ([7]) using either a microphone or a telephone for voice input. The interface is connected either by

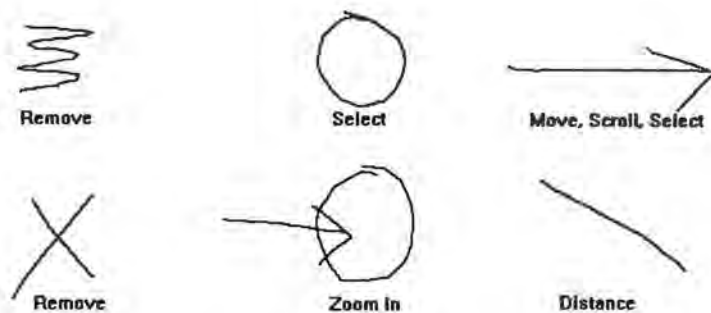


Figure 2: Sample gestures

modem or ethernet to a server machine which will manage database access, natural language processing and speech recognition for the application. The result is a mobile system that provides a synergistic pen/voice interface to remote databases.

In general, the speed of the system is quite acceptable. For gestural commands, which are handled locally on the user interface machine, a response is produced in less than one second. For handwritten commands, the time to recognize the handwriting, process the English query, access a database and begin to display the results on the user interface is less than three seconds (assuming an ethernet connection, and good network and database response). Solutions to verbal commands are displayed in three to five seconds after the end of speech has been detected; partial feedback indicating the current status of the speech recognition is provided earlier.

4 Approach

In order to implement the application described in the previous section, we chose to augment a proven agent- based architecture with functionalities developed for a synergistically multimodal application. The result is a flexible methodology for designing and implementing distributed multimodal applications.

4.1 Building Blocks

4.1.1 Open Agent Architecture

The Open Agent Architecture (OAA) [5] provides a framework for coordinating a society of agents which interact to solve problems for the user. Through the use of agents, the OAA provides distributed access to commercial applications, such as mail systems, calendar programs, databases, etc.

The Open Agent Architecture possesses several properties which make it a good candidate for our needs:

- An Interagent Communication Language (ICL) and Query Protocol have been developed, allowing agents to communicate among themselves. Agents can run on different platforms and be implemented in a variety of programming languages.
- Several natural language systems have been integrated into the OAA which convert English into the Interagent Communication Language. In addition, a speech recognition

agent has been developed to provide transparent access to the Corona speech recognition system.

- The agent architecture has been used to provide natural language and agent access to various heterogeneous data and knowledge sources.
- Agent interaction is very fine-grained. The architecture was designed so that a number of agents can work together, when appropriate in parallel, to produce fast responses to queries.

The architecture for the OAA, based loosely on Schwartz's FLiPSiDE system[23], uses a hierarchical configuration where client agents connect to a "facilitator" server. Facilitators provide content-based message routing, global data management, and process coordination for their set of connected agents. Facilitators can, in turn, be connected as clients of other facilitators. Each facilitator records the published functionality of their sub-agents, and when queries arrive in Interagent Communication Language form, they are responsible for breaking apart any complex queries and for distributing goals to the appropriate agents. An agent solving a goal may require supporting information and the agent architecture provides numerous means of requesting data from other agents or from the user.

Among the assortment of agent architectures, the Open Agent Architecture can be most closely compared to work by the ARPA knowledge sharing community [10]. The OAA's query protocol, Interagent Communication Language and Facilitator mechanisms have similar instantiations in the SHADE project, in the form of KQML, KIF and various independent capability matchmakers. Other agent architectures, such as General Magic's Telescript [11], MASCOS [20], or the CORBA distributed object approach [17] do not provide as fully developed mechanisms for interagent communication and delegation.

The Open Agent Architecture provides capability for accessing distributed knowledge sources through natural language and voice, but it is lacking integration with a synergistic multimodal interface.

4.1.2 TAPAGE

TAPAGE (edition de Tableaux par la Parole et la Geste) is a synergistic pen/voice system for designing and correcting tables.

To capture signals emitted during a user's interaction, TAPAGE integrates a set of modality agents, each responsible for a very specialized kind of signal [9]. The modality agents are connected to an "interpret agent" which is responsible for combining the inputs across all modalities to form a valid command for the application. The interpret agent receives filtered results from the modality agents, sorts the information into the correct fields, performs type-checking on the arguments, and prompts the user for any missing information, according to the model of the interaction. The interpret agent is also responsible for merging the data streams sent by the modality agents, and for resolving ambiguities among them, based on its knowledge of the application's internal state. Another function of the interpret agent is to produce reflexes: reflexes are actions output at the interface level without involving the functional core of the application.

The TAPAGE system can accept multimodal input, but it is not a distributed system; its functional core is fixed. In TAPAGE, the set of linguistic input is limited to a *verb object argument* format.

4.2 Synthesis

In the Open Agent Architecture, agents are distributed entities that can run on different machines, and communicate together to solve a task for the user. In TAPAGE, agents are used to provide streams of input to a central interpret process, responsible for merging incoming data. A generalization of these two types of agents could be :

Macro Agents: contain some knowledge and ability to reason about a domain, and can answer or make queries to other macro agents using the Interagent Communication Language.

Micro Agents: are responsible for handling a single input or output data stream, either filtering the signal to or from a hierarchically superior "interpret" agent.

The network architecture that we used was hierarchical at two resolutions – micro agents are connected to a superior macro agent, and macro agents are connected in turn to a facilitator agent. In both cases, a server is responsible for the supervision of its client sub-agents.

In order to describe our implementation, we will first give a description of each agent used in our application and then illustrate the flow of communication among agents produced by a user's request.

Speech Recognition (SR) Agent: The SR agent provides a mapping from the Interagent Communication Language to the API for the Decipher (Corona) speech recognition system [4], a continuous speech speaker independent recognizer based on Hidden Markov Model technology. This macro agent is also responsible for supervising a child micro agent whose task is to control the speech data stream. The SR agent can provide feedback to an interface agent about the current status and progress of the micro agent (e.g. "listening", "end of speech detected", etc.) This agent is written in C.

Natural Language (NL) Parser Agent: translates English expressions into the Interagent Communication Language (ICL). For a more complete description of the ICL, see [5]. The NL agent we selected for our application is the simplest of those integrated into the OAA. It is written in Prolog using Definite Clause Grammars, and supports a distributed vocabulary; each agent dynamically adds word definitions as it connects to the network. A current project is underway to integrate the Gemini natural language system [4], a robust bottom up parser and semantic interpreter specifically designed for use in Spoken Language Understanding projects.

Database Agents: Database agents can reside at local or remote locations and can be grouped hierarchically according to content. Micro agents can be connected to database agents to monitor relevant positions or events in real time. In our travel planning application, database agents provide maps for each city, as well as icons, vocabulary and information about available hotels, restaurants, movies, theaters, municipal buildings and tourist attractions. Three types of databases were used: Prolog databases, X.500 hierarchical databases, and data loaded automatically by scanning HTML pages from the World Wide Web (WWW). In one instance, a local newspaper provides weekly updates to its Mosaic-accessible list of current movie times and reviews, as well as adding several new restaurant reviews to a growing collection; this information is extracted by an HTML reading database agent and made accessible to the agent architecture. Descriptions and addresses of new restaurants are presented to the user on request, and the user can choose to add them to the permanent database by specifying positional coordinates on the map (eg. "add this new restaurant here"), information lacking in the WWW database.

Reference Resolution Agent: This agent is responsible for merging requests arriving in parallel from different modalities, and for controlling interactions between the user interface

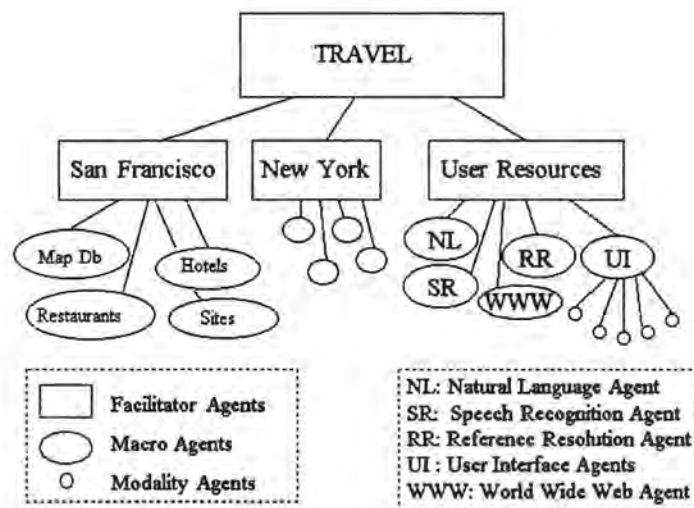


Figure 3: Agent Architecture for Map Application

agent, database agents and modality agents. In this implementation, the reference resolution agent is domain specific: knowledge is encoded as to what actions must be performed to resolve each possible type of ICL request in its particular domain. For a given ICL logical form, the agent can verify argument types, supply default values, and resolve argument references. Some argument references are descriptive (“How far is it to the hotel on Emerson Street?”); in this case, a domain agent will try to resolve the definite reference by sending database agent requests. Other references, particularly when contextual or deictic, are resolved by the user interface agent (“What are the rates for this hotel?”). Once arguments to a query have been resolved, this agent coordinates the actions and calculations necessary to produce the result of the request.

Interface Agent: This macro agent is responsible for managing what is currently being displayed to the user, and for accepting the user’s multimodal input. The Interface Agent also coordinates client modality agents and resolves ambiguities among them : handwriting and gestures are interpreted locally by micro agents and combined with results from the speech recognition agent, running on a remote speech server. The handwriting micro-agent interfaces with the Microsoft PenWindows API and accesses a handwriting recognizer by CIC Corporation. The gesture micro- agent accesses recognition algorithms developed for TAPAGE.

An important task for the interface agent is to record which objects of each type are currently salient, in order to resolve contextual references such as “the hotel” or “where I was before.” Deictic references are resolved by gestural or direct manipulation commands. If no such indication is currently specified, the user interface agent waits long enough to give the user an opportunity to supply the value, and then prompts the user for it.

We shall now give an example of the distributed interaction of agents for a specific query. In the following example, all communication among agents passes transparently through a facilitator agent in an undirected fashion; this process is left out of the description for brevity.

1. A user speaks: “How far is the restaurant from this hotel?”

2. The speech recognition agent monitors the status and results from its micro agent, sending feedback received by the user interface agent. When the string is recognized, a translation is requested.
3. The English request is received by the NL agent and translated into ICL form.
4. The reference resolution agent (RR) receives the ICL distance request containing one definite and one deictic reference and asks for resolution of these references.
5. The interface agent uses contextual structures to find what "the restaurant" refers to, and waits for the user to make a gesture indicating "the hotel", issuing prompts if necessary.
6. When the references have been resolved, the domain agent (RR) sends database requests asking for the coordinates of the items in question. It then calculates the distance according to the scale of the currently displayed map, and requests the user interface to produce output displaying the result of the calculation.

5 Conclusions

By augmenting an existing agent-based architecture with concepts necessary for synergistic multimodal input, we were able to rapidly develop a map-based application for a travel planning task. The resulting application has met our initial requirements: a mobile, synergistic pen/voice interface providing good natural language access to heterogeneous distributed knowledge sources. The approach used was general and should provide a for developing synergistic multimodal applications for other domains.

The system described here is one of the first that accepts commands made of synergistic combinations of spoken language, handwriting and gestural input. This fusion of modalities can produce more complex interactions than in many systems and the prototype application will serve as a testbed for acquiring a better understanding of multimodal input.

In the near future, we will continue to verify and extend our approach by building other multimodal applications. We are interested in generalizing the methodology even further; work has already begun on an agent-building tool which will simplify and automate many of the details of developing new agents and domains.

References

- [1] Allegayer, J, Jansen-Winkeln, R., Reddig, C. and Reithinger, N. "Bidirectional use of knowledge in the multi-modal NL access system XTRA". In Proceedings of IJCAI-89, Detroit, pp. 1492-1497.
- [2] Bolt, R. "Put that there: Voice and Gesture at the Graphic Interface". Computer Graphics, 14(3), 1980, pp. 262-270.
- [3] Bellik, Y. and Teil, D. "Les types de multimodalites", In Proc. IIM'92 (Paris), pp. 22-28.
- [4] Cohen, M., Murveit, H., Bernstein, J., Price, P., Weintraub, M., "The DECIPHER Speech Recognition System". 1990 IEEE ICASSP, pp. 77-80.

- [5] Cohen, P.R., Cheyer, A., Wang, M. and Baeg, S.C. "An Open Agent Architecture". In Proc. AAAI'94 - SA (Stanford), pp. 1-8.
- [6] Cohen, P. "The role of natural language in a multimodal interface". Proceedings of UIST'92, 143-149.
- [7] Dauphin DTR-1 User's Manual, Dauphin Technology, Inc. 337 E. Butterfield Rd., Suite 900, Lombard, Ill 60148.
- [8] Dowding, J., Gawron, J.M., Appelt, D., Bear, J., Cherny, L., Moore, B. and Moran D., "Gemini: A natural language system for spoken-language understanding", Technical Note 527, AI Center, SRI International, 333 Ravenswood Ave., Menlo Park, CA 94025, April 1993.
- [9] Faure, C. and Julia, L. "An Agent-Based Architecture for a Multimodal Interface". In Proc. AAAI'94 - IM4S (Stanford), pp. 82-86.
- [10] Genesereth, M. and Singh, N.P. "A knowledge sharing approach to software interoperability". Computer Science Department, Stanford University, unpublished ms., 1994.
- [11] General Magic, Inc., "Telescript Product Documentation", 1995.
- [12] Julia, L. and Faure, C. "A Multimodal Interface for Incremental Graphic Document Design". HCI International '93, Orlando.
- [13] Koons, D.B., Sparrell, C.J., and Thorisson, K.R. "Integrating Simultaneous Input from Speech, Gaze and Hand Gestures". In *Intelligent Multimedia Interfaces*, Edited by Mark Maybury, Menlo Park, CA, AAAI Press, 1993.
- [14] Maybury, M.T. (ed.), *Intelligent Multimedia Interfaces*, AAAI Press/MIT Press: Menlo Park, Ca, 1993.
- [15] Neal, J.G., and Shapiro, S.C. "Intelligent Multi-media Interface Technology". In *Intelligent User Interfaces*, Edited by J. Sullivan and S. Tyler, Addison-Wesley Pub. Co., Reading, MA, 1991.
- [16] Nigay, L. and Coutaz, J. "A Design Space for Multimodal Systems: Concurrent Processing and Data Fusion". In Proc. InterCHI'93 (Amsterdam), ACM Press, pp. 172-178.
- [17] Object Management Group, "The Common Object Request Broker: Architecture and Specification", OMG Document Number 91.12.1, December 1991.
- [18] Oviatt, S. "Toward Empirically-Based Design of Multimodal Dialogue Systems". In Proc. AAAI'94 - IM4S (Stanford), pp. 30-36.
- [19] Oviatt, S. and Olsen, E. "Integration Themes in Multimodal Human-Computer Interaction". Proceedings of ICSLP'94, Yokohama, pp. 551-554.
- [20] Park, S.K., Choi J.M., Myeong-Wuk J., Lee G.L., and Lim Y.H. "MASCOS : A Multi-Agent System as the Computer Secretary". Submitted for publication.
- [21] Pfaff, G. and Ten Hagen, P.J.W. *Seeheim workshop on User Interface Management Systems* (Berlin), Springer- Verlag.

- [22] Rhyne J. "Dialogue Management for Gestural Interfaces". *Computer Graphics*, 21(2), 1987, pp. 137-142.
- [23] Schwartz, D.G. "Cooperating heterogeneous systems: A blackboard-based meta approach". Technical Report 93-112, Center for Automation and Intelligent Systems Research, Case Western Reserve University, Cleveland Ohio, April 1993. Unpublished PhD. thesis.
- [24] Sullivan, J. and Tyler, S. (eds.), *Intelligent User Interfaces*, Addison-Wesley Pub. Co., Reading, MA, 1991.
- [25] Warren, D. and Pereira, F., "An Efficient Easily Adaptable System for Interpreting Natural Language Queries", in *American Journal of Computational Linguistics*, 8(3), 1982, pp. 110-123.
- [26] Wauchope, K., "Eucalyptus: Integrating Natural Language with a Graphical User Interface." Naval Research Laboratory Technical Report NRL/FR/5510-94-9711, in press, 1994.

EXHIBIT 2

Collection

Repository TU/e

Journal titles

Databases

New (e-)books

Home bib / Home Repository

Publications TU/e

Title Proceedings of the international conference on cooperative multimodel communication CMC/95, Eindhoven, May 24-26, 1995 : proceedings / ed. by Harry C. Bunt, Robbert-Jan Beun and Tijn Borghuis

Author H.C. Bunt [Editor]
R.J. Beun [Editor]
V.A.J. Borghuis [Editor]

Publisher Eindhoven : DENK: Samenwerkingsorgaan Brabantse Universiteiten, 1995

Pages 2 dl. (324 p.)

ISBN 90-9008315-4

Medialink <https://pure.tue.nl/ws/files/4264441/466003-1.pdf> - part i
<https://pure.tue.nl/ws/files/4264443/466003-2.pdf> - part ii

Serie CMC : cooperative multimodel communication : international conference ; 1

Link to this page <http://repository.tue.nl/83e18b16-6733-4d04-8913-71355ed77ca7>

DECLARATION OF MICHAEL MCTEAR CONCERNING THE INTERNATIONAL CONFERENCE ON COOPERATIVE MULTIMODAL COMMUNICATION (CMC /95) IN EINDHOVEN, MAY 24-26, 1995

I, Dr. Michael McTear, declare as follows:

1. I am over the age of 18, have never been convicted of a felony or crime of moral turpitude and am legally competent to make this declaration. I have personal knowledge of the matters stated herein.
2. I am an Emeritus Professor at Ulster University in Jordanstown, United Kingdom.
3. I have been employed by Ulster University for 25 years.
4. In my position, I conduct research and teach in the area of spoken language technologies.
5. I attended the First International Conference on Cooperative Multimodal Communication in Eindhoven, The Netherlands in May of 1995 ("CMC /95").
6. CMC /95 was attended by about 30-50 people. All of the attendees of CMC /95 were active and known in the area of multimodal communications.
7. People working in the area of multimodal communications in 1995 would have been aware of the CMC /95 conference because the number of researchers and developers working in the field of multimodal communication and spoken language technologies at that time was small. The papers submitted for the conference were requested by researchers and developers working in the field, and authored by researchers and developers working in the field.

8. I am familiar with the paper submitted by Adam Cheyer and Luc Julia of SRI International to CMC /95, and presented at CMC /95 entitled "Multimodal Maps: An Agent-Based Approach." This paper was available to researchers in the field of natural language processing and multimodal communication no later than May of 1995.
9. In 1995, I was generally familiar with the work of SRI International, especially in the areas of natural language processing and multimodal communication.
10. In 1995, SRI International was generally known by those of skill in the art to have been involved in natural language processing and multimodal communication. It would have been common for one in the field of natural language processing and/or multimodal communication to review and reference SRI International publications, technical documents, and conference presentations as a source of information.

All statements made herein of my own knowledge are true and all statements made on information and belief are believed to be true. I further understand that willful false statements and the like are punishable by fine or imprisonment, or both under Section 1001 of Title 18 of the United States Code. I declare under penalty of perjury of the laws of the United States that the foregoing is true and correct.

Executed on 5 December 2017 in BELFAST, UK
[DATE] [CITY/STATE]

Michael F. McTear

[SIGN NAME HERE]

MICHAEL F. MCTEAR

[PRINT NAME HERE]

DECLARATION OF GERT-JAN VAN VELZEN CONCERNING THE "PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON COOPERATIVE MULTIMODAL COMMUNICATION: CMC /95, EINDHOVEN, MAY 24-26, 1995" REFERENCE

I, Gert-Jan van Velzen declare as follows:

1. I am over the age of 18, have never been convicted of a felony or crime of moral turpitude and am legally competent to make this declaration. I have personal knowledge of the matters stated herein.
2. I am an Account Manager for the Collections Department of the National Library of the Netherlands in The Hague, Netherlands (the "Library").
3. I have been employed by the Library for 29 years.
4. I am familiar with the regularly conducted record-keeping, cataloging and indexing activities and practices of the Library as they pertain to all references, books, magazines and other publications place in circulation within the Library.
5. A true and accurate copy of the Library's copy of Proceedings of the International Conference on Cooperative Multimodal Communication: CMC /95, Eindhoven, May 24-26, 1995 (hereafter "1995 CMC Proceedings"), including the article. "Multimodal Maps: An Agent-based Approach" (hereinafter "Multimodal Maps Article") by A. Cheyer and L. Julia on pages 103-113, is attached as Exhibit A. I have reviewed the portions of this reference in Exhibit A and they reflect a true and accurate copy of the corresponding portions of the reference in the Library.

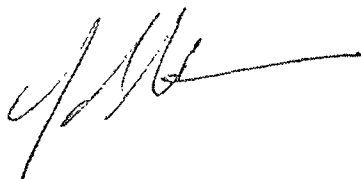
6. The Library maintains a catalog of all references that is available for members of the public to search. The catalog can be searched by subject, title, author, and keywords. An entry for the reference discussed in paragraph 5 was maintained in this catalog. An accurate copy of the catalog entry for the 1995 CMC Proceedings is attached as Exhibit B. The catalog can be searched by author, title, or keywords. In particular, the 1995 CMC Proceedings was cataloged under the keywords “multimedia,” “communicatie,” and “computertoepassingen,” as shown on the “Subject heading Depot” line of the catalog entry in Exhibit B. Based on my knowledge of practices of the Library and my review of the Library’s business records, the 1995 CMC Proceedings were cataloged, searchable, and accessible to the interested public from the Library at least by September 13, 1996.
7. The Library’s records that are regularly maintained in the course of its operation reflect that the 1995 CMC Proceedings, which includes the Multimodal Maps Article is in the Library’s collection and is available to members of the interested public.
8. The Library’s acquisition records for the 1995 CMC Proceedings is attached as Exhibit C. Based on my experience at the Library and knowledge of the Library’s practices, the Library’s acquisition records are made and kept in the ordinary course of the Library’s operation. Entries in the acquisition records are recorded at or near the time of the event that is being recorded. The entry “a (Te bestellen)” in Exhibit C indicates that the Library decided to order the 1995 CMC Proceedings on July 18, 1996. The entry “e (Besteld)” indicates that the Library ordered the 1995 CMC Proceedings on July 24,

1996. The entry “b (Bestelling ontvangen)” indicates that the 1995 CMC Proceeding was received by the Library for cataloging on August 8, 1996.

9. Exhibit D is a copy of the Library’s records that indicate when the 1995 CMC Proceedings was cataloged. Based on my experience at the Library and knowledge of the Library’s practices, the Library’s cataloging records are made and kept in the ordinary course of the Library’s operation. Entries in the cataloging records are recorded at or near the time of the event that is being recorded. Exhibit D includes an entry “4900 13-09-96 13:53:50.671” and “7001 13-09-96 : gdfg” that shows that the 1995 CMC Proceedings were cataloged on September 13, 1996. From September 13, 1996 until present the 1995 CMC Proceedings were listed in the Library’s online catalog available to members of the public. As shown in Exhibit D the labels “4900 13-09-96 13:53:50.671” and “7001 13-09-96 : gdfg”, these represent the time stamp – 4900 – and give a date for selection in the national bibliography of The Netherlands – 7001. Both of these dates are September 13, 1996.

All statements made herein of my own knowledge are true and all statements made on information and belief are believed to be true. I further understand that willful false statements and the like are punishable by fine or imprisonment, or both under Section 1001 of Title 18 of the United States Code. I declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct.

Executed on 14-12-2017 at The Hague, Netherlands.

A handwritten signature in black ink, appearing to read 'Gert-Jan van Velzen', is written over a horizontal line. The signature is stylized and cursive.

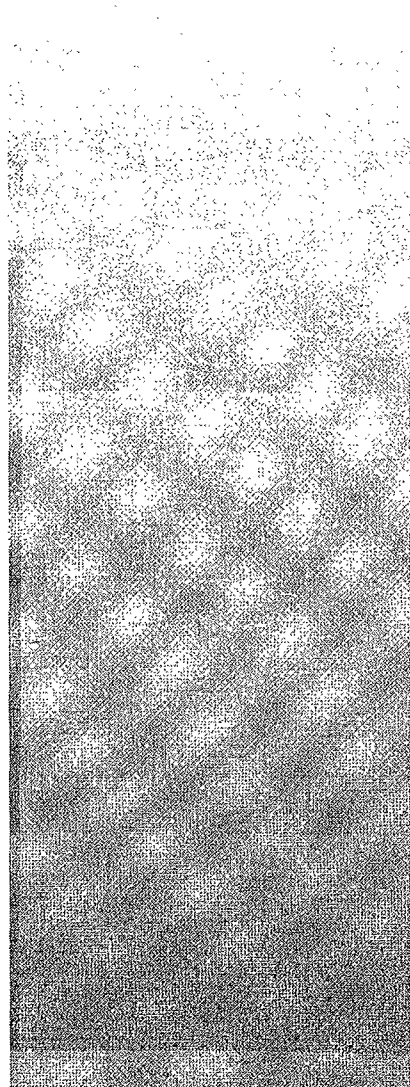
Gert-Jan van Velzen

EXHIBIT A

Samenwerkingsorgaan Brabantse Universiteiten

**Proceedings of the
International Conference on
Cooperative Multimodal
Communication CMC/95**

Part I



devices such as a mouse facilitate selection of an object or action, and drag and drop techniques allow items to be moved or combined with other entities or actions.

With the addition of electronic pen devices, gestural drawings add a new dimension direct manipulation interfaces. Gestures allow users to communicate a surprisingly wide range of meaningful requests with a few simple strokes. Research has shown that multiple gestures can be combined to form dialog, with rules of temporal grouping overriding temporal sequencing [22]. Gestural commands are particularly applicable to graphical or editing type tasks.

Direct manipulation interactions possess many desirable qualities: communication is generally fast and concise; input techniques are easy to learn and remember; the user has a good idea about what can be accomplished, as the visual presentation of the available actions is generally easily accessible. However, direct manipulation suffers from limitations when trying to access or describe entities which are not or can not be visualized by the user.

Limitations of direct manipulation style interfaces can be addressed by another interface technology, that of natural language interfaces. Natural language interfaces excel in describing entities that are not currently displayed on the monitor, in specifying temporal relations between entities or actions, and in identifying members of sets. These strengths are exactly the weaknesses of direct manipulation interfaces, and concurrently, the weaknesses of natural language interfaces (ambiguity, conceptual coverage, etc.) can be overcome by the strengths of direct manipulation.

Natural language content can be entered through different input modalities, including typing, handwriting, and speech. It is important to note that, while the same textual content can be provided by the three modalities, each modality has widely varying properties.

- Spoken language is the modality used first and foremost in human-human interactive problem solving [4]. Speech is an extremely fast medium, several times faster than typing or handwriting. In addition, speech input contains content that is not present in other forms of natural language input, such as prosody, tone and characteristics of the speaker (age, sex, accent).
- Typing is the most common way of entering information into a computer, because it is reasonably fast, very accurate, and requires no computational resources.
- Handwriting has been shown to be useful for certain types of tasks, such as performing numerical calculations and manipulating names which are difficult to pronounce [18, 19]. Because of its relatively slow production rate, handwriting may induce users to produce different types of input than is generated by spoken language; abbreviations, symbols and non-grammatical patterns may be expected to be more prevalent amid written input.

2.2 Combination of Modalities

As noted in the previous section, direct manipulation and natural language seem to be very complementary modalities. It is therefore not surprising that a number of multimodal systems combine the two.

Notable among such systems is the Cohen's Shoptalk system [6], a prototype manufacturing and decision-support system that aids in tasks such as quality assurance monitoring, and production scheduling. The natural language module of Shoptalk is based on the Chat-85

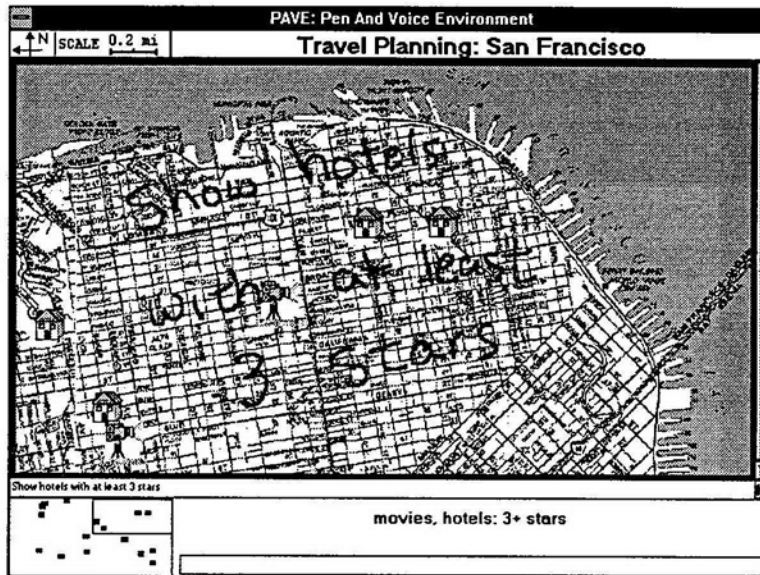


Figure 1: Multimodal Application for Travel Planning

natural language system [25] and is particularly good at handling time, tense, and temporal reasoning.

A number of systems have focused on combining the speed of speech with the reference provided by direct manipulation of a mouse pointer. Such systems include the XTRA system [1], CUBRICON [15], the PAC-Amodeus model [16], and TAPAGE [9].

XTRA and CUBRICON are both systems that combine complex spoken input with mouse clicks, using several knowledge sources for reference identification. CUBRICON's domain is a map-based task, making it similar to the application developed in this paper. However, the two are different in that CUBRICON can only use direct manipulation to indicate a specific item, whereas our system produces a richer mixing of modalities by adding both gestural and written language as input modalities.

The PAC-Amodeus systems such as VoicePaint and Notebook allow the user to synergistically combine vocal or mouse-click commands when interacting with notes or graphical objects. However, due to the selected domains, the natural language input is very simple, generally of the style "Insert a note here."

TAPAGE is another system that allows true synergistic combination of spoken input with direct manipulation. Like PAC-Amodeus, TAPAGE's domain provides only simple linguistic input. However, TAPAGE uses a pen-based interface instead of a mouse, allowing gestural commands. TAPAGE, selected as a building block for our map application, will be described more in detail in section 4.2.

Other interesting work regarding the simultaneous combination of handgestures and gaze can be found in [2, 13].

3 A Multimodal Map Application

In this section, we will describe a prototype map-based application for a travel planning domain. In order to provide the most natural user interface possible, the system permits the

user to simultaneously combine direct manipulation, gestural drawings, handwritten, typed and spoken natural language. When designing the system, other criteria were considered as well:

- The user interface must be light and fast enough to run on a handheld PDA while able to access applications and data that may require a more powerful machine.
- Existing commercial or research natural language and speech recognition systems should be used.
- Through the multimodal interface, a user must be able to transparently access a wide variety of data sources, including information stored in HTML form on the World Wide Web.

As illustrated in Figure 1, the user is presented with a pen sensitive map display on which drawn gestures and written natural language statements may be combined with spoken input. As opposed to a static paper map, the location, resolution, and content presented by the map change, according to the requests of the user. Objects of interest, such as restaurants, movie theaters, hotels, tourist sites, municipal buildings, etc. are displayed as icons. The user may ask the map to perform various actions. For example :

- *distance calculation* : e.g. "How far is the hotel from Fisherman's Wharf?"
- *object location* : e.g. "Where is the nearest post office?"
- *filtering* : e.g. "Display the French restaurants within 1 mile of this hotel."
- *information retrieval* : e.g. "Show me all available information about Alcatraz."

The application also makes use of multimodal (multimedia) output as well as input: video, text, sound and voice can all be combined when presenting an answer to a query.

During input, requests can be entered using gestures (see Figure 2 for sample gestures), handwriting, voice, or a combination of pen and voice. For instance, in order to calculate the distance between two points on the map, a command may be issued using the following:

- *gesture*, by simply drawing a line between the two points of interest.
- *voice*, by speaking "What is the distance from the post office to the hotel?"
- *handwriting*, by writing "dist p.o. to hotel?"
- *synergistic combination of pen and voice*, by speaking "What is the distance from here to this hotel?" while simultaneously indicating the specified locations by pointing or circling.

Notice that in our example of synergistic combination of pen and voice, the arguments to the verb "distance" can be specified before, at the same time, or shortly after the vocalization of the request to calculate the distance. If a user's request is ambiguous or underspecified, the system will wait several seconds and then issue a prompt requesting additional information.

The user interface runs on pen-equipped PC's or a Dauphin handheld PDA ([7]) using either a microphone or a telephone for voice input. The interface is connected either by

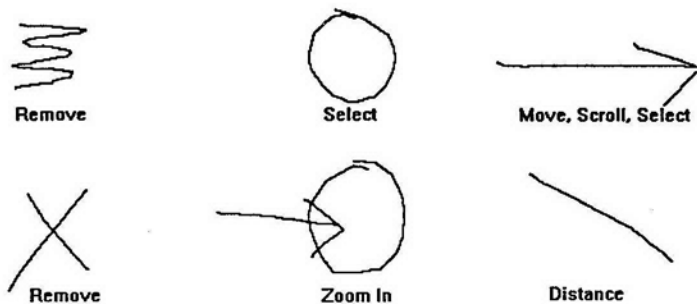


Figure 2: Sample gestures

modem or ethernet to a server machine which will manage database access, natural language processing and speech recognition for the application. The result is a mobile system that provides a synergistic pen/voice interface to remote databases.

In general, the speed of the system is quite acceptable. For gestural commands, which are handled locally on the user interface machine, a response is produced in less than one second. For handwritten commands, the time to recognize the handwriting, process the English query, access a database and begin to display the results on the user interface is less than three seconds (assuming an ethernet connection, and good network and database response). Solutions to verbal commands are displayed in three to five seconds after the end of speech has been detected; partial feedback indicating the current status of the speech recognition is provided earlier.

4 Approach

In order to implement the application described in the previous section, we chose to augment a proven agent-based architecture with functionalities developed for a synergistically multimodal application. The result is a flexible methodology for designing and implementing distributed multimodal applications.

4.1 Building Blocks

4.1.1 Open Agent Architecture

The Open Agent Architecture (OAA) [5] provides a framework for coordinating a society of agents which interact to solve problems for the user. Through the use of agents, the OAA provides distributed access to commercial applications, such as mail systems, calendar programs, databases, etc.

The Open Agent Architecture possesses several properties which make it a good candidate for our needs:

- An Interagent Communication Language (ICL) and Query Protocol have been developed, allowing agents to communicate among themselves. Agents can run on different platforms and be implemented in a variety of programming languages.
- Several natural language systems have been integrated into the OAA which convert English into the Interagent Communication Language. In addition, a speech recognition

agent has been developed to provide transparent access to the Corona speech recognition system.

- The agent architecture has been used to provide natural language and agent access to various heterogeneous data and knowledge sources.
- Agent interaction is very fine-grained. The architecture was designed so that a number of agents can work together, when appropriate in parallel, to produce fast responses to queries.

The architecture for the OAA, based loosely on Schwartz's FLiPSiDE system[23], uses a hierarchical configuration where client agents connect to a "facilitator" server. Facilitators provide content-based message routing, global data management, and process coordination for their set of connected agents. Facilitators can, in turn, be connected as clients of other facilitators. Each facilitator records the published functionality of their sub-agents, and when queries arrive in Interagent Communication Language form, they are responsible for breaking apart any complex queries and for distributing goals to the appropriate agents. An agent solving a goal may require supporting information and the agent architecture provides numerous means of requesting data from other agents or from the user.

Among the assortment of agent architectures, the Open Agent Architecture can be most closely compared to work by the ARPA knowledge sharing community [10]. The OAA's query protocol, Interagent Communication Language and Facilitator mechanisms have similar instantiations in the SHADE project, in the form of KQML, KIF and various independent capability matchmakers. Other agent architectures, such as General Magic's Telescript [11], MASCOs [20], or the CORBA distributed object approach [17] do not provide as fully developed mechanisms for interagent communication and delegation.

The Open Agent Architecture provides capability for accessing distributed knowledge sources through natural language and voice, but it is lacking integration with a synergistic multimodal interface.

4.1.2 TAPAGE

TAPAGE (edition de Tableaux par la Parole et la Geste) is a synergistic pen/voice system for designing and correcting tables.

To capture signals emitted during a user's interaction, TAPAGE integrates a set of modality agents, each responsible for a very specialized kind of signal [9]. The modality agents are connected to an "interpret agent" which is responsible for combining the inputs across all modalities to form a valid command for the application. The interpret agent receives filtered results from the modality agents, sorts the information into the correct fields, performs type-checking on the arguments, and prompts the user for any missing information, according to the model of the interaction. The interpret agent is also responsible for merging the data streams sent by the modality agents, and for resolving ambiguities among them, based on its knowledge of the application's internal state. Another function of the interpret agent is to produce reflexes: reflexes are actions output at the interface level without involving the functional core of the application.

The TAPAGE system can accept multimodal input, but it is not a distributed system; its functional core is fixed. In TAPAGE, the set of linguistic input is limited to a *verb object argument* format.

4.2 Synthesis

In the Open Agent Architecture, agents are distributed entities that can run on different machines, and communicate together to solve a task for the user. In TAPAGE, agents are used to provide streams of input to a central interpret process, responsible for merging incoming data. A generalization of these two types of agents could be :

Macro Agents: contain some knowledge and ability to reason about a domain, and can answer or make queries to other macro agents using the Interagent Communication Language.

Micro Agents: are responsible for handling a single input or output data stream, either filtering the signal to or from a hierarchically superior “interpret” agent.

The network architecture that we used was hierarchical at two resolutions – micro agents are connected to a superior macro agent, and macro agents are connected in turn to a facilitator agent. In both cases, a server is responsible for the supervision of its client sub-agents.

In order to describe our implementation, we will first give a description of each agent used in our application and then illustrate the flow of communication among agents produced by a user’s request.

Speech Recognition (SR) Agent: The SR agent provides a mapping from the Interagent Communication Language to the API for the Decipher (Corona) speech recognition system [4], a continuous speech speaker independent recognizer based on Hidden Markov Model technology. This macro agent is also responsible for supervising a child micro agent whose task is to control the speech data stream. The SR agent can provide feedback to an interface agent about the current status and progress of the micro agent (e.g. “listening”, “end of speech detected”, etc.) This agent is written in C.

Natural Language (NL) Parser Agent: translates English expressions into the Interagent Communication Language (ICL). For a more complete description of the ICL, see [5]. The NL agent we selected for our application is the simplest of those integrated into the OAA. It is written in Prolog using Definite Clause Grammars, and supports a distributed vocabulary; each agent dynamically adds word definitions as it connects to the network. A current project is underway to integrate the Gemini natural language system [4], a robust bottom up parser and semantic interpreter specifically designed for use in Spoken Language Understanding projects.

Database Agents: Database agents can reside at local or remote locations and can be grouped hierarchically according to content. Micro agents can be connected to database agents to monitor relevant positions or events in real time. In our travel planning application, database agents provide maps for each city, as well as icons, vocabulary and information about available hotels, restaurants, movies, theaters, municipal buildings and tourist attractions. Three types of databases were used: Prolog databases, X.500 hierarchical databases, and data loaded automatically by scanning HTML pages from the World Wide Web (WWW). In one instance, a local newspaper provides weekly updates to its Mosaic-accessible list of current movie times and reviews, as well as adding several new restaurant reviews to a growing collection; this information is extracted by an HTML reading database agent and made accessible to the agent architecture. Descriptions and addresses of new restaurants are presented to the user on request, and the user can choose to add them to the permanent database by specifying positional coordinates on the map (eg. “add this new restaurant here”), information lacking in the WWW database.

Reference Resolution Agent: This agent is responsible for merging requests arriving in parallel from different modalities, and for controlling interactions between the user interface

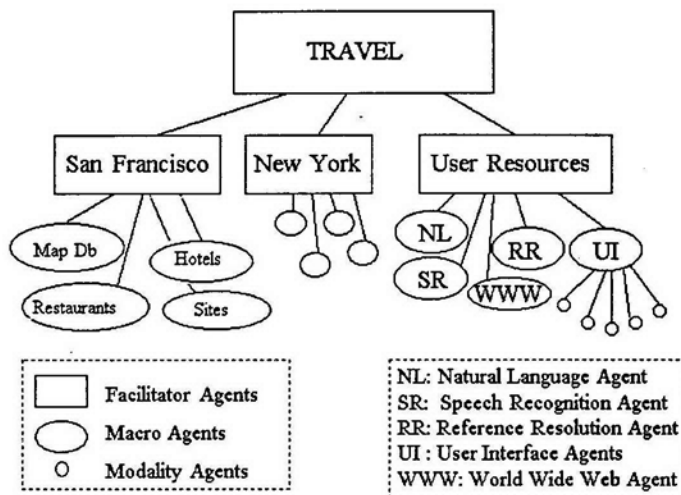


Figure 3: Agent Architecture for Map Application

agent, database agents and modality agents. In this implementation, the reference resolution agent is domain specific: knowledge is encoded as to what actions must be performed to resolve each possible type of ICL request in its particular domain. For a given ICL logical form, the agent can verify argument types, supply default values, and resolve argument references. Some argument references are descriptive (“How far is it to the hotel on Emerson Street?”); in this case, a domain agent will try to resolve the definite reference by sending database agent requests. Other references, particularly when contextual or deictic, are resolved by the user interface agent (“What are the rates for this hotel?”). Once arguments to a query have been resolved, this agent coordinates the actions and calculations necessary to produce the result of the request.

Interface Agent: This macro agent is responsible for managing what is currently being displayed to the user, and for accepting the user’s multimodal input. The Interface Agent also coordinates client modality agents and resolves ambiguities among them: handwriting and gestures are interpreted locally by micro agents and combined with results from the speech recognition agent, running on a remote speech server. The handwriting micro-agent interfaces with the Microsoft PenWindows API and accesses a handwriting recognizer by CIC Corporation. The gesture micro-agent accesses recognition algorithms developed for TAPAGE.

An important task for the interface agent is to record which objects of each type are currently salient, in order to resolve contextual references such as “the hotel” or “where I was before.” Deictic references are resolved by gestural or direct manipulation commands. If no such indication is currently specified, the user interface agent waits long enough to give the user an opportunity to supply the value, and then prompts the user for it.

We shall now give an example of the distributed interaction of agents for a specific query. In the following example, all communication among agents passes transparently through a facilitator agent in an undirected fashion; this process is left out of the description for brevity.

1. A user speaks: “How far is the restaurant from this hotel?”

2. The speech recognition agent monitors the status and results from its micro agent, sending feedback received by the user interface agent. When the string is recognized, a translation is requested.
3. The English request is received by the NL agent and translated into ICL form.
4. The reference resolution agent (RR) receives the ICL distance request containing one definite and one deictic reference and asks for resolution of these references.
5. The interface agent uses contextual structures to find what "the restaurant" refers to, and waits for the user to make a gesture indicating "the hotel", issuing prompts if necessary.
6. When the references have been resolved, the domain agent (RR) sends database requests asking for the coordinates of the items in question. It then calculates the distance according to the scale of the currently displayed map, and requests the user interface to produce output displaying the result of the calculation.

5 Conclusions

By augmenting an existing agent-based architecture with concepts necessary for synergistic multimodal input, we were able to rapidly develop a map-based application for a travel planning task. The resulting application has met our initial requirements: a mobile, synergistic pen/voice interface providing good natural language access to heterogeneous distributed knowledge sources. The approach used was general and should provide a for developing synergistic multimodal applications for other domains.

The system described here is one of the first that accepts commands made of synergistic combinations of spoken language, handwriting and gestural input. This fusion of modalities can produce more complex interactions than in many systems and the prototype application will serve as a testbed for acquiring a better understanding of multimodal input.

In the near future, we will continue to verify and extend our approach by building other multimodal applications. We are interested in generalizing the methodology even further; work has already begun on an agent-building tool which will simplify and automate many of the details of developing new agents and domains.

References

- [1] Allegayer, J, Jansen-Winkel, R., Reddig, C. and Reithinger, N. "Bidirectional use of knowledge in the multi-modal NL access system XTRA". In Proceedings of IJCAI-89, Detroit, pp. 1492-1497.
- [2] Bolt, R. "Put that there: Voice and Gesture at the Graphic Interface". Computer Graphics, 14(3), 1980, pp. 262-270.
- [3] Bellik, Y. and Teil, D. "Les types de multimodalites", In Proc. IIM'92 (Paris), pp. 22-28.
- [4] Cohen, M., Murveit, H., Bernstein, J., Price, P., Weintraub, M., "The DECIPHER Speech Recognition System". 1990 IEEE ICASSP, pp. 77-80.

- [5] Cohen, P.R., Cheyer, A., Wang, M. and Baeg, S.C. "An Open Agent Architecture". In Proc. AAI'94 - SA (Stanford), pp. 1-8.
- [6] Cohen, P. "The role of natural language in a multimodal interface". Proceedings of UIST'92, 143-149.
- [7] Dauphin DTR-1 User's Manual, Dauphin Technology, Inc. 337 E. Butterfield Rd., Suite 900, Lombard, Ill 60148.
- [8] Dowding, J., Gawron, J.M., Appelt, D., Bear, J., Cherny, L., Moore, B. and Moran D., "Gemini: A natural language system for spoken-language understanding", Technical Note 527, AI Center, SRI International, 333 Ravenswood Ave., Menlo Park, CA 94025, April 1993.
- [9] Faure, C. and Julia, L. "An Agent-Based Architecture for a Multimodal Interface". In Proc. AAI'94 - IM4S (Stanford), pp. 82-86.
- [10] Genesereth, M. and Singh, N.P. "A knowledge sharing approach to software interoperation". Computer Science Department, Stanford University, unpublished ms., 1994.
- [11] General Magic, Inc., "Telescript Product Documentation", 1995.
- [12] Julia, L. and Faure, C. "A Multimodal Interface for Incremental Graphic Document Design". HCI International '93, Orlando.
- [13] Koons, D.B., Sparrell, C.J., and Thorisson, K.R. "Integrating Simultaneous Input from Speech, Gaze and Hand Gestures". In *Intelligent Multimedia Interfaces*, Edited by Mark Maybury, Menlo Park, CA, AAI Press, 1993.
- [14] Maybury, M.T. (ed.), *Intelligent Multimedia Interfaces*, AAI Press/MIT Press: Menlo Park, Ca, 1993.
- [15] Neal, J.G., and Shapiro, S.C. "Intelligent Multi-media Interface Technology". In *Intelligent User Interfaces*, Edited by J. Sullivan and S. Tyler, Addison-Wesley Pub. Co., Reading, MA, 1991.
- [16] Nigay, L. and Coutaz, J. "A Design Space for Multimodal Systems: Concurrent Processing and Data Fusion". In Proc. InterCHI'93 (Amsterdam), ACM Press, pp. 172-178.
- [17] Object Management Group, "The Common Object Request Broker: Architecture and Specification", OMG Document Number 91.12.1, December 1991.
- [18] Oviatt, S. "Toward Empirically-Based Design of Multimodal Dialogue Systems". In Proc. AAI'94 - IM4S (Stanford), pp. 30-36.
- [19] Oviatt, S. and Olsen, E. "Integration Themes in Multimodal Human-Computer Interaction". Proceedings of ICSLP'94, Yokohama, pp. 551-554.
- [20] Park, S.K., Choi J.M., Myeong-Wuk J., Lee G.L., and Lim Y.H. "MASCOS : A Multi-Agent System as the Computer Secretary". Submitted for publication.
- [21] Pfaff, G. and Ten Hagen, P.J.W. *Seeheim workshop on User Interface Management Systems* (Berlin), Springer- Verlag.

- [22] Rhyne J. "Dialogue Management for Gestural Interfaces". *Computer Graphics*, 21(2), 1987, pp. 137-142.
- [23] Schwartz, D.G. "Cooperating heterogeneous systems: A blackboard-based meta approach". Technical Report 93-112, Center for Automation and Intelligent Systems Research, Case Western Reserve University, Cleveland Ohio, April 1993. Unpublished PhD. thesis.
- [24] Sullivan, J. and Tyler, S. (eds.), *Intelligent User Interfaces*, Addison-Wesley Pub. Co., Reading, MA, 1991.
- [25] Warren, D. and Pereira, F., "An Efficient Easily Adaptable System for Interpreting Natural Language Queries", in *American Journal of Computational Linguistics*, 8(3), 1982, pp. 110-123.
- [26] Wauchope, K., "Eucalyptus: Integrating Natural Language with a Graphical User Interface." Naval Research Laboratory Technical Report NRL/FR/5510-94-9711, in press, 1994.

EXHIBIT B



Title: [Proceedings of the International Conference on Cooperative Multimodel Communication CMC/95 : Eindhoven, May 24-26, 1995 / Harry Bunt, Robbert-Jan Beun & Tijn Borghuis \(eds.\)](#)

Collaborator: [Hendrik Cornelis Bunt \(1944-\); Robbert-Jan Beun](#)

Congress: [International Conference on Cooperative Multimodel Communication CMC/95 \(1995 ; Eindhoven\)](#)

Year: [1995]

Publisher: [\[Tilburg : Katholieke Universiteit Brabant\]](#)
[\[Eindhoven : Technische Universiteit Eindhoven\]](#)

Note: Met lit. opg., reg

Extent: 2 dl. (VII, 324 p)

Illustration: ill

Size: 30 cm

ISBN: 90-9008315-4

Subject heading Depot: [multimedia](#); [communicatie](#); [computertoepassingen](#)

Request number: 5085886 [-5085887]

Loan indication: for inspection only

Lending information: This item is available.

Availability: [Goto list of volumes](#) 2 volumes

Information Availability

| | |
|------|-------------------------|
| DI.2 | Request |
| DI.1 | Request |



EXHIBIT C

Aax | Proceedings of the International Conference on Cooperative Multimodal Communication CMO'95 : Eindhoven, May 24-26, 1995 / Harry Bunt, [1995]

ISBN 90-9008315-4 | Tilburg : Katholieke Universiteit Brabant | 5085386 [5085887]

Best. **Peak3**

| | | | |
|--|--|--------|---|
| Bestelnnummer <input type="checkbox"/> | 86033831 | PPN | 144863272 |
| Opdr bestelnnummer | <input type="text"/> | EPN | 223040797 |
| Plaatsing | <input type="text"/> | | |
| Leverancier <input type="checkbox"/> | 8008315 | | samenwerkingsverband brabantse universiteiten |
| Verzendwijze | <input checked="" type="radio"/> Print <input type="radio"/> Email | | |
| Type | <input type="checkbox"/> h Boekbestelling DEPOT gratis | | |
| Stadium | <input type="checkbox"/> y Belaabaarstellen | | |
| Selectiecode | <input type="text"/> | | |
| Instituutcode | 4 | | Eoeken depot Collectie 1 |
| Bestemming | b | | Eoeken Depot-collectie |
| Leveringsnummer | 96049200 | Budget | z00b66 |

Onm. **Gescht.** **Lev.**

| | |
|--------------------------|--------------------------|
| Beginslatus | Erdslatus |
| b (Bestelling ontvangen) | y (Belaatbaarstellen) |
| e (Besteld) | b (Bestelling ontvangen) |
| a (Te bestellen) | e (Besteld) |
| | a (Te bestellen) |

| | |
|-------------------------------|---------------------------------|
| Invoerdatum | 18-07-1996 11:18 |
| Multilectatum | 05-08-1996 14:49 |
| Besteld op | 24-07-1996 |
| Verz.factuur | <input type="checkbox"/> Bedrag |
| Toegevoegd nr. | <input type="text"/> |
| oorspronkelijk besteld aantal | 1 |
| Referentienr. | <input type="text"/> |
| Activeren op | 18-07-1996 |
| Bedrag op bon? | <input type="checkbox"/> |
| Aanvrager | hvp |

EXHIBIT D

Ingevoerd: 1001:23-01-96 Gewijzigd: 1999:22-08-13 06:39:29 Status: 9999:99-99-99

0500 Aax
0501 #tekst=txt %%rdacontent/dut
0502 #zonder medium=n %%rdamedia/dut
0503 #band=nc %%rdacarrier/dut
1100 1995 \$ [1995]
1121 u
1500 /1eng
1700 /1nl
2000 9090083154
2020 B9635959
2097 #OCoLC#69071749
3011 Harry@Bunt!068920075!Hendrik Cornelis Bunt (1944-) (ISNI 0000 0001 2149 0086)
3012 Robbert-Jan@Beun!075105888!Robbert-Jan Beun (ISNI 0000 0000 8317 9093)
3161 @International Conference on Cooperative Multimodel Communication CMC/95
(Eindhoven) : 1995
4000 @Proceedings of the International Conference on Cooperative Multimodel
Communication CMC/95 : Eindhoven, May 24-26, 1995 / Harry Bunt, Robbert-Jan Beun &
Tijn Borghuis (eds.)
4030 [Tilburg : Katholieke Universiteit Brabant]
4031 [Eindhoven : Technische Universiteit Eindhoven]
4060 2 dl. (VII, 324 p)
4061 ill
4062 30 cm
4204 Met lit. opg., reg
5201 !12160800X!multimedia
5202 !075635143!communicatie
5203 !075603195!computertoepassingen
3521 !075385899!@Katholieke Universiteit Brabant, Tilburg
3522 !075382903!@Technische Universiteit Eindhoven
4701 ea
4900 13-09-96 13:53:50.671
7001 13-09-96 : gdfg
7100 5085886 [-5085887] !d! @ f
8008 rp/29
8009 rp/32
7900 18-09-96 14:29:57.169
7800 223040797



www.archive.org
415.561.6767
415.840-0391 e-fax

Internet Archive
300 Funston Avenue
San Francisco, CA 94118

AFFIDAVIT OF CHRISTOPHER BUTLER

1. I am the Office Manager at the Internet Archive, located in San Francisco, California. I make this declaration of my own personal knowledge.

2. The Internet Archive is a website that provides access to a digital library of Internet sites and other cultural artifacts in digital form. Like a paper library, we provide free access to researchers, historians, scholars, and the general public. The Internet Archive has partnered with and receives support from various institutions, including the Library of Congress.

3. The Internet Archive has created a service known as the Wayback Machine. The Wayback Machine makes it possible to surf more than 450 billion pages stored in the Internet Archive's web archive. Visitors to the Wayback Machine can search archives by URL (i.e., a website address). If archived records for a URL are available, the visitor will be presented with a list of available dates. The visitor may select one of those dates, and then begin surfing on an archived version of the Web. The links on the archived files, when served by the Wayback Machine, point to other archived files (whether HTML pages or images). If a visitor clicks on a link on an archived page, the Wayback Machine will serve the archived file with the closest available date to the page upon which the link appeared and was clicked.

4. The archived data made viewable and browseable by the Wayback Machine is compiled using software programs known as crawlers, which surf the Web and automatically store copies of web files, preserving these files as they exist at the point of time of capture.

5. The Internet Archive assigns a URL on its site to the archived files in the format [http://web.archive.org/web/\[Year in yyyy\]\[Month in mm\]\[Day in dd\]\[Time code in hh:mm:ss\]/\[Archived URL\]](http://web.archive.org/web/[Year in yyyy][Month in mm][Day in dd][Time code in hh:mm:ss]/[Archived URL]). Thus, the Internet Archive URL <http://web.archive.org/web/19970126045828/http://www.archive.org/> would be the URL for the record of the Internet Archive home page HTML file (<http://www.archive.org/>) archived on January 26, 1997 at 4:58 a.m. and 28 seconds (1997/01/26 at 04:58:28). A web browser may be set such that a printout from it will display the URL of a web page in the printout's footer. The date assigned by the Internet Archive applies to the HTML file but not to image files linked therein. Thus images that appear on a page may not have been archived on the same date as the HTML file. Likewise, if a website is designed with "frames," the date assigned by the Internet Archive applies to the frameset as a whole, and not the individual pages within each frame.

6. Attached hereto as Exhibit A are true and accurate copies of printouts of the Internet Archive's records of the HTML files for the URLs and the dates specified in the footer of the printout.

7. I declare under penalty of perjury that the foregoing is true and correct.

DATE: 12/18/17



Christopher Butler

CALIFORNIA JURAT

See Attached Document.

A notary public or other officer completing this certificate verifies only the identity of the individual who signed the document to which this certificate is attached, and not the truthfulness, accuracy, or validity of that document.

State of California
County of San Francisco

Subscribed and sworn to (or affirmed) before me on this

18th day of December, 2017, by

Christopher Butler,

proved to me on the basis of satisfactory evidence to be the person who appeared before me.

Signature: 



Exhibit A

[Next](#) | [Up](#) | [Previous](#)

Next: [Introduction](#)

Multimodal Maps: An Agent-based Approach

*Adam CHEYER and Luc JULIA
SRI International
333 Ravenswood Ave
Menlo Park, CA 94025 - USA*

Abstract:

In this paper, we discuss how multiple input modalities may be combined to produce more natural user interfaces. To illustrate this technique, we present a prototype map-based application for a travel planning domain. The application is distinguished by a synergistic combination of handwriting, gesture and speech modalities; access to existing data sources including the World Wide Web; and a mobile handheld interface. To implement the described application, a distributed network of heterogeneous software agents was augmented by appropriate functionality for developing synergistic multimodal applications.

-
- [Introduction](#)
 - [Natural Input](#)
 - [Input Modalities](#)
 - [Combination of Modalities](#)
 - [A Multimodal Map Application](#)
 - [Approach](#)
 - [Building Blocks](#)
 - [Open Agent Architecture](#)
 - [TAPAGE](#)
 - [Synthesis](#)
 - [CONCLUSIONS](#)
 - [Acknowledgements](#)
 - [References](#)
 - [About this document ...](#)

*Adam Cheyer
Mon Aug 12 15:07:21 PDT 1996*

[Next](#) | [Up](#) | [Previous](#)

Next: [Natural Input](#) **Up:** [Multimodal Maps: An Agent-based](#) **Previous:** [Multimodal Maps: An Agent-based](#)

Introduction

As computer systems become more powerful and complex, efforts to make computer interfaces more simple and natural become increasingly important. Natural interfaces should be designed to facilitate communication in ways people are already accustomed to using. Such interfaces should allow users to concentrate on the tasks they are trying to accomplish, not worry about what they must do to control the interface.

In this paper, we begin by discussing what input modalities humans are comfortable using when interacting with computers, and how these modalities should best be combined in order to produce natural interfaces. In section three, we present a prototype map-based application for the travel planning domain which uses a synergistic combination of several input modalities. Section four describes the agent-based approach we used to implement the application and the work on which it is based. In section five, we summarize our conclusions and future directions.

Adam Cheyer

Mon Aug 12 15:07:21 PDT 1996

[Next](#) | [Up](#) | [Previous](#)

Next: [Input Modalities](#) Up: [Multimodal Maps: An Agent-based](#) Previous: [Introduction](#)

Natural Input

- [Input Modalities](#)
 - [Combination of Modalities](#)
-

Adam Cheyer

Mon Aug 12 15:07:21 PDT 1996

[Next](#) | [Up](#) | [Previous](#)

Next: [Combination of Modalities](#) **Up:** [Natural Input](#) **Previous:** [Natural Input](#)

Input Modalities

Direct manipulation interface technologies are currently the most widely used techniques for creating user interfaces. Through the use of menus and a graphical user interface, users are presented with sets of discrete actions and the objects on which to perform them. Pointing devices such as a mouse facilitate selection of an object or action, and drag and drop techniques allow items to be moved or combined with other entities or actions.

With the addition of electronic pen devices, gestural drawings add a new dimension to direct manipulation interfaces. Gestures allow users to communicate a surprisingly wide range of meaningful requests with a few simple strokes. Research has shown that multiple gestures can be combined to form dialog, with rules of temporal grouping overriding temporal sequencing [\[\[23\]\]](#). Gestural commands are particularly applicable to graphical or editing type tasks.

Direct manipulation interactions possess many desirable qualities: communication is generally fast and concise; input techniques are easy to learn and remember; the user has a good idea about what can be accomplished, as the visual presentation of the available actions is generally easily accessible. However, direct manipulation suffers from limitations when trying to access or describe entities which are not or can not be visualized by the user.

Limitations of direct manipulation style interfaces can be addressed by another interface technology, that of natural language interfaces. Natural language interfaces excel in describing entities that are not currently displayed on the monitor, in specifying temporal relations between entities or actions, and in identifying members of sets. These strengths are exactly the weaknesses of direct manipulation interfaces, and concurrently, the weaknesses of natural language interfaces (ambiguity, conceptual coverage, etc.) can be overcome by the strengths of direct manipulation [\[\[6\]\]](#).

Natural language content can be entered through different input modalities, including typing, handwriting, and speech. It is important to note that, while the same textual content can be provided by the three modalities, each modality has widely varying properties.

- Spoken language is the modality used first and foremost in human-human interactive problem solving [\[\[4\]\]](#). Speech is an extremely fast medium, several times faster than typing or handwriting. In addition, speech input contains content that is not present in other forms of natural language input, such as prosody, tone and characteristics of the speaker (age, sex, accent).
- Typing is the most common way of entering information into a computer, because it is reasonably fast, very accurate, and requires no computational resources.
- Handwriting has been shown to be useful for certain types of tasks, such as performing numerical calculations and manipulating names which are difficult to pronounce [\[\[18\]](#), [\[20\]\]](#). Because of its relatively slow production rate, handwriting may induce users to produce different types of input than is generated by spoken language; abbreviations, symbols and non-grammatical patterns may be expected to be more prevalent amid written input.

[Next](#) | [Up](#) | [Previous](#)

Next: [Combination of Modalities](#) **Up:** [Natural Input](#) **Previous:** [Natural Input](#)

Adam Cheyer

Mon Aug 12 15:07:21 PDT 1996

[Next](#) | [Up](#) | [Previous](#)

Next: [Approach](#) **Up:** [Multimodal Maps: An Agent-based](#) **Previous:** [Combination of Modalities](#)

A Multimodal Map Application

In this section, we will describe a prototype map-based application for a travel planning domain. In order to provide the most natural user interface possible, the system permits the user to simultaneously combine direct manipulation, gestural drawings, handwritten, typed and spoken natural language. When designing the architecture for the system, other criteria were considered as well:

- The user interface must be light and fast enough to run on a handheld PDA while able to access applications and data that may require a more powerful machine.
- Existing commercial or research natural language and speech recognition systems should be used.
- Through the multimodal interface, a user must be able to transparently access a wide variety of data sources, including information stored in HTML format on the World Wide Web.



Figure 1: Multimodal Application for Travel Planning

The map functionality, interface design, and classes of input data of the system presented here is based on a design by Oviatt and Cohen, used by them in a wizard-of-oz simulation system designed to explore complex interactions of modalities [19]. The agent-based architecture used to realize Oviatt and Cohen's design is new, as is its application to travel planning.

As illustrated in Figure 1, the user is presented with a pen sensitive map display on which drawn gestures and handwritten natural language statements may be combined with spoken input. As opposed to a static paper map, the location, resolution, and content presented by the map change, according to the requests of the user. Objects of interest, such as restaurants, movie theaters, hotels, tourist sites, municipal buildings, etc. are displayed as icons. The user may ask the map to perform various actions.

Figure 2: Sample gestures

[Next](#) | [Up](#) | [Previous](#)

Next: [Approach](#) **Up:** [Multimodal Maps: An Agent-based](#) **Previous:** [Combination of Modalities](#)

Adam Cheyer

Mon Aug 12 15:07:21 PDT 1996

[Next](#) | [Up](#) | [Previous](#)

Next: [Building Blocks](#) **Up:** [Multimodal Maps: An Agent-based](#) **Previous:** [A Multimodal Map Application](#)

Approach

In order to implement the application described in the previous section, we chose to augment a proven agent-based architecture with functionalities developed for a synergistically multimodal application. The result is a flexible methodology for designing and implementing distributed multimodal applications.

-
- [Building Blocks](#)
 - [Open Agent Architecture](#)
 - [TAPAGE](#)
 - [Synthesis](#)
-

Adam Cheyer

Mon Aug 12 15:07:21 PDT 1996

[Next](#) | [Up](#) | [Previous](#)

Next: [Open Agent Architecture](#) **Up:** [Approach](#) **Previous:** [Approach](#)

Building Blocks

- [Open Agent Architecture](#)
 - [TAPAGE](#)
-

Adam Cheyer

Mon Aug 12 15:07:21 PDT 1996

[Next](#) | [Up](#) | [Previous](#)

Next: [TAPAGE](#) Up: [Building Blocks](#) Previous: [Building Blocks](#)

Open Agent Architecture

The Open Agent Architecture (OAA) [\[15\]](#) provides a framework for coordinating a society of agents which interact to solve problems for the user. Through the use of agents, the OAA provides distributed access to commercial applications, such as mail systems, calendar programs, databases, etc.

The Open Agent Architecture possesses several properties which make it a good candidate for our needs:

- An Interagent Communication Language (ICL) and Query Protocol have been developed, allowing agents to communicate among themselves. Agents can run on different platforms and be implemented in a variety of programming languages.
- Several natural language systems have been integrated into the OAA which convert English into the Interagent Communication Language. In addition, a speech recognition agent has been developed to provide transparent access to the Corona speech recognition system.
- The agent architecture has been used to provide natural language and agent access to various heterogeneous data and knowledge sources.
- Agent interaction is very fine-grained. The architecture was designed so that a number of agents can work together, when appropriate in parallel, to produce fast responses to queries.

The architecture for the OAA, based loosely on Schwartz's FLiPSiDE system [\[24\]](#), uses a hierarchical configuration where client agents connect to a "facilitator" server. Facilitators provide content-based message routing, global data management, and process coordination for their set of connected agents. Facilitators can, in turn, be connected as clients of other facilitators. Each facilitator records the published functionality of their sub-agents, and when queries arrive in Interagent Communication Language form, they are responsible for breaking apart any complex queries and for distributing goals to the appropriate agents. An agent solving a goal may require supporting information and the agent architecture provides numerous means of requesting data from other agents or from the user.

Among the assortment of agent architectures, the Open Agent Architecture can be most closely compared to work by the ARPA knowledge sharing community [\[10\]](#). The OAA's query protocol, Interagent Communication Language and Facilitator mechanisms have similar instantiations in the SHADE project, in the form of KQML, KIF and various independent capability matchmakers. Other agent architectures, such as General Magic's Telescript [\[11\]](#), MASCOS [\[21\]](#), or the CORBA distributed object approach [\[17\]](#) do not provide as fully developed mechanisms for interagent communication and delegation.

The Open Agent Architecture provides capability for accessing distributed knowledge sources through natural language and voice, but it is lacking integration with a synergistic multimodal interface.

[Next](#) | [Up](#) | [Previous](#)

Next: [TAPAGE](#) Up: [Building Blocks](#) Previous: [Building Blocks](#)

Adam Cheyer

Mon Aug 12 15:07:21 PDT 1996

the user on request, and the user can choose to add them to the permanent database by specifying positional coordinates on the map (eg. ``add this new restaurant here''), information lacking in the WWW database.

Reference Resolution Agent: This agent is responsible for merging requests arriving in parallel from different modalities, and for controlling interactions between the user interface agent, database agents and modality agents. In this implementation, the reference resolution agent is domain specific: knowledge is encoded as to what actions must be performed to resolve each possible type of ICL request in its particular domain. For a given ICL logical form, the agent can verify argument types, supply default values, and resolve argument references. Some argument references are descriptive (``How far is it to the hotel on Emerson Street?"); in this case, a domain agent will try to resolve the definite reference by sending database agent requests. Other references, particularly when contextual or deictic, are resolved by the user interface agent (``What are the rates for this hotel?"). Once arguments to a query have been resolved, this agent coordinates the actions and calculations necessary to produce the result of the request.

Interface Agent: This macro agent is responsible for managing what is currently being displayed to the user, and for accepting the user's multimodal input. The Interface Agent also coordinates client modality agents and resolves ambiguities among them : handwriting and gestures are interpreted locally by micro agents and combined with results from the speech recognition agent, running on a remote speech server. The handwriting micro-agent interfaces with the Microsoft PenWindows API and accesses a handwriting recognizer by CIC Corporation. The gesture micro- agent accesses recognition algorithms developed for TAPAGE.

An important task for the interface agent is to record which objects of each type are currently salient, in order to resolve contextual references such as ``the hotel" or ``where I was before." Deictic references are resolved by gestural or direct manipulation commands. If no such indication is currently specified, the user interface agent waits long enough to give the user an opportunity to supply the value, and then prompts the user for it.

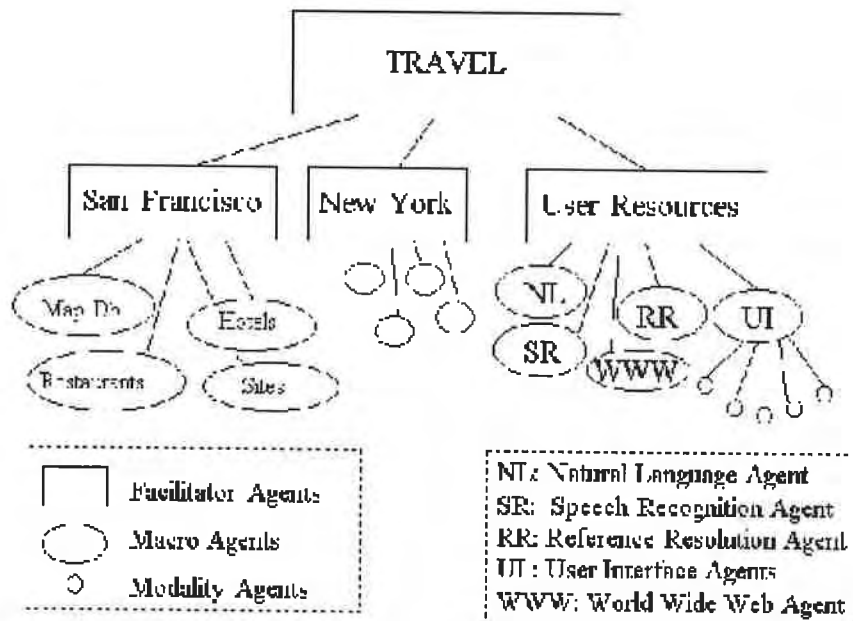


Figure 3: Agent Architecture for Map Application

We shall now give an example of the distributed interaction of agents for a specific query. In the following example, all communication among agents passes transparently through a facilitator agent in an undirected fashion; this process is left out of the description for brevity.

1. A user speaks: ``How far is the restaurant from this hotel?''
2. The speech recognition agent monitors the status and results from its micro agent, sending feedback received by the user interface agent. When the string is recognized, a translation is requested.
3. The English request is received by the NL agent and translated into ICL form.
4. The reference resolution agent (RR) receives the ICL distance request containing one definite and one deictic reference and asks for resolution of these references.
5. The interface agent uses contextual structures to find what ``the restaurant'' refers to, and waits for the user to make a gesture indicating ``the hotel'', issuing prompts if necessary.
6. When the references have been resolved, the domain agent (RR) sends database requests asking for the coordinates of the items in question. It then calculates the distance according to the scale of the currently displayed map, and requests the user interface to produce output displaying the result of the calculation.

[Next](#) | [Up](#) | [Previous](#)

Next: [CONCLUSIONS](#) **Up:** [Approach](#) **Previous:** [TAPAGE](#)

Adam Cheyer

Mon Aug 12 15:07:21 PDT 1996

[Next](#) | [Up](#) | [Previous](#)

Next: [Acknowledgements](#) **Up:** [Multimodal Maps: An Agent-based](#) **Previous:** [Synthesis](#)

CONCLUSIONS

By augmenting an existing agent-based architecture with concepts necessary for synergistic multimodal input, we were able to rapidly develop a map-based application for a travel planning task. The resulting application has met our initial requirements: a mobile, synergistic pen/voice interface providing good natural language access to heterogeneous distributed knowledge sources. The approach used was general and should provide a means for developing synergistic multimodal applications for other domains.

The system described here is one of the first that accepts commands made of synergistic combinations of spoken language, handwriting and gestural input. This fusion of modalities can produce more complex interactions than in many systems and the prototype application will serve as a testbed for acquiring a deeper understanding of multimodal input.

In the near future, we will continue to verify and extend our approach by building other multimodal applications. We are interested in generalizing the methodology further; work has already begun on an agent-building tool which will simplify and automate many of the details of developing new agents and domains.

Adam Cheyer

Mon Aug 12 15:07:21 PDT 1996

[Next](#) | [Up](#) | [Previous](#)

Next: [References](#) **Up:** [Multimodal Maps: An Agent-based](#) **Previous:** [CONCLUSIONS](#)

Acknowledgements

The work reported here would not have been possible without the inspiration of Sharon Oviatt and Phil Cohen under whose direction we worked for a year on a project (NSF Grant No. IRI-9213472) in which the combination of modalities contained in the interface presented here was crystallized and studied via simulations. Neither they nor their sponsors, of course, are responsible for the work presented here.

Adam Cheyer

Mon Aug 12 15:07:21 PDT 1996

[Next](#) | [Up](#) | [Previous](#)

Next: [About this document](#) **Up:** [Multimodal Maps: An Agent-based](#) **Previous:** [Acknowledgements](#)

References

- 1 Allegayer, J, Jansen-Winkeln, R., Reddig, C. and Reithinger, N. ``Bidirectional use of knowledge in the multi-modal NL access system XTRA". In Proceedings of IJCAI-89, Detroit, pp. 1492-1497.
- 2 Bolt, R. ``Put that there: Voice and Gesture at the Graphic Interface". Computer Graphics, 14(3), 1980, pp. 262-270.
- 3 Bellik, Y. and Teil, D. ``Les types de multimodalites", In Proc. IIM'92 (Paris), pp. 22-28.
- 4 Cohen, M., Murveit, H., Bernstein, J., Price, P., Weintraub, M., ``The DECIPHER Speech Recognition System". 1990 IEEE ICASSP, pp. 77-80.
- 5 Cohen, P.R., Cheyer, A., Wang, M. and Baeg, S.C. ``An Open Agent Architecture". In Proc. AAI'94 - SA (Stanford), pp. 1-8.
- 6 Cohen, P. ``The role of natural language in a multimodal interface". Proceedings of UIST'92, 143-149.
- 7 Dauphin DTR-1 User's Manual, Dauphin Technology, Inc. 337 E. Butterfield Rd., Suite 900, Lombard, Ill 60148.
- 8 Dowding, J., Gawron, J.M., Appelt, D., Bear, J., Cherny, L., Moore, B. and Moran D., ``Gemini: A natural language system for spoken-language understanding", Technical Note 527, AI Center, SRI International, 333 Ravenswood Ave., Menlo Park, CA 94025, April 1993.
- 9 Faure, C. and Julia, L. ``An Agent-Based Architecture for a Multimodal Interface". In Proc. AAI'94 - IM4S (Stanford), pp. 82-86.
- 10 Genesereth, M. and Singh, N.P. ``A knowledge sharing approach to software interoperation". Computer Science Department, Stanford University, unpublished ms., 1994.
- 11 General Magic, Inc., ``Telescript Product Documentation", 1995.

- 12 Julia, L. and Faure, C. "A Multimodal Interface for Incremental Graphic Document Design". HCI International '93, Orlando.
- 13 Koons, D.B., Sparrell, C.J., and Thorisson, K.R. "Integrating Simultaneous Input from Speech, Gaze and Hand Gestures". In *Intelligent Multimedia Interfaces*, Edited by Mark Maybury, Menlo Park, CA, AAAI Press, 1993.
- 14 Maybury, M.T. (ed.), *Intelligent Multimedia Interfaces*, AAAI Press/MIT Press: Menlo Park, Ca, 1993.
- 15 Neal, J.G., and Shapiro, S.C. "Intelligent Multi-media Interface Technology". In *Intelligent User Interfaces*, Edited by J. Sullivan and S. Tyler, Addison-Wesley Pub. Co., Reading, MA, 1991.
- 16 Nigay, L. and Coutaz, J. "A Design Space for Multimodal Systems: Concurrent Processing and Data Fusion". In Proc. InterCHI'93 (Amsterdam), ACM Press, pp. 172-178.
- 17 Object Management Group, "The Common Object Request Broker: Architecture and Specification", OMG Document Number 91.12.1, December 1991.
- 18 Oviatt, S. "Toward Empirically-Based Design of Multimodal Dialogue Systems". In Proc. AAAI'94 - IM4S (Stanford), pp. 30-36.
- 19 Oviatt, S. "Multimodal Interfaces for Dynamic Interactive Maps". In Proc. CHI '96, (Vancouver), pp. 95-102.
- 20 Oviatt, S. and Olsen, E. "Integration Themes in Multimodal Human-Computer Interaction". Proceedings of ICSLP'94, Yokohama, pp. 551-554.
- 21 Park, S.K., Choi J.M., Myeong-Wuk J., Lee G.L., and Lim Y.H. "MASCOS : A Multi-Agent System as the Computer Secretary". Submitted for publication.
- 22 Pfaff, G. and Ten Hagen, P.J.W. *Seeheim workshop on User Interface Management Systems* (Berlin), Springer- Verlag.
- 23 Rhyne J. "Dialogue Management for Gestural Interfaces". *Computer Graphics*, 21(2), 1987, pp. 137-142.
- 24

Schwartz, D.G. ``Cooperating heterogeneous systems: A blackboard-based meta approach". Technical Report 93-112, Center for Automation and Intelligent Systems Research, Case Western Reserve University, Cleveland Ohio, April 1993. Unpublished PhD. thesis.

25

Sullivan, J. and Tyler, S. (eds.), *Intelligent User Interfaces*, Addison-Wesley Pub. Co., Reading, MA, 1991.

26

Warren, D. and Pereira, F., ``An Efficient Easily Adaptable System for Interpreting Natural Language Queries", in *American Journal of Computational Linguistics*, 8(3), 1982, pp. 110-123.

27

Wauchope, K., ``Eucalyptus: Integrating Natural Language with a Graphical User Interface." Naval Research Laboratory Technical Report NRL/FR/5510-94-9711, in press, 1994.

Adam Cheyer

Mon Aug 12 15:07:21 PDT 1996

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

DISH NETWORK CORPORATION and DISH NETWORK L.L.C.,
Petitioner

v.

IPA TECHNOLOGIES, INC.,
Patent Owner

U.S. Patent Nos. 6,523,061; 6,742,021; and 6,757,718

DECLARATION OF SCOTT BENNETT, Ph.D.
16 December 2017

TABLE OF CONTENTS

| | Page |
|--|-------------|
| I. INTRODUCTION | 1 |
| II. BACKGROUND AND QUALIFICATIONS..... | 1 |
| • University Librarian, Yale University, New Haven, CT, 1994-2001;..... | 1 |
| • Director, The Milton S. Eisenhower Library, The Johns Hopkins University, Baltimore, MD, 1989-1994;..... | 1 |
| • Assistant University Librarian for Collection Management, Northwestern University, Evanston, | 2 |
| • Assistant Professor of English, University of | 2 |
| III. PRELIMINARIES..... | 3 |
| IV. OPINIONS REGARDING INDIVIDUAL DOCUMENTS | 9 |
| <i>Authentication</i> | 9 |
| <i>Public Accessibility</i> | 11 |
| <i>Conclusion</i> | 13 |
| V. ATTACHMENTS | 13 |
| VI. CONCLUSION | 14 |

I, Scott Bennett, hereby declare under penalty of perjury:

I. INTRODUCTION

1. I have personal knowledge of the facts and opinions set forth in this declaration, I believe them to be true, and if called upon to do so, I would testify competently to them. I have been warned that willful false statements and the like are punishable by fine or imprisonment, or both.

2. I am a retired academic librarian working as a Managing Partner of the firm Prior Art Documentation Services LLC at 711 South Race Street, Urbana, IL, 61801-4132. Attached as Appendix A is a true and correct copy of my Curriculum Vitae describing my background and experience.

3. I have been retained by Baker Botts LLP to authenticate and establish the dates of public accessibility of certain documents in *inter partes* review proceedings for U.S. Patent Nos. 6,523,061; 6,742,021; and 6,757,718. For this service, I am being paid my usual hourly fee of \$91/hour. My compensation in no way depends on the substance of my testimony or the outcome of this proceeding.

II. BACKGROUND AND QUALIFICATIONS

4. I was previously employed as follows:

- University Librarian, Yale University, New Haven, CT, 1994-2001;
- Director, The Milton S. Eisenhower Library, The Johns Hopkins University, Baltimore, MD, 1989-1994;

- Assistant University Librarian for Collection Management, Northwestern University, Evanston, IL, 1981-1989;
- Instructor, Assistant, and Associate Professor of Library Administration, University of Illinois at Urbana-Champaign, Urbana, IL, 1974-1981; and
- Assistant Professor of English, University of Illinois at Urbana-Champaign, 1967-1974.

5. Over the course of my work as a librarian, professor of English, researcher, and author of nearly fifty scholarly papers and other publications, I have had extensive experience with catalog records and online library management systems built around Machine-Readable Cataloging (MARC) standards. I also have substantial experience in authenticating printed documents and establishing the date when they were accessible to researchers.

6. In the course of more than fifty years of academic life, I have myself been an active researcher. I have collaborated with many individual researchers and, as a librarian, worked in the services of thousands of researchers at four prominent research universities. Over the years, I have read some of the voluminous professional literature on the information seeking behaviors of academic researchers. And as an educator, I have a broad knowledge of the ways in which students in a variety of disciplines learn to master the bibliographic

resources used in their disciplines. In all of these ways, I have a general knowledge of how researchers work.

III. PRELIMINARIES

7. *Scope of this declaration.* I am not a lawyer and I am not rendering an opinion on the legal question of whether any particular document is, or is not, a “printed publication” under the law.

8. I am, however, rendering my expert opinion on the authenticity of the documents referenced herein and on when and how each of these documents was disseminated or otherwise made available to the extent that persons interested and ordinarily skilled in the subject matter or art, exercising reasonable diligence, could have located the documents before January 5, 1998.

9. I am informed by counsel that an item is considered authentic if there is sufficient evidence to support a finding that the item is what it is claims to be. I am also informed that authenticity can be established based on the contents of the documents themselves, such as the appearance, contents, substance, internal patterns, or other distinctive characteristics of the item, taken together with all of the circumstances. I am further informed that an item is considered authentic if it is at least 20 years old, in a condition that creates no suspicion of its authenticity, and in a place where, if authentic, it would likely be.

10. I am informed by counsel that a given reference is publicly accessible upon a satisfactory showing that such document has been disseminated or otherwise made available to the extent that persons interested and ordinarily skilled in the subject matter or art exercising reasonable diligence can locate it. I have also been informed by counsel that materials available in a library constitute printed publications if they are cataloged and indexed (such as by subject) according to general library practices that make the references available to members of the interested public.

11. *Materials considered.* In forming the opinions expressed in this declaration, I have reviewed the documents and attachments referenced herein. These materials are records created in the ordinary course of business by publishers, libraries, indexing services, and others. From my years of experience, I am familiar with the process for creating many of these records, and I know these records are created by people with knowledge of the information in the record. Further, these records are created with the expectation that researchers and other members of the public will use them. All materials cited in this declaration and its attachments are of a type that experts in my field would reasonably rely upon and refer to in forming their opinions.

12. *Persons of ordinary skill in the art.* I am told by counsel that the subject matter of this proceeding relates to the navigation of electronic data.

13. I have been informed by counsel that a “person of ordinary skill in the art at the time of the inventions” is a hypothetical person who is presumed to be familiar with the relevant field and its literature at the time of the inventions. This hypothetical person is also a person of ordinary creativity, capable of understanding the scientific principles applicable to the pertinent field.

14. I am told by counsel that persons of ordinary skill in this subject matter or art would have had at least a Bachelor of Science in Computer Science, Computer Engineering, Electrical Engineering, or an equivalent field as well as at least 2 years of academic or industry experience in any type of data navigation technology.

15. It is my opinion that such a person would have been engaged in academic research, learning through study and practice in the field and possibly through formal instruction the bibliographic resources relevant to his or her research. In the 1980s and 1990s such a person would have had access to a vast array of long-established print resources in electrical/computer engineering and computer science as well as to a rich and fast changing set of online resources providing indexing information, abstracts, and full text services for electrical/computer engineering and computer science.

16. *Library catalog records.* Some background on MARC formatted records, OCLC, WorldCat, and OCLC's Connexion is needed to understand the library catalog records discussed in this declaration.

17. Libraries world-wide use the MARC format for catalog records; this machine readable format was developed at the Library of Congress in the 1960s.

18. The MARC Field 008 identifies the date when this first catalog record was entered on the file. This date persists in all subsequent uses of the first catalog record, although newly-created records for the same document, separate from the original record, will show a new date. It is not unusual to find multiple catalog records for the same document.

19. WorldCat is the world's largest public online catalog, maintained by the Online Computer Library Center, Inc., or OCLC, and built with the records created by the thousands of libraries that are members of OCLC. WorldCat provides a user-friendly interface for the public to use MARC records; it requires no knowledge of MARC tags and codes. WorldCat records appear in many different catalogs, including the Statewide Illinois Library Catalog. The date a given catalog record was created (corresponding to the MARC Field 008) appears in some detailed WorldCat records as the Date of Entry.

20. Whereas WorldCat records are very widely available, the availability of MARC formatted records varies from library to library.

21. When an OCLC participating institution acquires a document for which it finds no previously created record in OCLC, or when the institution chooses not to use an existing record, it creates a record for the document using OCLC's Connexion, the bibliographic system used by catalogers to create MARC records. Connexion automatically supplies the date of record creation in the MARC Field 008.

22. Once the MARC record is created by a cataloger at an OCLC participating member institution, it becomes available to other OCLC participating members in Connexion and also in WorldCat, where persons interested and ordinarily skilled in the subject matter or art, exercising reasonable diligence, can locate it.

23. When a book has been cataloged, it will normally be made available to readers soon thereafter—normally within a few days or (at most) within a few weeks of cataloging.

24. *Publications in series.* A library typically creates a MARC catalog record for a series of closely related publications, such as the proceedings of an annual conference, when the library receives its first issue. When the institution receives subsequent issues/volumes of the series, the issues/volumes are checked in (sometimes using a date stamp), added to the institution's holdings records, and

made available very soon thereafter—normally within a few days of receipt or (at most) within a few weeks of receipt.

25. The initial series record will often not reflect all of the subsequent changes in publication details (including minor variations in title, etc.).

26. When a library does not intend systematically to acquire all publications in a given series, but adds individual volumes of the series to its collections, the library will typically treat each such volume as an individual book, or monograph. In this case, the 008 Field MARC will record the date when the record for that individual volume, not the series, was created.

27. It is sometimes possible to find both a series and a monograph library catalog record for the same publication.

IV. OPINIONS REGARDING INDIVIDUAL DOCUMENTS

Document 1. Adam Cheyer and Luc Julia, Multimodal Maps: An Agent-Based Approach.”

First published in the Proceedings of the International Conference on Cooperative Multimodal Communication: CMC /95, Endhoven, May 24-26, 1995, pp. 103-113.

Subsequently published in Multimodal Human Computer Communication: Systems, Techniques, and Experiments [selected papers from the First International Conference on Cooperative Multimodal Communication, Eindhoven, May 1995], Harry C. Bunt, ed., in Lecture Notes in Artificial Intelligence 1374 (Berlin: Springer, 1998), pp. 111-121.

Authentication

28. Document 1 is a research paper by Adam Cheyer and Luc Julia, presented at the first International Conference on Cooperative Multimodal Communication, Eindhoven, May 1995, and published both in the 1995 proceedings of that conference and in a selection of papers from the May 1995 conference issued by Springer in 1998.

29. Attachment 1a is a true and accurate online copy of Part I of the proceedings of the International Conference on Cooperative Multimodal Communication CMC/95, including Document 1, from a Technische Universiteit Endhoven Web site <https://pure.tue.nl/ws/files/4264441/466003-1.pdf>. Attachment 1b is a true and accurate print copy of Document 1, along with the cover, title page and title page verso, preface, conference committee information, and contents pages, from the Universitat Bibliothek Erlangen-Nürnberg. Attachment 1c is a

true and accurate copy of that library's catalog record for the CMC/95 conference, in which Document 1 was published. I have compared Document 1 in Attachments 1a and 1b and find them to be substantively identical.

30. Attachment 1d is a true and accurate copy of Document 1 as published in the Springer book (along with the book's cover, preliminary leaves, half title page, title page and title page verso, preface, and contents pages) from the University of Illinois at Urbana-Champaign Library. Attachment 1e is a true and accurate copy of the University of Illinois at Urbana-Champaign Library catalog record for Multimodal Human Computer Communication, showing the series title Lecture Notes in Computer Science 1374 and the holdings for volume 1374 of that series.

31. Attachments 1a, 1b, and 1d are in a condition that creates no suspicion about their authenticity. Specifically, Document 1 in these Attachments is not missing any intermediate pages of the article's text, the text on each page appears to flow seamlessly from one page to the next, and there are no visible alterations to the document. Attachment 1a was found at a Technische Universiteit Eindhoven Web site, and Attachments 1b and 1d were found within the custody of libraries—places where, if authentic, these documents would likely be found.

32. I conclude, based on finding Document 1 both online and in libraries and on finding library catalog records for Document 1, that Document 1 is an

authentic document and that Attachments 1a, 1b, and 1d are an authentic copies of Document 1.

Public Accessibility

33. Document 1 entered the realm of public discourse in late May 1995, when it was presented at the International Conference on Cooperative Multimodal Communication / CMC /95, Endhoven, the Netherlands. The scope of the conference is suggested by the 20 papers and 7 posters presented there, as indicated by the Attachment 1a table of contents.

34. Attachment 1a, an online copy of Part I of the CMC/95 conference proceedings from a Technische Universiteit Endhoven Web site, includes a library label from the Bibliotheek Instituut voor Perceptie Onderzoek [Institute for Perception Research], a unit of the Technische Universiteit Endhoven. The University cover sheet in Attachment 1a identifies Attachment 1a as the “Publisher’s PDF, also known as Version of Record.”

35. Attachment 1f is a true and accurate copy of the Statewide Illinois Library Catalog record for the Proceedings of the International Conference on Cooperative Multimodal Communication: CMC 95, showing this volume was published in 1995 and is held by 1 library world-wide. The date of entry for this record is 4 August 1995. An ordinarily skilled researcher could have discovered the Attachment 1f catalog record by using at least three different search strategies:

(1) by looking for the title of the publication, i.e., Proceedings of the International Conference on Cooperative Multimodal Communication; (2) by looking for the conference name, i.e., International Conference on Cooperative Multimedia Communication; and (3) by looking for the editor's name, i.e., Harry Bunt. In my opinion, an ordinarily skilled researcher, exercising reasonable diligence, would have had no difficulty finding copies of the Proceedings of the International Conference on Cooperative Multimodal Communication: CMC 95 in 1995.

36. I conclude that Document 1, as published in the Proceedings of the International Conference on Cooperative Multimodal Communication / CMC 95, was publicly available in at least one library by September 1995.

37. Attachment 1g is a true and accurate copy of the Statewide Illinois Library Catalog record for the Springer book, Multimodal Human Computer Communication, showing this volume was published in 1998 and is held by 10 libraries world-wide. The date of entry for this record is 29 April 1998. An ordinarily skilled researcher could have discovered the Attachment 1g catalog record by using at least four different search strategies: (1) by looking for the title of the publication, i.e., Multimodal human computer communication; (2) by looking for the conference name, i.e., International Conference on Cooperative Multimedia Communication; (3) by looking for the editor's name, i.e., Harry Bunt; and (4) by looking for the name of the Springer series, i.e., Lecture notes in

computer science. In my opinion, an ordinarily skilled researcher, exercising reasonable diligence, would have had no difficulty finding copies of the Springer book, *Multimodal Human Computer Communication* in 1998.

38. I conclude that Document 1, as published in the Springer book *Multimodal Human Computer Communication*, was publicly available in at least one library at least by May 1998.

Conclusion

39. Based on the evidence presented here—online and book publication, and library records,—it is my opinion that Document 1, as published in the *Proceedings of the International Conference on Cooperative Multimodal Communication / CMC 95*, was publicly available in at least one library by no later than September 1995. It is my further opinion that Document 1, as published in the Springer book *Multimodal Human Computer Communication*, was publicly available in at least one library at least by May 1998.

V. ATTACHMENTS

40. The attachments attached hereto are true and correct copies of the materials identified above. Helen Sullivan is a Managing Partner in Prior Art Documentation Services LLC. One of her primary responsibilities in our

partnership is to secure the bibliographic documentation used in attachments to our declarations.

41. Ms. Sullivan and I work in close collaboration on the bibliographic documentation needed in each declaration. I will sometimes request specific bibliographic documents or, more rarely, secure them myself. In all cases, I have carefully reviewed the bibliographic documentation used in my declaration. My signature on the declaration indicates my full confidence in the authenticity, accuracy, and reliability of the bibliographic documentation used.

42. Each Attachment has been marked with an identifying label on the top of each page. However, no alterations other than these noted labels appear in these attachments, unless otherwise noted. All attachments were created on 24 August – 14 September 2017 and all URLs referenced in this declaration were available 14 September 2017.

VI. CONCLUSION

43. I reserve the right to supplement my opinions in the future to respond to any arguments that Patent Owner or its expert(s) may raise and to take into account new information as it becomes available to me.

44. I declare that all statements made herein of my knowledge are true, and that all statements made on information and belief are believed to be true, and that these statements were made with the knowledge that willful false statements

and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Executed this 16th day of December, 2017 in Urbana, Illinois.

A handwritten signature in blue ink that reads "Scott Bennett". The signature is written in a cursive style with a large initial "S".

Scott Bennett

Appendix A

SCOTT BENNETT
Yale University Librarian Emeritus

711 South Race
Urbana, Illinois 61801-4132
2scottbb@gmail.com
217-367-9896

EMPLOYMENT

Retired, 2001. Retirement activities include:

- Managing Partner in Prior Art Documentation Services, LLC, 2015-. This firm provides documentation services to patent attorneys
- Consultant on library space design, 2004-2017 . This consulting practice was rooted in a research, publication, and public speaking program conducted since I retired from Yale University in 2001. I served more than 50 colleges and universities in the United States and abroad with projects ranging in likely cost from under \$50,000 to over \$100 million.
- Senior Advisor for the library program of the **Council of Independent Colleges**, 2001-2009
- Member of the Wartburg College Library Advisory Board, 2004-
- Visiting Professor, Graduate School of Library and Information Science, **University of Illinois at Urbana-Champaign**, Fall 2003

University Librarian, **Yale University**, 1994-2001

Director, The Milton S. Eisenhower Library, **The Johns Hopkins University**, Baltimore, Maryland, 1989-1994

Assistant University Librarian for Collection Management, **Northwestern University**, Evanston, Illinois, 1981-1989

Instructor, Assistant and Associate Professor of Library Administration, **University of Illinois at Urbana-Champaign**, 1974-1981

Assistant Professor of English, **University of Illinois at Urbana-Champaign**, 1967-1974

Woodrow Wilson Teaching Intern, **St. Paul's College**, Lawrenceville, Virginia, 1964-1965

EDUCATION

University of Illinois, M.S., 1976 (Library Science)
Indiana University, M.A., 1966; Ph.D., 1967 (English)
Oberlin College, A.B. magna cum laude, 1960 (English)

HONORS AND AWARDS

Morningside College (Sioux City, IA) Doctor of Humane Letters, 2010

American Council of Learned Societies Fellowship, 1978-1979; Honorary Visiting Research Fellow, Victorian Studies Centre, **University of Leicester**, 1979; **University of Illinois** Summer Faculty Fellowship, 1969

Indiana University Dissertation Year Fellowship and an **Oberlin College** Haskell Fellowship, 1966-1967; **Woodrow Wilson** National Fellow, 1960-1961

PROFESSIONAL ACTIVITIES

American Association for the Advancement of Science: Project on Intellectual Property and Electronic Publishing in Science, 1999-2001

American Association of University Professors: University of Illinois at Urbana-Champaign Chapter Secretary and President, 1975-1978; Illinois Conference Vice President and President, 1978-1984; national Council, 1982-1985, Committee F, 1982-1986, Assembly of State Conferences Executive Committee, 1983-1986, and Committee H, 1997-2001 ; Northwestern University Chapter Secretary/Treasurer, 1985-1986

Association of American Universities: Member of the Research Libraries Task Force on Intellectual Property Rights in an Electronic Environment, 1993-1994, 1995-1996

Association of Research Libraries: Member of the Preservation Committee, 1990-1993; member of the Information Policy Committee, 1993-1995; member of the Working Group on Copyright, 1994-2001; member of the Research Library Leadership and Management Committee, 1999-2001; member of the Board of Directors, 1998-2000

Carnegie Mellon University: Member of the University Libraries Advisory Board, 1994

Center for Research Libraries: Program Committee, 1998-2000

Johns Hopkins University Press: Ex-officio member of the Editorial Board, 1990-1994; Co-director of Project Muse, 1994

Library Administration and Management Association, Public Relations Section, Friends of the Library Committee, 1977-1978

Oberlin College: Member of the Library Visiting Committee, 1990, and of the Steering Committee for the library's capital campaign, 1992-1993; President of the Library Friends, 1992-1993, 2004-2005; member, Friends of the Library Council, 2003-

Research Society for Victorian Periodicals: Executive Board, 1971-1983; Co-chairperson of the Executive Committee on Serials Bibliography, 1976-1982; President, 1977-1982

A Selected Edition of W.D. Howells (one of several editions sponsored by the MLA Center for Editions of American Authors): Associate Textual Editor, 1965-1970; Center for Editions of American Authors panel of textual experts, 1968-1970

Victorian Studies: Editorial Assistant and Managing Editor, 1962-1964

Wartburg College: member, National Advisory Board for the Vogel Library, 2004-

Some other activities: Member of the **Illinois State Library** Statewide Library and Archival Preservation Advisory Panel; member of the **Illinois State Archives** Advisory Board; member of a committee advising the **Illinois Board of Higher Education** on the cooperative management of research collections; chair of a major collaborative research project conducted by the **Research Libraries Group** with support from Conoco, Inc.; active advisor on behalf of the **Illinois Conference AAUP** to faculty and administrators on academic freedom and tenure matters in northern Illinois.

Delegate to **Maryland Governor's Conference on Libraries and Information Service**; principal in initiating state-wide preservation planning in Maryland; principal in an effort to widen the use of mass deacidification for the preservation of library materials through cooperative action by the **Association of Research Libraries** and the **Committee on Institutional Cooperation**; co-instigator of a campus-wide information service for **Johns Hopkins University**; initiated efforts with the **Enoch Pratt Free Library** to provide information services to Baltimore's Empowerment Zones; speaker or panelist on academic publishing, copyright, scholarly communication, national and regional preservation planning, mass deacidification.

Consultant for the **University of British Columbia** (1995), **Princeton University** (1996), **Modern Language Association**, (1995, 1996), **Library of Congress** (1997), **Center for Jewish History** (1998, 2000-), **National Research Council** (1998); Board of Directors for the **Digital Library Federation**, 1996-2001; accreditation visiting team at **Brandeis University** (1997); mentor for **Northern Exposure to Leadership** (1997); instructor and mentor for ARL's **Leadership and Career Development Program** (1999-2000)

At the **Northwestern University Library**, led in the creation of a preservation department and in the renovation of the renovation, for preservation purposes, of the Deering Library book stacks.

At the **Milton S. Eisenhower Library**, led the refocusing and vitalization of client-centered services; strategic planning and organizational restructuring for the library; building renovation planning. Successfully completed a \$5 million endowment campaign for the humanities collections and launched a \$27 million capital campaign for the library.

At the **Yale University Library**, participated widely in campus-space planning, university budget planning, information technology development, and the promotion of effective teaching and learning; for the library has exercised leadership in space planning and renovation, retrospective conversion of the card catalog, preservation, organizational development, recruitment of minority librarians, intellectual property and copyright issues, scholarly communication, document delivery services among libraries, and instruction in the use of information resources. Oversaw approximately \$70 million of library space renovation and construction. Was co-principal investigator for a grant to plan a digital archive for Elsevier Science.

Numerous to invitations speak at regional, national, and other professional meetings and at alumni meetings. Lectured and presented a series of seminars on library management at the **Yunnan University Library**, 2002. Participated in the 2005 International Roundtable for Library and Information Science sponsored by the **Kanazawa Institute of Technology** Library Center and the Council on Library and Information Resources.

PUBLICATIONS

“Putting Learning into Library Planning,” *portal: Libraries and the Academy*, 15, 2 (April 2015), 215-231.

“How librarians (and others!) love silos: Three stories from the field “ available at the Learning Spaces Collaboratory Web site, <http://www.pkallsc.org/>

“Learning Behaviors and Learning Spaces,” *portal: Libraries and the Academy*, 11, 3 (July 2011), 765-789.

“Libraries and Learning: A History of Paradigm Change,” *portal: Libraries and the Academy*, 9, 2 (April 2009), 181-197. Judged as the best article published in the 2009 volume of *portal*.

“The Information or the Learning Commons: Which Will We Have?” *Journal of Academic Librarianship*, 34 (May 2008), 183-185. One of the ten most-cited articles published in JAL, 2007-2011.

“Designing for Uncertainty: Three Approaches,” *Journal of Academic Librarianship*, 33 (2007), 165–179.

“Campus Cultures Fostering Information Literacy,” *portal: Libraries and the Academy*, 7 (2007), 147-167. Included in Library Instruction Round Table Top Twenty library instruction articles published in 2007

“Designing for Uncertainty: Three Approaches,” *Journal of Academic Librarianship*, 33 (2007), 165–179.

“First Questions for Designing Higher Education Learning Spaces,” *Journal of Academic Librarianship*, 33 (2007), 14-26.

“The Choice for Learning,” *Journal of Academic Librarianship*, 32 (2006), 3-13.

With Richard A. O’Connor, “The Power of Place in Learning,” *Planning for Higher Education*, 33 (June-August 2005), 28-30

“Righting the Balance,” in *Library as Place: Rethinking Roles, Rethinking Space* (Washington, DC: Council on Library and Information Resources, 2005), pp. 10-24

Libraries Designed for Learning (Washington, DC: Council on Library and Information Resources, 2003)

“The Golden Age of Libraries,” in *Proceedings of the International Conference on Academic Librarianship in the New Millennium: Roles, Trends, and Global Collaboration*, ed. Haipeng Li (Kunming: Yunnan University Press, 2002), pp. 13-21. This is a slightly different version of the following item.

“The Golden Age of Libraries,” *Journal of Academic Librarianship*, 24 (2001), 256-258

“Second Chances. An address . . . at the annual dinner of the Friends of the Oberlin College Library November 13 1999,” Friends of the Oberlin College Library, February 2000

“Authors’ Rights,” *The Journal of Electronic Publishing* (December 1999), <http://www.press.umich.edu/jep/05-02/bennett.html>

“Information-Based Productivity,” in *Technology and Scholarly Communication*, ed. Richard Ekman and Richard E. Quandt (Berkeley, 1999), pp. 73-94

“Just-In-Time Scholarly Monographs: or, Is There a Cavalry Bugle Call for Beleaguered Authors and Publishers?” *The Journal of Electronic Publishing* (September 1998), <http://www.press.umich.edu/jep/04-01/bennett.html>

“Re-engineering Scholarly Communication: Thoughts Addressed to Authors,” *Scholarly Publishing*, 27 (1996), 185-196

“The Copyright Challenge: Strengthening the Public Interest in the Digital Age,” *Library Journal*, 15 November 1994, pp. 34-37

“The Management of Intellectual Property,” *Computers in Libraries*, 14 (May 1994), 18-20

“Repositioning University Presses in Scholarly Communication,” *Journal of Scholarly Publishing*, 25 (1994), 243-248. Reprinted in *The Essential JSP. Critical Insights into the World of Scholarly Publishing. Volume 1: University Presses* (Toronto: University of Toronto Press, 2011), pp. 147-153

“Preservation and the Economic Investment Model,” in *Preservation Research and Development. Round Table Proceedings, September 28-29, 1992*, ed. Carrie Beyer (Washington, D.C.: Library of Congress, 1993), pp. 17-18

“Copyright and Innovation in Electronic Publishing: A Commentary,” *Journal of Academic Librarianship*, 19 (1993), 87-91; reprinted in condensed form in *Library Issues: Briefings for Faculty and Administrators*, 14 (September 1993)

with Nina Matheson, “Scholarly Articles: Valuable Commodities for Universities,” *Chronicle of Higher Education*, 27 May 1992, pp. B1-B3

“Strategies for Increasing [Preservation] Productivity,” *Minutes of the [119th] Meeting [of the Association of Research Libraries]* (Washington, D.C., 1992), pp. 39-40

“Management Issues: The Director’s Perspective,” and “Cooperative Approaches to Mass Deacidification: Mid-Atlantic Region,” in *A Roundtable on Mass Deacidification*, ed. Peter G. Sparks (Washington, D.C.: Association of Research Libraries, 1992), pp. 15-18, 54-55

“The Boat that Must Stay Afloat: Academic Libraries in Hard Times,” *Scholarly Publishing*, 23 (1992), 131-137

“Buying Time: An Alternative for the Preservation of Library Material,” *ACLS Newsletter*, Second Series 3 (Summer, 1991), 10-11

“The Golden Stain of Time: Preserving Victorian Periodicals” in *Investigating Victorian Journalism*, ed. Laurel Brake, Alex Jones, and Lionel Madden (London: Macmillan, 1990), pp. 166-183

“Commentary on the Stephens and Haley Papers” in *Coordinating Cooperative Collection Development: A National Perspective*, an issue of *Resource Sharing and Information Networks*, 2 (1985), 199-201

“The Editorial Character and Readership of *The Penny Magazine*: An Analysis,” *Victorian Periodicals Review*, 17 (1984), 127-141

“Current Initiatives and Issues in Collection Management,” *Journal of Academic Librarianship*, 10 (1984), 257-261; reprinted in *Library Lit: The Best of 85*

“Revolutions in Thought: Serial Publication and the Mass Market for Reading” in *The Victorian Periodical Press: Samplings and Soundings*, ed. Joanne Shattock and Michael Wolff (Leicester: Leicester University Press, 1982), pp. 225-257

“Victorian Newspaper Advertising: Counting What Counts,” *Publishing History*, 8 (1980), 5-18

“Library Friends: A Theoretical History” in *Organizing the Library’s Support: Donors, Volunteers, Friends*, ed. D.W. Krummel, Allerton Park Institute Number 25 (Urbana: University of Illinois Graduate School of Library Science, 1980), pp. 23-32

“The Learned Professor: being a brief account of a scholar [Harris Francis Fletcher] who asked for the Moon, and got it,” *Non Solus*, 7 (1980), 5-12

“Prolegomenon to Serials Bibliography: A Report to the [Research] Society [for Victorian Periodicals],” *Victorian Periodicals Review*, 12 (1979), 3-15

“The Bibliographic Control of Victorian Periodicals” in *Victorian Periodicals: A Guide to Research*, ed. J. Don Vann and Rosemary T. VanArsdel (New York: Modern Language Association, 1978), pp. 21-51

“John Murray’s Family Library and the Cheapening of Books in Early Nineteenth Century Britain,” *Studies in Bibliography*, 29 (1976), 139-166. Reprinted in Stephen Colclough and Alexis Weedon, eds., *The History of the Book in the West: 1800-1914*, Vol. 4 (Farnham, Surrey: Ashgate, 2010), pp. 307-334.

with Robert Carringer, “Dreiser to Sandburg: Three Unpublished Letters,” *Library Chronicle*, 40 (1976), 252-256

“David Douglas and the British Publication of W. D. Howells’ Works,” *Studies in Bibliography*, 25 (1972), 107-124

as primary editor, W. D. Howells, *Indian Summer* (Bloomington: Indiana University Press, 1971)

“The Profession of Authorship: Some Problems for Descriptive Bibliography” in *Research Methods in Librarianship: Historical and Bibliographic Methods in Library Research*, ed. Rolland E. Stevens (Urbana: University of Illinois Graduate School of Library Science, 1971), pp. 74-85

edited with Ronald Gottesman, *Art and Error: Modern Textual Editing* (Bloomington: Indiana University Press, 1970)--also published in London by Methuen, 1970

“Catholic Emancipation, the *Quarterly Review*, and Britain’s Constitutional Revolution,” *Victorian Studies*, 12 (1969), 283-304

as textual editor, W. D. Howells, *The Altrurian Romances* (Bloomington: Indiana University Press, 1968); introduction and annotation by Clara and Rudolf Kirk

as associate textual editor, W. D. Howells, *Their Wedding Journey* (Bloomington: Indiana University Press, 1968); introduction by John Reeves

“A Concealed Printing in W. D. Howells,” *Papers of the Bibliographic Society of America*, 61 (1967), 56-60

editor, *Non Solus*, A Publication of the University of Illinois Library Friends, 1974-1981

editor, Robert B. Downs Publication Fund, University of Illinois Library, 1975-1981

Reviews, short articles, etc. in *Victorian Studies*, *Journal of English and German Philology*, *Victorian Periodicals Newsletter*, *Collection Management*, *Nineteenth-Century Literature*, *College & Research Libraries*, *Scholarly Publishing Today*, *ARL Newsletter*, *Serials Review*, *Library Issues*, *S[society for] S[scholarly] P[ublishing] Newsletter*, and *Victorian Britain: An Encyclopedia*

Proceedings of the international conference on cooperative multimodel communication CMC/95, Eindhoven, May 24-26, 1995

Bunt, H.C.; Beun, R.J.; Borghuis, V.A.J.

Published: 01/01/1995

Document Version

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the author's version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

Citation for published version (APA):

Bunt, H. C., Beun, R. J., & Borghuis, V. A. J. (Eds.) (1995). Proceedings of the international conference on cooperative multimodel communication CMC/95, Eindhoven, May 24-26, 1995: proceedings. (CMC : cooperative multimodel communication : international conference; Vol. 1). Eindhoven: DENK: Samenwerkingsorgaan Brabantse Universiteiten.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 14. Sep. 2017

**Proceedings of the
International Conference on
Cooperative Multimodal
Communication CMC/95
Part I**

Eindhoven, May 24-26, 1995

Harry Bunt, Robbert-Jan Beun & Tijn Borghuis (eds.)

I P O



9690453

E I N D H O V E N

**ex
32/1**

**BIBLIOTHEEK
INSTITUUT VOOR PERCEPTIE
ONDERZOEK**

CIP GEGEVENS KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Harry Bunt, Robbert-Jan Beun & Tijn Borghuis

Proceedings of the International Conference on Cooperative
Multimodal Communication, Eindhoven, May 24-26, 1995

ISBN 90-9008315-4

trefw.: mens-machine communicatie, multimedia, user-interfaces

Preface

Communication is a bidirectional activity that comes naturally in multimodal form, involving both verbal and nonverbal, vocal, visual, tactile and other means of interaction. Natural communication is also cooperative, in that the participants make an effort to understand each other, and act in a way that takes each other's goals and purposes into account, for instance helping a dialogue partner to obtain relevant information.

Technical developments increasingly allow the realization of human-computer interfaces where more sophisticated forms of visual and auditory, verbal and nonverbal information are used by the computer and where the user is allowed a greater variety of forms of expression. Two crucial aspects of natural communication are, however, still conspicuously absent in existing user interfaces:

- real cooperation from the part of the computer, based on a good understanding of the user's wants;
- true multimodality in the sense of fully integrated, simultaneous use of several modalities to convey a complex message.

As a result, human-computer communication is generally felt to be only marginally cooperative, and to be unnatural and primitive, compared to natural human communication.

The present conference aims at contributing to improving the state of the art in cooperative multimodal human-computer communication, bringing together researchers involved in the design, implementation, and application of forms of cooperative human-computer communication where natural language (typed or spoken) is used in combination with other modalities, such as visual feedback and direct manipulation. The conference focuses on formal, computational, and user aspects of building cooperative multimodal dialogue systems, with the following topics being identified in the call for papers:

- cooperativity in multimodal dialogue
- natural language semantics in a multimodal context
- formal and computational models of dialogue context
- incremental knowledge representation and dialogue
- interacting with visual domain representations
- collaborative problem solving
- constraint-based approaches to animation and visual modelling
- effective use of different interactive modalities
- modelling temporal aspects of multimodal communication
- type theory and natural language interpretation

In response to the call for papers, we have received submissions from all over the world (Europe, North America, Asia, Australia), from which the programme committee has selected 17 for paper presentation and 8 for poster presentation at the conference. In addition, the conference features a presentation of the multimodal DenK-project, which has provided the inspiration for organizing this conference, and invited papers by Mark Maybury, Wolfgang Wahlster, Bonnie Webber and Kent Wittenburg.

I would like to use this occasion to thank the members of the programme committee for reviewing the submitted contributions for the conference, and the members of the organizing committee plus the staff at the Institute for Perception Research IPO, which hosts the conference, for all their efforts to make the conference run smoothly. Particular thanks are due to the Samenwerkingsorgaan Brabantse Universiteiten (the organization for cooperation between the universities in the province of Brabant, i.e. the universities of Tilburg and Eindhoven), and to the Royal Dutch Academy of Sciences (KNAW) for their financial support.

Harry Bunt
Program Committee chairman.

Program Committee

Harry Bunt (chair)

Norman Badler

Walther von Hahn

Hans Kamp

Joseph Mariani

Paul Mc Kevitt

Kees van Overveld

Donia Scott

Bonnie Webber

Jeroen Groenendijk

Dieter Huber

John Lee

Mark Maybury

Rob Nederpelt

Ray Perrault

Wolfgang Wahlster

Kent Wittenburg

Organizing Committee

Robbert-Jan Beun (chair)

Tijn Borghuis

Harry Bunt

Rob Nederpelt

Marianne Wagemans

Sponsorship

Koninklijke Nederlandse Akademie van Wetenschappen (KNAW)

Samenwerkingsorgaan Brabantse Universiteiten (SOBU)

ACL Special Interest Group in Multimedia (SIGMEDIA)

Table of Contents

PART I

Invited Papers

Toward Cooperative Multimedia Interaction (abstract) 3
Mark T. Maybury

Instructing Animated Agents: Viewing Language in Behavioral Terms 5
Bonnie Webber

Visual Language Parsing: If I Had a Hammer... 17
Kent Wittenburg

Submitted Papers

Contexts in Dialogue 37
Tijn Borghuis

Management of Non-Standard Devices for Multimodal User Interfaces under UNIX/X11 49
Patrick Bourdot, Mike Krus and Rachid Gherbi

The Role of Multimodal Communication in Cooperation and Intention Recognition: The Case of Air Traffic Control 63
Marie-Christine Bressolle, Bernard Pavard and Marcel Leroux

Cooperative Multimodal Communication in the DenK Project 79
Harry Bunt, René Ahn, Robbert-Jan Beun, Tijn Borghuis and Kees van Overveld

Multimodal Maps: An Agent Based Approach 103
Adam Cheyer and Luc Julia

Object Reference During Task-related Terminal Dialogues 115
Anita Cremers

Speakers' Responses to Requests for Repetition in a Multimedia Language Processing Environment 129
Laurel Fais, Kyung-ho Loken-Kim and Young-Duk Park

A Cooperative Approach for Multimodal Presentation Planning 145
Yi Han and Ingrid Zukerman

PART II

Studies into Full Integration of Language and Action 161
Carla Huls and Edwin Bos

Referent Identification Requests in Multimodal Dialogues 175
Tsuneaki Kato and Yukiko I. Nakano

Anaphora in Multimodal Discourse 193
John Lee and Keith Stenning

Towards Adequate Representation Technologies for Multimodal Interfaces . 207
Jean Claude Martin, Remko Veldman and Dominique Bérroule

**Designing a Multimedia Interface for Operators Assembling
Circuit Boards** 225
Fergal McCaffery, Michael McTear and Maureen Murphy

Synthesizing Cooperative Conversation 237
*Catherine Pelachaud, Justine Cassell, Norman Badler, Mark Steedman,
Scott Prevost and Matthew Stone*

Topic Management in Information Dialogues 257
Mieke Rats

**An Approach to Solving the Symbol Grounding Problem: Neural Networks
for Object Naming and Retrieval** 273
N.J. Sales and R.G. Evans

**Modeling and Processing of the Oral and Tactile Activities
in the Georal Tactile System** 287
J. Siroux, M. Guyomard, F. Multon and C. Remondeau

Poster presentations

**The Generalized Display Processor: a Platform for Real-time
Interactive Computer Animation** 299
Gino van Bergen and Kees van Overveld

**Communication and Co-ordination Difficulties During Interactive
Tele-Teaching in the Independent Problem-Solving Format** 301
Martin Colbert

Automatic Generation of Statistical Graphics 303
Massimo Fasciano and Guy Lapalme

Computer-Aided Negotiation Support in Hypermedia Multi-Agent Systems 307
Igor V. Kotenko and Dmitry L. Krechman

Multimodal Dialogue Semantics Against a Dynamic World Model 311
Susann Luperfoy and David Duff

Two Basic Orientations of Subject in World and in Human-Computer Communication 315
Olga I. Marchenko

Internalized Contexts in NL Semantics 317
Wlodek Zadrozny

Author Index 321

Subject Index 323

Invited Papers

Toward Cooperative Multimedia Interaction (abstract)

Mark T. Maybury

The MITRE Corporation, Artificial Intelligence Center
Mail Stop K331, 202 Burlington Road Bedford, MA 01730
e-mail: maybury@mitre.org

Key words: Content-based multimedia information retrieval, intelligent interfaces, multimedia databases.

Multimedia information and communication permeates our daily lives. We are continuously presented with multimedia information via newspapers, radios, televisions, and, increasingly, from our interactions on the global information infrastructure. When we directly converse with one another, we utilize a wide array of media to interact, including spoken language, gestures, and drawings. In both human-human communication and in our multimedia artifacts, we rely upon multiple sensory systems or modes of communication including vision, audition, and taction. Although humans have a natural facility for managing and exploiting multiple input and output media, computers do not.

Providing machines with the ability to interpret multimedia input and generate coordinated multimedia output would be a valuable facility for a number of key applications such as information retrieval and analysis, training, and decision support. A number of exciting applications are already emerging from laboratories. Examples include customized electronic newspapers, adaptive multimedia interfaces, and interactive multimedia digital libraries. Commercially, multimedia databases promise access to massive, heterogeneous collections of audio, video, text, as well as structured data. Finally, multimedia interfaces are emerging that can intelligently exploit multiple media and modes to facilitate human-computer communication.

Despite this exciting potential, serious fundamental scientific questions remain unanswered. What is the character of multimedia information? How should media be represented? What is a media and what is a mode? How do humans process multimedia information? How can we develop efficient and effective algorithms and systems to assist in the creation, management, and interaction with multimedia information? How can we leverage the rapidly converging domains of communications, information retrieval, speech, language and image processing, and virtual reality to enhance human-machine and human-human communication? What are the appropriate metrics and methods for evaluating progress in this area while at the same time encouraging scientific innovation? Can we discover what it means to support cooperative multimedia communication, or move toward understanding the principles of multimedia communication?

This talk will attempt to shed light on some of these questions by highlighting research which aims to support enhanced multimedia interaction. First, the talk will summarize techniques being developed to interpret and generate multiple media, including spoken and

written natural language, graphics, non-speech audio, maps, animation. Subsequently, it will consider how advances in text, spoken language, and image understanding can be applied to support retrieval of complex media (e.g., imagery, text & graphics, video). The talk will describe a vision of the confluence of intelligent systems and multimedia to support cooperative multimedia interaction.

Instructing Animated Agents: Viewing language in behavioral terms

Bonnie Webber

Department of Computer & Information Science
University of Pennsylvania
Philadelphia PA USA 19104-6389
e-mail: bonnie@central.cis.upenn.edu

Abstract

One activity of Penn's Center for Human Modelling and Simulation has been the exploration of Natural Language instructions and other high-level task specifications to create animated simulations of virtual human agents carrying out tasks. The work builds on JACK, an animation system developed at Penn, that provides simulated human models with a growing repertoire of naturalistic behaviors. The value in using high-level task specifications to create animated simulations is that the same specification can be used to produce different animations in different situations, without additional animator or programmer intervention.

But animated simulation driven by Natural Language instructions can provide another benefit, by forcing us to consider what aspects of language convey information relevant to behavior. What our studies to date have revealed is that more of an utterance conveys such information than its main verb and argument structure.

To demonstrate an analysis of linguistic constructs in terms of behavioral specifications and constraints, I show how instructions containing "until" clauses can be analysed in terms of perceptual activities and the conditions they are used to assess, and how the resulting analysis contributes to understanding how an agent is supposed to carry these instructions out.

keywords: task specifications, human figure animation, pragmatics.

1 Introduction

My group at Penn has been exploring the use of Natural Language instructions and other high-level task specifications to create realistic animated simulations of virtual human agents carrying out tasks. The work builds on *Jack*TM, an animation system developed at the University of Pennsylvania's Center for Human Modelling and Simulation. *Jack* provides biomechanically reasonable and anthropometrically-scaled human models with a growing repertoire of naturalistic behaviors such as walking, stepping, looking, reaching, turning, grasping, strength-based lifting, and both obstacle and self-collision avoidance [1]. The value in using high-level task specifications to create animated simulations is that the same specification will produce agent behavior that is appropriate to different environments and/or different conditions, without additional animator or programmer intervention. The resulting simulations

thus afford a relatively inexpensive way to carry out human factors studies in computer-aided design or to use in Virtual Reality training, especially in exercises involving multiple agents or a variety of environments.

But animated simulation driven by Natural Language instructions can provide another benefit, by forcing us to consider what aspects of language convey information relevant to behavior. What our studies to date have revealed is that much more of an utterance than its main verb and argument structure help an agent along. Ignoring these other sources of information can lead animated agents to behave in ways that viewers find odd, if not totally bizarre. To avoid this, more research in Natural Language Processing should address relationships between language and behavior.

To demonstrate how linguistic constructs can be analysed in terms of behavioral specifications and constraints, I will show in the main body of the paper how instructions containing "until" clauses such as

- (1) Squeeze riveter handles **until** rivet stem breaks off.

can be analysed in terms of perceptual activities and the conditions they are used to assess, and how the resulting analysis contributes to understanding what an agent is intended to do in response. Before I begin though, I want to call the reader's attention to our earlier work, as background to this presentation.

In 1990, Barbara Di Eugenio and I did a study of gerundive adjuncts in Natural Language instructions [15] such as

- (2) Unroll each strip onto the wall, *smoothing the foil into place vertically (not side to side) to avoid warping and curling at the edges.*
- (3) Sew the head front to back, *leaving the neck edge open.*
- (4) As you work, clean the surface thoroughly each time you change grits, **vacuuming off all the dust and wiping the wood with a rag dampened with turpentine or paint thinner.**

A gerundive adjunct is a type of *free adjunct* – a nonfinite predicative phrase with the function of an adverbial subordinate clause [14]. Progressive gerundive adjuncts are fairly common in instructions that specify physical activities. In his analysis of a wide range of free adjuncts in English narrative [14], Stump focussed on their truth-conditional properties, distinguishing between *strong* and *weak* adjuncts:

- (5a) *Having unusually long arms*, John can touch the ceiling.
- (5b) *Standing on the chair*, John can touch the ceiling.
- (6a) *Being a businessman*, Bill smokes cigars.
- (6b) *Lying on the beach*, Bill smokes cigars.

Stump calls the adjuncts in both a sentences *strong*, because their actual truth is uniformly entailed. He calls those in the b sentences *weak*, because their actual truth can fail to be entailed. Stump notes the causal flavor of strong adjuncts: in the a sentences above, the main clause assertion is true *because* the adjunct is. Weak adjuncts, on the other hand, have

a conditional sense: it is (only) when the condition described in the adjunct is true that the main clause assertion is true.

While Stump's observations appear to be both correct and relevant for narrative text, Di Eugenio and I were concerned with the behavioral import of gerundive adjuncts. The main interpretative decision turned out to be whether two separate actions were being specified as in Example 2 (if so, there was a further question as to the specific temporal relationship between the two actions) or only one, as in Example 3 and 4. To determine what action an agent is intended to perform in the latter case, the agent needs to determine whether the adjunct further specifies the action specified in the main clause (e.g., providing information about manner, extent, side effects to avoid, etc.), as in Example 3 or whether the action specifications in the main and adjunct clauses were related by *generation* [8], as in Example 4. In the latter, the generated action provides a reason for doing the generating action, although we found in subsequent work that the generation relation can convey more information relevant to behavior than just purpose.

Specifically, in her doctoral thesis research ([4], see also [6]), Di Eugenio focussed on instructions containing "purpose clauses" of the form

Do α to do β

showing that when they are interpreted as conveying a generation relationship, the relationship may not be between the given α and β , but between a more specific action α' and β . For example, in

(7) Cut a square in half to make two triangles.

Di Eugenio showed that the action the agent is meant to carry out is not just cutting a square in half (α) but rather the more specific action, cutting a square in half along a diagonal (α'). She then showed how a *description lattice* created in a knowledge representation formalism such as CLASSIC or LOOM can be used to carry out the relevant reasoning. The point I want to emphasize though is that *this systematic inference had gone unnoticed* until utterances were analysed in terms of specifying or constraining behavior.

2 "Until" Clauses

Instructions containing "until" clauses highlight the rule of perceptual activity in behavior and in behavioral specifications. Obviously, agents use perception when they carry out tasks: if the agent's task is building a brick wall, the agent will use perception to lay the next bed of mortar, to find the next brick to lay, to maneuver the brick to an appropriate place on the mortar bed, to notice and remove excess mortar, etc.

But Natural Language also uses perceptual tests to *specify* behavior, as I will try to show with "until" clauses. Now, in a programming language like Pascal, the Boolean condition in an "until" expression

repeat <statement-sequence> until <Boolean-expr>

can be assessed by just computing its value. However, for a human agent to comply with an instruction containing an "until" clause, the agent must

- understand the *source* of the condition to be checked and the *actions* she must take to assess it;

- understand what, if any, actions she is assumed to be doing when the condition is to be assessed;
- determine how to efficiently integrate both sets of actions.

The data on which this analysis is based are drawn from six chapters of two volumes of home repair instructions scanned in by Joseph Rosenzweig, a graduate student at the University of Pennsylvania: Dorling Kindersley's *Home Repair Encyclopedia* [10] and the *Reader's Digest New Complete Do-It-Yourself Manual* [13]. The data consist of 80 instructions containing "until clauses", of the form

(Do) α **until** κ .

The chapters were chosen randomly, not because of their subject matter, and all sentences containing "until" clauses were extracted from them. Some of the instructions concern repair jobs (e.g. fixing broken china, repairing cracked parquet, etc.), and the others concern construction of concrete, asphalt, and/or masonry structures.

While the ideas have not yet been implemented, I am assuming an agent architecture that contains, at the very least:

- one or more low-level Sense-Control-Act (S-C-A) loops, that can be modified from above by
- a process-based (as opposed to state-space) task specification such as that recently proposed by Pym, Pryor and Murphy, using the process algebra [12], or the parallel transition network representation (PaT-Net) we have begun to use in much of our animated simulation work [2, 3].

With respect to such an architecture, a wide class of Natural Language instructions, including those with "until" clauses, would be interpreted as process-based representational structures that set the S-C-A loops and interpret both their success and error conditions.

3 Assessing the Specified Condition

The first thing to note is that perception alone may be insufficient to determine whether a condition holds: one or more actions may first be necessary to bring the world into a state in which an appropriate observation can be made. Such actions I will follow Kirsh and Maglio [9] in calling *epistemic*.¹ That is, in the case of

(8) Squeeze riveter handles **until** rivet stem breaks off.

the agent does not have to do anything special to be able to observe the rivet stem breaking off. On the other hand, to determine whether the condition holds in

(9) Wait for the filler to set and rub it down, first with a needle file and then with glasspaper, **until** it lies flush with the surface.

¹Kirsh and Maglio use the term *pragmatic* action for ones whose purpose is to bring an agent closer to her goal.

the agent must assess the condition using tactile perception, which in turn requires her to stop rubbing the filler with glasspaper and feel the filler-surface area with her fingertip(s).

In many cases, as in example 9 above, the agent is assumed to know how to detect the condition and no explicit guidance is given. In some cases however, explicit guidance is given in the form of relevant epistemic actions and a directly perceivable condition. For example, the instructions below provide guidance in determining whether water contains salt.

(10) Change the water daily **until** all the salts have gone. To test this, hold a spoonful of the water over a flame so that the water evaporates. There should be no salts left.

One reason for giving such explicit guidance is that, as with other actions, epistemic and perceptual actions can have side effects that the agent might find undesirable. To avoid them, an alternative procedure may be specified in the instructions – e.g.

(11) Leave this glaze for a short time **until** it becomes “tacky” (a test strip on an old tile will indicate when it is ready).

“Tacky” is usually assessed through touch, but the assessment leaves fingerprints, which are undesirable on the object being repaired. So an alternative procedure is suggested, where the side effect won’t matter.

Of course, it is possible that the specified condition cannot be directly perceived and that no procedure for determining it is provided – e.g.

(12) Mix the powders a little at a time **until** the proportions look right,

The agent is then left to her own devices.

An interesting case is where the condition to be tested for is the agent’s ability to perform the next action in the sequence. While the condition may be tested several times and found not to hold, when it *is* found to hold, the next action has effectively been performed – e.g.

(13) Chip brick with chisel **until** it can be removed.

(14) After loosening stone with pick and shovel, pry it out with one 2x4, then with the other, **until** you can use one of the levers as a ramp to get stone out of hole.

A more specific form of this condition identifies both the next action α_{next} that the agent is looking to perform and the changes to the world (produced through either her current action or through an independent process – see Section 4) that will eventually enable her to perform α_{next} . For example:

(15) Standard wallpapers are removed by sponging with warm water **until** the paper is **soft enough** to scrape off.

(16) Continue along the skirting, inserting more wedges as you go, **until** the skirting is **loose enough** to pull away from the wall.

Notice that both “soft” and “loose” are vague predicates – there is no definitive test for **soft(X)** or **loose(X)**, not even tests specific to the type of X (e.g. a soft pudding vs. a soft stomach). As such, an instruction of the form

(Do) α until Y is soft/loose

is underspecified in a way that could lessen an agent's ability to perform α successfully. On the other hand, if the condition is only specified in terms of the agent's ability to perform α_{next} , she has less information from which to derive the relationship between what is currently happening to the world and when it may be relevant to try to perform α_{next} . Thus having conditions specified in terms of both change and ability can give an agent sufficient information to succeed.

There may, of course, be several ways to assess a condition, and with further experience, an agent may change which one she uses. So in the earlier brick-laying example (repeated here)

(17) Press it down **until** the mortar is about 3/8 inch thick.

an inexperienced agent may have to interrupt her pressing to measure with a ruler the amount of the mortar still remaining beneath the brick. With experience, the agent may learn to simply eyeball thickness. In creating realistic animations, we can have our agents' skills reflect any degree of experience, as long as it is clear what they are supposed to represent.

4 Determining the Agent's Intended Action

As noted earlier, instructions with "until" clauses have the general form

(Do) α **until** κ

Semantically, α must be interpretable as a Vendlerian "activity" or a *process* in Moens and Steedman's terminology [11] – that is, a temporally-extended action with no intrinsic culmination point. If α cannot be directly interpreted as a process, it must be coerced into such an interpretation. Moens and Steedman, for example, note how "for phrases" such as "for five minutes", can coerce what they term an *culminated process* – i.e., a temporally-extended action with a culmination point – into a process either through iteration of the basic action or through loss of its intrinsic culmination, as in:

(18) Play the Moonlight Sonata for 1 minute.

(19) Play the Moonlight Sonata for 1 day.

In the first case, the intrinsic culmination point is lost (one stops after a minute, not when one reaches the end of the piece), and in the second, playing the sonata must be repeated until it fills the whole day.

The first thing to note in interpreting instructions with "until" clauses, is that coercions such as the above can help to determine what the agent is supposed to be doing and what its relationship is to the condition to be assessed. In the most straightforward case, α is the process that affects the world either *cumulatively* until κ is the case

(20) Squeeze riveter handles **until** rivet stem breaks off.

or *nondeterministically* until κ is the case

(21) Try sample specks on the piece **until** you get a get a good match, wiping them away each time **until** you find the right colour.

As the condition-effecting process, α may either be a *simple process* such as in the “squeeze” example above or in

(22) Rotate the plate **until** the guide fingers touch the rod lightly

or what Moens and Steedman call an *iterated process*

(23) Strike set with fat end of hammer **until** rivet head is rounded off.

(24) Fill in low spots and strike off again **until** concrete is level with the top of the form.

On the other hand, when it is an *independent process* that affects the world either cumulatively or non-deterministically, the agent may not be responsible for doing anything other than actions needed to assess the specified condition κ .

(25) Let poultice stand **until** it dries.

(26) Stop work and wait **until** the water evaporates and the concrete stiffens slightly.

The independent process that produces the specified condition is often one that has been initiated by a previous action taken by the agent. If the process is *cumulative*, the condition to be assessed may either be its *end stage*, as in example 25, or an *intermediate stage*, as in example 26, where the process must be interrupted, lest the concrete harden completely. (I speculate that this independent process could alternatively be non-deterministic rather than cumulative, but I do not have any examples yet as evidence.)

An independent process may also be involved in producing the specified condition when the agent is herself engaged in a non-wait process – e.g.,

(27) Place the article in a plastic container and add distilled water Change the water daily **until** all the salts have gone.

(28) Heat larger pieces first with a broad flame, otherwise they may distort. Heat the joint in the centre **until** it is red hot.

The existence of an independent process can also affect what will happen if the agent stops her non-wait process – say to check whether the condition holds.

In example 27, the agent’s action of changing the water *enables* the process of drawing salt out of the article to continue. If the amount of salt in the water and on the surface of the article are in equilibrium, the process will stop on its own accord. Thus, if the agent fails to act, the specified condition “all the salts have gone” will never be achieved. The agent’s action provides, in a sense, the resources needed for the process to continue.

In example 28, on the other hand, the agent is not providing additional resources through her action but rather *maintaining* the existing situation, which in turns *enables* the heating process to continue. If the agent stops her maintenance action – e.g., to check whether the center joint is red hot – the joint will start to cool.

I noted above two forms of coercion from an activity with a culmination point to the *process* against which an “until” clause can be interpreted. I noted such coercions help to determine what the agent is meant to be doing. Here I want to suggest a third type of coercion. While I pose it as an alternative to the analysis given by Moens and Steedman in [11], it adheres to their basic event ontology and thus provides additional evidence for it. The suggestion is motivated by the following example:

(29) If solder gets runny or if iron smokes, turn off iron **until** it cools a bit.

I think it is obvious that what the agent is *meant* to do is to turn the soldering iron off (at which point it will start to cool) and then wait some amount of time until the iron is cooler and has stopped smoking. The question is what that interpretation derives from.²

Turning off an appliance is a *culmination* in Moens and Steedman's terminology, an activity that gives rise to a change in the world but that a speaker views as happening instantaneously. Moens and Steedman note that a "for" adverbial (which, like an "until" clause, requires a process) in combination with a culmination seems to denote a time period *following* the culmination. For example

(30) John left the room for a few minutes.

But they deny that such a durative interpretation is correct, suggesting that the phrase expresses *intention* rather than duration, since the following utterance would be true even if John is only out of the room for an instance:

(31) John left the room for a half hour, but returned immediately to get his umbrella.

I do not believe that the "until" clause in Example 29 has this property. Consider the related sentence

(32) John turned off the microphone until his hiccups disappeared.

The inference that the microphone stayed off for the full period until John's hiccups disappeared cannot be denied.³

(33) ??John turned off the microphone until his hiccups disappeared, but had turn it on again before they disappeared, to get the audience's attention.

I would argue then that the coercion that *seemed* to Moens and Steedman to be the case – that the process in question is a coercion of the *consequent state* that takes hold at the culmination point of "turn off" and continues until the agent intervenes – is *actually* the case. I believe that such a coercion is only possible if the culmination initiates an independent process, but this needs additional evidence to either support or deny.

There is one more point I want to make about how an agent derives the action she is meant to carry out: being told what condition to check can also convey information as to *how* she must act in order to check it. As such, perceptual conditions can function just like purpose clauses [4, 5, 6, 7] in guiding an agent to the more specific action she is intended to carry out as well as conveying what perceivable condition should lead her to stop it. Consider, for example:

²I thank Joseph Rosenzweig for his contributions to the following analysis.

³The issue of whether John turned the microphone back on after that is quite separate. The following examples should show that one's belief about what a agent is meant to do *after* a condition holds is strongly influenced by everyday expectations:

- i. Slow down your car until you are out of the school zone (at which point you can speed up).
- ii. Slow down your car until you reach Mary's house (at which point you should stop).
- iii. Slow down your car until you reach the end of the cul-de-sac (at which point you should turn around).

- (34) Have your helper move the tape side ways **until** the 4-foot mark on the tape coincides with the 5-foot mark on the rule.
- (35) To make sure that all corners are square, measure diagonals AD and BC, and move stake D **until** the diagonals are equal.

Without the “until” clause , the “move” verb phrases above are underspecified: they do not tell the agent (or her helper) what *direction* to move in. The “until” clauses, by indicating the condition to be achieved, conveys direction by implication – whatever direction will most directly lead to the condition becoming true.

5 Integrating Pragmatic and Epistemic Actions

To create a realistic animated simulation, one needs to figure out how an agent’s pragmatic and epistemic actions should be integrated. There are two interesting points about this issue:

- Since all actions require resources, the agent must determine whether pragmatic and epistemic actions can be carried out in parallel, or whether they must be interleaved.
- Even if they can be carried out in parallel, checking a condition has a cost and often undesirable side effects as well, so the agent may prefer to do it as little as possible, without preventing her pragmatic actions from coming to a successful conclusion. This means recognizing *when to start* checking for the specified condition and *how often* to do so.

The impression I get from the instructions I have looked at so far is that lexical semantics can only contribute to the solution of the first problem, in terms of what can be derived from aspectual type and aspectual coercion. For example, when a *culminated process* is coerced to a process through iteration of the basic action, the perceptual condition can be checked at the end of each iteration, as in:

- (36) Strike set with fat end of hammer **until** rivet head is rounded off.
- (37) Fill in low spots and strike off again **until** concrete is level with the top of the form.

On the other hand, I do think the instructions themselves help suggest answers to the questions of *when to start* checking for the specified condition and *how often* to do so. Here I am returning to the notions of *cumulative* effects and *non-deterministic* effects I introduced earlier. First consider a condition that results cumulatively from an on-going process. If the cumulative effect is perceivable, then based on the expected rate of the process, an agent can delay checking the condition until the point that the effect is likely to take hold. For example, in

- (38) Chip brick with chisel **until** it can be removed.

it is not worth the agent’s effort to start checking her ability to remove the brick each time she’s dislodged another chip. If the cumulative effect is not perceivable, then it is as if the condition were a non-deterministic result of the process. In the case of conditions that arise non-deterministically, then the existence of a reliable probabilistic model of the process might be incorporated into an efficient perceptual strategy.⁴

⁴This suggestion is due to Joseph Rosenzweig.

The examples so far only address the cost of checking conditions and therefore the desirability of a policy that delays them as long as possible and does them as infrequently as possible. I also want to call attention to the danger of starting to check a cumulative condition too soon, a danger that can be avoided by delaying checking:

(39) Let cement dry **until** kraft paper won't stick to either surface.

Checking too soon can result in kraft paper stuck to the surface.

6 Conclusion

I hope to have shown that, by forcing us to consider what aspects of language convey information relevant to behavior, animated simulations of realistic human agents can allow us to better understand language, and by doing so, allow us to better employ such agents for our benefit. Even though we have already shown that following instructions requires attention to more of an utterance than its main verb and argument structure, I believe we have just scratched the surface of what language can provide to agents. I hope that more researchers will now find it of interest to look more into relationships between Natural Language and behavior.

Acknowledgements

The author would like to thank Joseph Rosenzweig and Mark Steedman for comments on earlier drafts of the paper. This research is partially supported by ARO DAAL03-89-C-0031, DMSO DAAH04-94-G-0402 and ARPA DAAH04-94-G-0426.

References

- [1] Badler, N., Phillips, C. and Webber, B. *Simulating Humans: Computer Graphics, Animation, Control*. Oxford: Oxford University Press, 1993.
- [2] Badler, N., Webber, B., Becket, W., Douville, B., Geib, C., Moore, M., Pelachaud, C., Reich, B., Stone, M. Planning for Animation. To appear in N. Maginet-Thalmann and D. Thalmann (eds.), *Computer Animation*, Englewood Cliffs, NJ: Prentice Hall.
- [3] Becket, W. The *Jack* Lisp api. Technical Report MS-CIS-94-01/Graphics Lab 59, University of Pennsylvania, Philadelphia, PA 19104-6389, 1994.
- [4] Di Eugenio, B. *Understanding Natural Language Instructions: a Computational Approach to Purpose Clauses*. PhD Thesis, University of Pennsylvania, August 1993. (Technical Report MS-CIS-93-91.)
- [5] Di Eugenio, B. Action Representation and Natural Language Instructions. *Proc. 4th Int'l Conference on Principles of Knowledge Representation and Reasoning (KR '94)*, Bonn Germany, May 1994.
- [6] Di Eugenio, B. An Action Representation Formalism to Interpret Natural Language Instructions. *Computational Intelligence*, to appear 1995.

- [7] Di Eugenio, B. and Webber, B. Plan Recognition in Understanding Instructions. *Proc. 1st. Int'l Conference on Artificial Intelligence Planning Systems*, College Park MD, June 1992.
- [8] Goldman, A. *A Theory of Human Action*. New York: Prentice-Hall, 1970.
- [9] Kirsh, D. and Maglio, P. On Distinguishing Epistemic from Pragmatic Actions. *Cognitive Science* 18(4), 1994, pp. 513-549.
- [10] McGowan, J. and DuBern, R. (eds.) *Home Repair*. London: Dorling Kindersley Ltd., 1991.
- [11] Moens, M. and Steedman, M. Temporal Ontology and Temporal Reference. *Computational Linguistics*, 14(2):15-28, 1988.
- [12] Pym, D., Pryor, L. and Murphy, D. Actions as Processes: A position on planning. Working Notes, *AAAI Spring Symposium on Extending Theories of Action*. Stanford CA, March 1995, pp. 169-173.
- [13] *Reader's Digest New Complete Do-It-Yourself Manual*. Pleasantville NY: The Reader's Digest Association, 1991.
- [14] Stump, G. *The Semantic Variability of Absolute Constructions*. Dordrecht: D. Reidel, 1985.
- [15] Webber, B. and DiEugenio, B. Free Adjuncts in Natural Language Instructions. *Thirteenth International Conference on Computational Linguistics (COLING)*, Helsinki Finland, August 1990, pp. 395-400.

Visual Language Parsing: If I Had a Hammer...*

Kent Wittenburg

Bellcore, Rm. MRE 2A-347 445 South St. Morristown, NJ 07962-1910 USA
e-mail: kentw@bellcore.com

Abstract

Since the 1960s, grammatical formalisms and parsing methods developed originally for natural language strings have been extended to parse two-dimensional visual expressions such as mathematics notation and various kinds of diagrams. Part of the goal of this talk will be to summarize the highlights of historical and ongoing work in this area. Many technical issues remain. But despite all of the effort, there has been negligible impact on the real world of graphics and visual language interfaces. Why? As with all tech transfer issues, some of the reasons may be beyond a researcher's control. However, I believe that two of the contributing factors in the case of visual language parsing can and should be addressed by the research field. First, the field needs to consolidate and communicate its results. This is in fact not trivial for higher-dimensional parsing, and I will try to illustrate why. Second, researchers have to look harder for the right application domains. One of the obvious applications is the interpretation of visual language expressions constructed with GUIs. While grammatical representation and parsing may bring something to the table, the problem is viewed by the industry as solved and in fact, for many visual languages, parsing may be intractable from a theoretical point of view. I'll discuss some other application areas and my experience with them: design support, smart screen layout for electronic publishing, and hierarchical visualization of large flowgraphs.

Key words: visual languages, higher-dimensional grammars, parsing, graphics.

1 Introduction

Many members of this research community probably feel, as I do, that the theories and practice of computational linguistics might contribute to characterizing expressions in nonverbal as well as verbal media. A generalization of language technologies to encompass expressions in these other media might even lead to significant advances in human-computer communication. I come to this subject as someone who, for the past seven years, has been concerned with the generalization of grammatical representation and parsing techniques to what are commonly referred to as higher-dimensional languages. Visually-oriented examples of higher dimensional languages include mathematical notation, finite-state diagrams, flowcharts, chemistry diagrams, and electronics schematics. These examples all have the property that the syntax seems to be relatively well-behaved and "generative." We can envision, at least naively, that methods for string-based languages could be extended to account for representations of these

*Copyright ©1995 Bellcore, All Rights Reserved

visual expressions. It seems reasonable to suppose one could enumerate a finite vocabulary of symbols and a set of relations among symbols that might be used to compose higher level expressions for the syntax. The semantics, in turn, even if procedural, stands a chance of bearing a close relationship to a syntactic structure that is associated with a derivation, or parse tree.

What I mean by visual languages here is then a class of notations that might reasonably be construed as languages in the classical sense. That is, we can first characterize an infinite set of expressions using a finite discrete vocabulary together with a set of combinatory operations. We then characterize languages with grammars that can generate (or perhaps just recognize) subsets of the freely composable expressions. While such a definition certainly does not preclude languages that might incorporate temporal or 3-dimensional spatial relations, my experience has been in the two-dimensional graphical domain and that is what I will focus on here.

By the way, not all sorts of nonverbal expressions that we might want to include in multimodal communication will pass the test of visual-language-hood. For instance, simple pointing and hand-gesturing behaviors are not always usefully decomposable into a collection of discrete events with certain relations between them. It has been argued by Weimer and Ganapathy (1992), for instance, that three-dimensional hand-tracking as an input modality is best suited for continuous physical manipulations rather than for discrete symbolic expressions as we find in languages in the classical sense.

The most obvious application of visual language technologies is in the human-computer interface. Visual language interfaces are not the same as Graphical User Interfaces (GUIs) nor are they just a cover term for visualization. A visual language interface implies that composition operators used in the language used to instruct the computer are two (or more)-dimensional and that the program semantics in some way depends on these geometric or topological relationships. There is the implication that users must be able to interactively construct and/or manipulate expressions in the visual language. Visualization systems do not necessarily support this sort of interactivity; the main emphasis is on the generation side. Plain old graphical user interfaces, while not necessarily visual language interfaces themselves, may be used to construct visual language expressions. For instance, a standard graphical editor might be used to construct a graph consisting of geometric shapes for nodes and of lines for arcs. The test of a visual language interface is whether that graph is interpretable by the underlying application program. Besides mouse pointing and clicking, other input devices may of course be utilized for forming visual language expressions. For example, a data glove might be utilized to form expressions in American Sign Language for a program that was capable of interpreting such gestures.

As it turns out, there are some very challenging technical problems in producing tractable recognition and parsing algorithms for higher-dimensional languages, visual languages being one example. But the first impression on a newcomer to this area has to be the amazing proliferation of approaches to representing higher-dimensional grammars. While I can't hope to chart all this work in a short talk such as this, I can at least provide mention of some of the more visible landmarks.

2 Theoretical Computer Science and Engineering Literature

Natural generalizations of strings from a formal standpoint include the following classes of expressions:

two-(or n-) dimensional arrays

trees

graphs

Any of these basic formal constructs may be further enhanced through the addition of attributes. There are numerous proposals for rewriting systems for all these forms. Rosenfeld (1990) has written one of the few articles I've come across that attempts to synthesize results across all of them. From the engineering pattern-matching perspective, Fu (1974) is the classic reference. Although interest in array Π grammars seems to have largely died out, there was a lot of formal work in the 60s and 70s. Tree grammars have received less attention, with the notable exception of tree-adjoining grammars in computational linguistics (Joshi 1985), than the more general subject of graph grammars.

There is still an active research community in formal studies of graph grammars (see, e.g., Ehrig et al. (1991)). There was another workshop on graph grammars in the fall of 1994, which will lead to another Lecture Notes on Computer Science volume some time later this year. A now long-outdated bibliography on graph grammars (Nagl 1983) is no less than 33 pages long. Many members of the graph grammar community recognize the need to synthesize results, but it is not easy to do. There are many kinds of graphs, and even more definitions for rewrite rules for graphs. There are, however, signs of convergence on a Chomsky-style hierarchy for graph language classes. Brandenburg (1989) has shown there to be a general class of polynomial-time recognizable graph grammars characterized by having the finite Church Rosser property (confluence) and by generating connected graphs of bounded degree. This general class has come to be known as context-free graph grammars. In practice, parsing of even this restricted class of graphs may in fact not be feasible since the degree of the polynomial may be high. An approach to achieving efficient parsing in practice has been to use so-called programmed grammars, a technique for adding procedural control methods to the parser (Bunke 1982).

The basic idea at the core of higher-dimensional approaches is to enhance the classical definition of context-free string grammars by substituting other mathematical constructs for the expression class that comprises the input and output of each replacement step in a derivation.

Context-free Higher Dimensional Grammars

$G = (N, T, S, P)$

N is a set of nonterminal labels

T is a set of terminal labels disjoint from N

S , a member of N , is the start label

P is a set of productions of the form $A \rightarrow a$

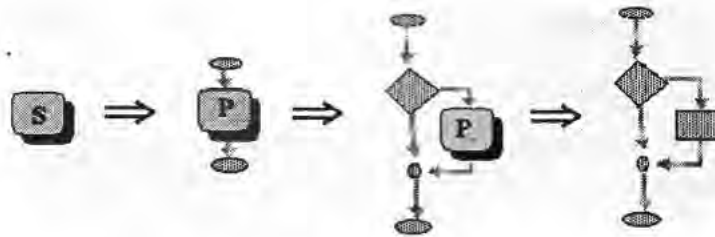


Figure 1: An example of a derivation in a node-replacing graph grammar.

where a , a replacement for A , is a composite mathematical construct such as an n -dimensional array, a tree, a graph, a set of relations...

For array grammars, a cell in an array might be replaced in a derivation step by another array, but something has to be said about how the surrounding context will be affected by such a replacement. You can't replace a single cell in the middle of a two-dimensional array with an arbitrary two-dimensional array and still have a coherent array as a result unless you somehow do some shuffling. Definitions for replacement operations for trees or graphs also are, unfortunately, not so obvious as they are for strings. Tree-adjointing grammars, well-known in the computational linguistic community, defines replacement through the operation of tree adjunction. Node-replacing graph grammars must specify how incoming and outgoing arcs of a nonterminal node will be rerouted to nodes of the replacement graph on the right-hand-side of the production. There are many variants of such replacement operations in the literature of array and graph grammars. Each definitional variant of a replacement operation is typically accompanied by a unique definition of grammar productions.

Figure 1 shows a generic example of a derivation that involves attributed node replacement in a simple flowgraph language.

3 Visual Language Literature

Paralleling the theoretical computer science literature, there has been since the 60s a body of grammar work that has focused on graphics and pictures, and even architectural designs. Applications in handwriting, mathematics, and character recognition provided one thread. Another was parsing of hand-drawn diagrams. Shaw's work on Picture Description Languages (Shaw 1969) is often cited. The basic idea there was to rewrite pictures to pictures, where a particular representation was developed that seems to have been primarily motivated by line drawings and handwriting recognition. Anderson's early work on mathematical notation (Anderson 1968) was another important milestone.

The establishment of an annual IEEE workshop on visual languages provided another avenue for work on visual grammars. There has been a somewhat disconnected series of alternative frameworks proposed including Positional Grammars (Chang 1988), (Costagliola et al. 1991); Picture Layout Grammars (Golin and Reiss 1989); Constraint Set Grammars (Helm and Marriott 1991); and Relation(al) Grammars (Crimi et al. 1991), (Wittenburg et al. 1991), (Wittenburg 1992, 1993). One influence on some recent work in this area has come from constraint logic programming, which is evident in Helm and Marriott's work.

Not all of the visual language frameworks fall into the context-free arena, where derivations with tree structures are maintained. Rekers (1994) has incorporated work from general

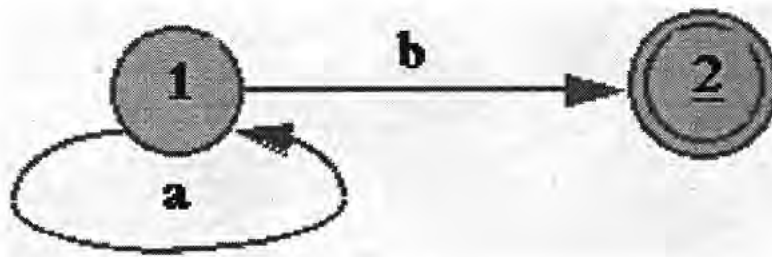


Figure 2a

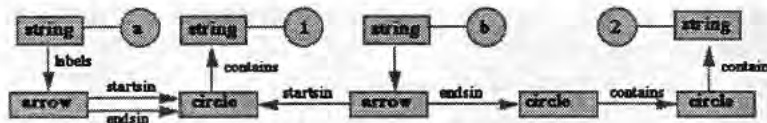


Figure 2b

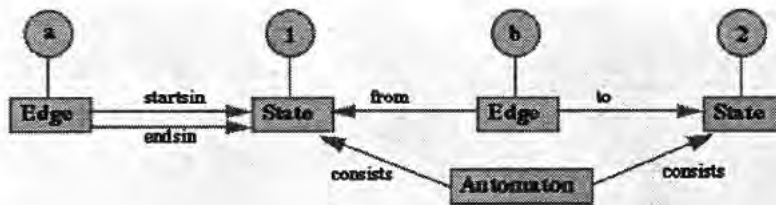


Figure 2c

Figure 2: An example of a non context-free approach to language interpretation.

graph-rewriting. Meyer (1992) has incorporated general inferencing from logic programming paradigms as has Pineda (1992). Golin and Reiss (1989), working in the attribute grammar paradigm, has suggested a mechanism that allows for some limited node sharing in derivation trees. The computational complexity of most of these approaches is unknown, although Golin (1991) has reported a polynomial bound on recognition for Picture Layout Grammars.

Figure 2, adapted from Rekers (1994), is an example of a non context-free graph-rewriting system used in visual language interpretation. Figure 2a shows an example of the input, a finite state diagram. Finite state diagrams are an interesting case since they seem to be one of the more basic examples of visual languages and yet they are not context-free. Figure 2b is the lexical representation used as input to the syntactic processor. Figure 2c is the result of a syntax analysis. All of these representations are graphs and graph rewriting systems are used to move from one level of representation to the next.

4 Impact to Date

What sort of impact on real-world visual language applications has been achieved by all this work in high-dimensional grammars? Particularly when compared with the influence of string-based grammars on computing, it is startling to note that high-dimensional grammars in

general have had such little impact. Henry Baird (1990) in a survey of industrial applications of syntactic and structural pattern recognition (SSPR) writes that ROutside of ... OCR, very few applications of SSPR have surfaced. S He attributes the relatively low acceptance to a number of factors, some of which have to do with problems particular to image recognition. For example, the complexity of image segmentation and other low-level processes tend to take center stage in real world imaging applications. Practitioners are then reluctant to turn to other technologies that may be perceived as complex and unproven. Further, many image recognition problems are less like a formal language problem in which idealized models may be articulated than they are akin to general real-world perception, which formal grammars are probably unsuited for. He also mentions the problem of fragmentation in the technical literature, along with a lack of attention to real world engineering problems such as error management and clear statements of which problems a particular approach is best suited for.

While interpreting visual language expressions in GUIs may not share the lower-level segmentation problems, the other comments hold. There are also some barriers to acceptance in using parsing for visual language interpretation in particular. One is the lack of articulation of exactly what benefits parsing technologies will bring to visual language interfaces. In discussions in newsgroups such as comp.lang.visual and at IEEE Visual Language Workshops, it is not uncommon to hear the need for parsing visual languages questioned. And if this sentiment is coming from relatively academically oriented communities, one can only suppose that industrial application groups would be at a complete loss as to why one would want to parse graphical input at all. The fact is that commercial visual language programming systems have done quite well at interpretation without using grammars or parsing. The way it is done is to develop customized event handling methods, a skill that user interface developers utilize all the time. These can be very complex systems with many unexpected interactions, but at least they are familiar.

We need to distinguish the proposal to declaratively represent visual languages with grammars from the proposal to use parsing methods in the interface. The principled alternative to parsing is to use syntax-directed editing through generation. A good example is the work of Backlund et al. (1990). The main arguments for using grammatical representation are the following:

- (1) By providing a layer of declarative representation, visual language grammars can obviate the need to build complex event handling systems anew for each variant of a visual language.
- (2) Since visual language grammars may be decoupled from parsing and generation algorithms, they may offer flexibility in processing the order of user input expressions as well as provide for optimized algorithms for particular purposes.
- (3) The abstract structure associated with a derivation tree can be used for various purposes such as information hiding through visual encapsulation, higher-level editing operations, layout, and attribute-based semantic evaluation (for translation or code generation).

In my view, the jury is still out on the tradeoffs of using parsing in visual language interfaces as opposed to syntax-driven editing methods. In some of my current work I am exploring these questions, which I hope to report on shortly. In any event, it is safe to say that visual language researchers need to pay more attention to these issues if they expect their work to be accepted by commercial software developers.

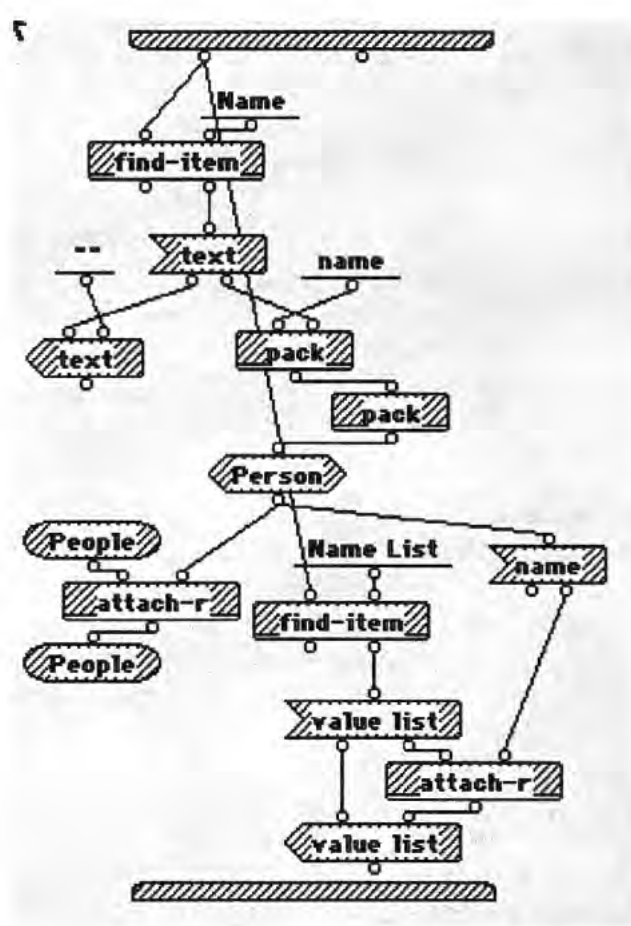


Figure 3: A visual programming language expression.

Another factor influencing acceptance of visual language grammars and parsing is what I call the disconnect between the user-level language as represented by gestures and the resulting two-dimensional graphical expressions. Consider the example of a visual language expression as evidenced in a commercial visual language programming environment, ProGraph.

Contrary to the world of textual input, in which there are standard keyboards with standard character sets, it is not the case that there is a standard way of creating a visual language expression such as that in Figure 3. In particular, interface designers will most certainly not want users to create each of the small circles (the "ports") in this expression by hand. Rather, as a user adds an arc between nodes of this graph, the small circles will be added automatically. Also, certain graphical objects in this example are nonstandard (such as the thingamajigs at the top and bottom of the figure). From a user-interface perspective, editors for creating such expressions will need to have customized palettes for their basic vocabulary along with customized gestures for adding or changing expressions. Unfortunately, the lack of such standardization makes it difficult to envision how a graphical correlate of YACC could ever achieve the same sort of acceptance that YACC has. The idealized scenario for visual language researchers is that one might use any old graphical editor to create pictures, just as one might use any old text editor to create text, and then a universal YACC-like tool could

be used to interpret the graphics. The fact is that complex graphical expressions are more difficult and time-consuming to create than text. It very much helps to have customized "short-cuts" in editors used to create them. Furthermore, there is no single graphical representation standard equivalent to the ASCII character set. Postscript might be the best candidate for such a standard, but despite the fact that a few researchers like Kahn and Saraswat (1990) have made use of it, it is not clear whether the objects and relations are at the appropriate level of abstraction for a more universal parser. There has been no concerted attempt in the visual languages community to make use of it at any rate.

5 Other Applications of Visual Language Parsing Technology

While the barriers to utilizing grammars and parsing for full interpretation in visual language interfaces do not appear to be insurmountable, there may also be other applications of the technology that are more amenable to technology transfer. Three that my collaborators and I have looked into are design assistance (Weitzman and Wittenburg 1993), multimedia document generation (Weitzman and Wittenburg 1994), and hierarchical visualization of workflows, current work at Bellcore.

5.1 Design assistance

Computers have provided access to tools for doing tasks that have traditionally only been done by design professionals. We should not expect that users of these tools be designers or have the necessary design expertise. Therefore, as design moves from traditional mediums to the electronic studio, representation of design knowledge becomes crucial in order to support a dialog between designer and machine. Weitzman and Wittenburg (1992) have suggested how higher-dimensional rules together with a bottom-up parser that can recognize fragments in the visual language of the design application can be utilized to provide assistance.

In an example page layout design scenario shown in Figure 4, the grammar rules capture a particular graphic style and embody various layout conventions such as graphic rule bars above chapter titles and section headings; default font sizes and styles; and spacings for margins.

The interaction sequence begins with the user selecting primitive elements from a palette and adding them to the working space. In this example, there are four basic categories of input of type text, number, image, and graphic rule. As things proceed, the system interactively parses the input and makes suggestions to automatically form new composite structures and install various constraints. Typically, multiple graphical constraints are used to enforce the position and size relations between elements. Constraints may also make individual changes to elements (e.g., changing their color or font specification). Relationships can be defined so that the elements involved only roughly match the desired requirements. In this way, input can be loosely sketched and the application of the rules will clean up the input.

At the beginning of the sequence in Figure 4, the user has added three basic elements: a text object, a number object, and an image object. On the right hand side of Figures 4a-c is an agenda, which is a visual indication of design assistance actions that can be exercised. These are the result of the parser recognizing expressions in the input. In Figure 4.a, it can be noted that the number item is roughly above and left aligned to the text item. A pattern has

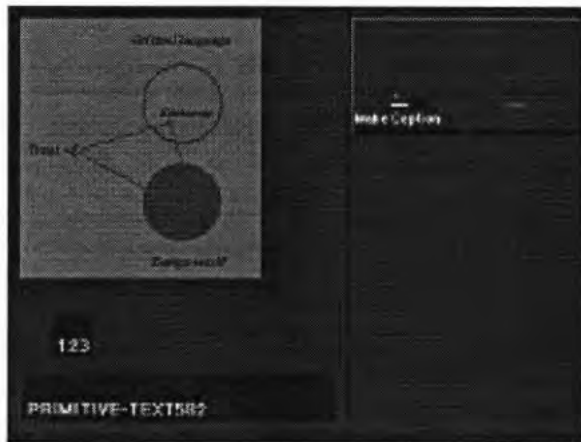


Figure 4a

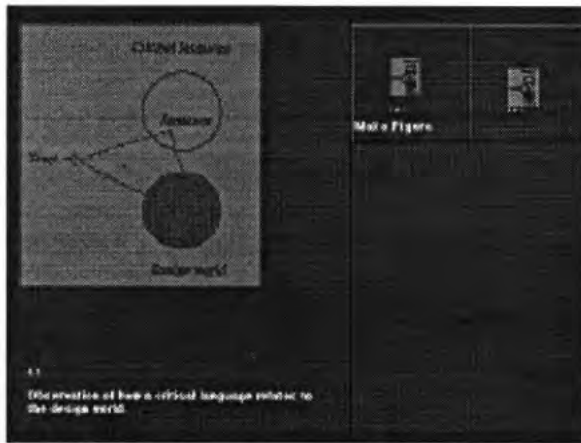


Figure 4b

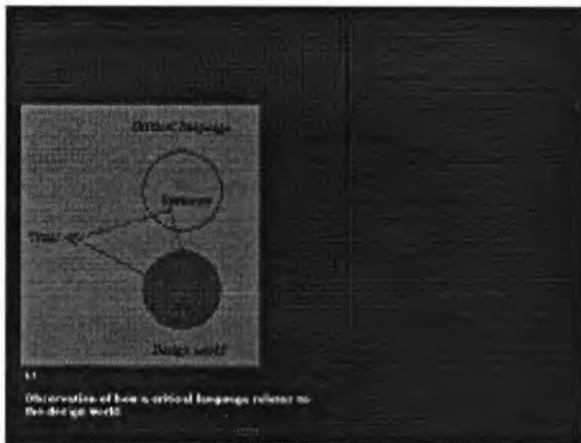


Figure 4c

Figure 4: A page layout design sequence.



Figure 5: A presentation for a large display.

been recognized that can lead to the automatic creation of a caption object. In Figure 4.b, this action has been exercised, and, in the process, the basic elements are left-aligned exactly and certain font styles are selected. Then this newly created caption object is added to the input and another fragment is recognized whose constituents are an image and a caption. The action to create a figure composite object is exercised, which results in constraints for left-alignment and certain marginal spacing arrangements.

The architecture for such design assistance is really not so different than for visual language interpretation. Obviously there is a different "semantics." Other than that, the primary difference has to do with the fact that a full interpretation, and thus a full parse, is not required, so there is a different interaction loop. This loop allows more flexibility to the user, which is appropriate in applications where suggestions and help are welcome but deviations from the norm are to be tolerated. In such an application, bottom-up parsing rather than syntax-directed editing really is a requirement.

5.2 Multimedia Document Generation

If one removes the user from the loop in the last example and assumes a derivation tree for the entire input, rather than just fragments, then one comes to an architecture that Weitzman and Wittenburg (1994) have proposed for the purpose of electronic document delivery in heterogeneous environments. Figures 5 and 6 illustrate one of the primary motivations for a system such as this.

The information is the same in the two figures but its presentation is not, motivated by differences in presentation resources available in the end-delivery environments. The design in Figure 5 is appropriate for a large, high-resolution display; the one in Figure 6 is appropriate for a small screen device, such as a hand-held digital assistant. Figure 6 displays just the first step of a complete repair procedure. As part of the presentation, the horizontal bar at the top of the page becomes an active object which controls the presentation of remaining



Figure 6: A presentation for a small display.

elements. As the user interacts with the bar, information is presented temporally that is all laid out spatially in Figure 5.

While designers could create each of these alternative designs by hand, making good use of design assistance rules, another suggestion is that the designers might be able to author a set of general realization templates (using design assistance technology there as well). These "templates" then could then be employed on demand for delivery of information in network electronic publishing environments. Our suggestion has been that the templates themselves can be coupled with grammatical rules in the form of attributes, i.e., the rule "semantics." A translation step employing attribute evaluation produces forms for creation of the relevant media objects and temporal and spatial constraints that need to be satisfied. Constraint solving is then utilized to produce the final layout. Parsing may be employed, if needed, to create the hierarchical derivation trees, but it is also possible to create the needed hierarchical structure in other ways. Figure 7 is an overview of the architecture.

Brandenburg (1994) has proposed a similar architecture for layout of hierarchically structured graphs in which he uses dynamic programming methods to control the search through alternative design solutions at each node of the derivation tree. He assumes classical attribution techniques, rather than general constraint solving, for the actual computation of values necessary for full layout specifications.

5.3 Hierarchical Visualization of Flowgraphs

The last application area I will mention is one involving visualization of large repositories of flowgraphs. As in the previous examples, a derivation tree is the starting point for providing useful services to the underlying application. The application in question involves support of representation, modeling, and redesign of work process flows in the telecommunications business. There can often be a large set of interconnected workflows that might be associated

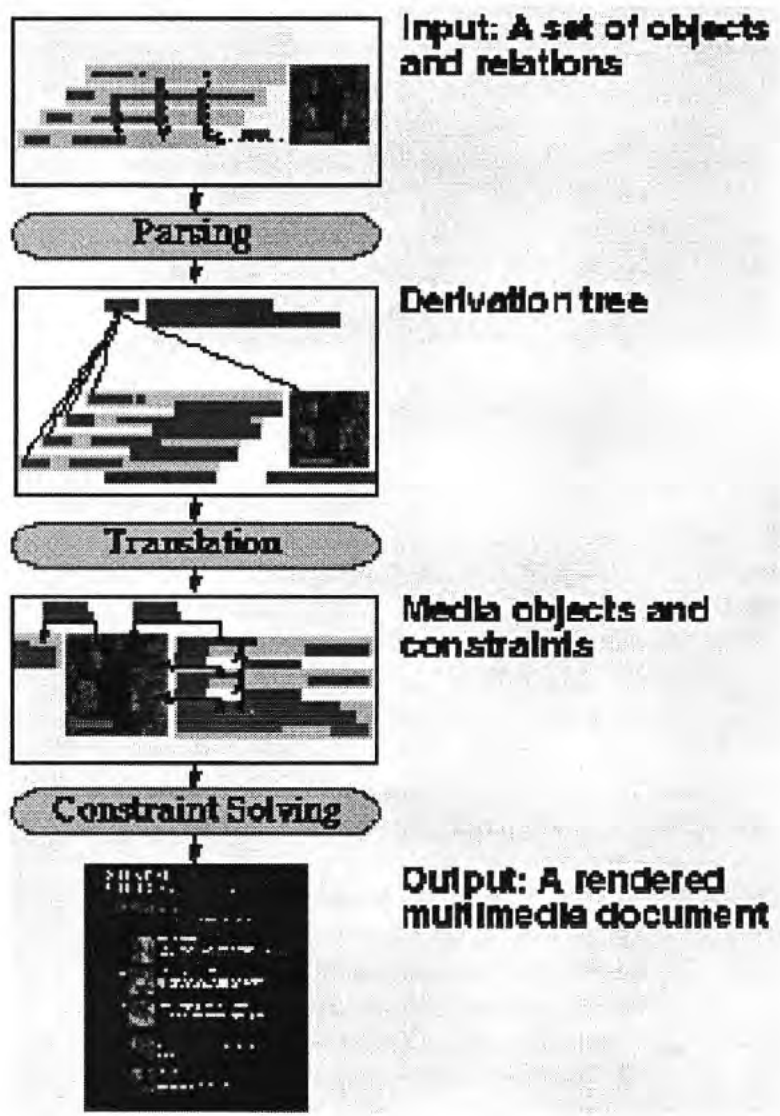


Figure 7: System architecture.

with a single work center in a telecommunications organization. Existing commercial flow-graph drawing tools standardly support the feature of hierarchically structured flowgraphs, where a single node in a graph can be expanded into another window, in which more detail is shown. However, these hierarchical structures must be assembled by hand and, once created, they are permanent. A feature that we are offering in our tool is for users to be able to dynamically select various possible subgraphs in a larger graph through interactive parsing. The subgraphs can then be collapsed or expanded to suit the visualization needs at hand. This allows the creation of views of workflow processes that can span a much larger set of the relevant business domain but that nonetheless contain an appropriate level of detail. Figure 8a shows a subgraph that has been selected through interactive parsing. In Figure 8b, the user has chosen to collapse that subgraph. Future plans include extensions that will allow the system to do more of the work in selecting appropriate designs for optimized views.

6 Conclusion

At the very least, I hope this brief survey has convinced you that lest you think otherwise, you are not the first one to have thought of the idea of using grammar technologies in visual domains. There has been no lack of theoretical work in this area. What is really needed is more emphasis on applications and an effort to consolidate results and defragment the literature. We have to somehow overcome the impulse of every red-blooded researcher to invent yet another visual grammar framework. There are exciting prospects on the horizon for bringing together work in constraint solving, rule-based inferencing, and grammar representation. As far as applications go, I hope to have brought to light some issues regarding the use of grammar technologies for visual language interpretation. If the visual language community confronts these issues head on, perhaps we can see visual language parsing as a part of our future commercial software products. Meanwhile, there are other intriguing application domains that should not be overlooked. The ones I've mentioned all involve forms of cooperative communication between human and computer. There are no doubt many avenues open for exploring the integration of multiple communication modalities in which formal grammars may play some useful role.

Acknowledgements

I'm grateful to Bernd Meyer of FernUniversitaet, Germany for conversations and his materials on consolidation and classification of visual languages; also to Franz Brandenburg of Passau University for sharing his knowledge on graph grammars. The use of Relational Grammars for improver-based design and multimedia document generation has been part of a continuing collaboration with Louis Weitzman, formerly of the MIT Media Lab and currently a consultant with Bellcore. My primary collaborator on hierarchical visualization of process flow diagrams is Cliff Behrens.

Constraints on time and space have compelled me to leave out many relevant references on the subject of visual language parsing. Readers are encouraged to follow the trails headed by the references I have managed to include. There are also various information sources available through my World Wide Web home page, including a larger bibliography. The URL is <http://community.bellcore.com/kentw/home-page.html>.

References

- Anderson, R.H. 1968. Syntax-Directed Recognition of Hand-Printed Two-Dimensional Mathematics. In M. Klerer and J. Reinfelds, eds., *Interactive Systems for Experimental Applied Mathematics*, Academic Press.
- Backlund, B., O. Hagsand, and B. Pehrson. 1990. Generation of Visual Language-oriented Design Environments. *Journal of Visual Languages and Computing* 1:333-354.
- Baird, H. 1990. Industrial Applications. In H. Bunke and A. Sanfeliu, eds. *Syntactic and Structural Pattern Recognition: Theory and Applications*. World Scientific, pp. 369-380.
- Brandenburg, F. 1989. On Polynomial Time Graph Grammars. In Goos, G., and J. Hartmanis (eds.), *STACS 88: 5th Annual Symposium on Theoretical Aspects of Computer Science*, Lecture Notes on Computer Science 294, Springer-Verlag, pp. 227-236.
- Brandenburg, F. 1994. Designing Graph Drawings by Layout Graph Grammars. In R. Tamassia and I.G. Tollis, eds., *Graph Drawing: DIMACS International Workshop*, pp. 416-427. Lecture Notes in Computer Science 894, Springer-Verlag.
- Bunke, H. 1982. Attributed Programmed Graph Grammars and their Application to Schematic Diagram Interpretation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 4:574-582.
- Chang, S.-K. 1988. The Design of a Visual Language Compiler. In *Proceedings of IEEE Workshop on Visual Languages*, pp. 84-91, October 10-12, Pittsburgh, Pennsylvania, USA.
- Costagliola, G., M. Tomita, and S.K. Chang (1991) A Generalized Parser for 2-D Languages. In *Proceedings of IEEE Workshop on Visual Languages*, pp. 98-104, October 8-11, Kobe, Japan.
- Crimi, A., A. Guercio, G. Nota, G. Pacini, G. Tortora, and M. Tucci (1991) Relation Grammars and their Application to Multi-dimensional Languages. *Journal of Visual Languages and Computing* 2:333-346.
- Ehrig, H., H.-J. Kreowski, and G. Rozenberg, eds. 1991. *Graph Grammars and their Application to Computer Science: 4th International Workshop*, Bremen, Germany, March 1990. Lecture Notes in Computer Science 542, Springer-Verlag.
- Fu, K.-S. 1974. *Syntactic Methods in Pattern Recognition*. Academic Press.
- Golin, E., and S. Reiss. 1989. The Specification of Visual Language Syntax. In *Proceedings of IEEE Workshop on Visual Languages*, pp. 105-110, October 4-6, Rome, Italy.
- Golin, E. 1991. Parsing Visual Languages with Picture Layout Grammars. *Journal of Visual Languages and Computing* 2:371-393.

Helm, R., and Marriott, K. 1991. A Declarative Specification and Semantics for Visual Languages, *Journal of Visual Languages and Computing* 2:311-331.

Joshi, A. 1985. Tree Adjoining Grammars: How Much Context-Sensitivity is Required to Provide Reasonable Structural Descriptions? In D. Dowty, L. Karttunen, and A. Zwicky, eds., *Natural Language Processing: Theoretical, Computational and Psychological Perspectives*, Cambridge University Press, pp. 206-250.

Kahn, K., and V. Saraswat. 1990. Complete Visualizations of Concurrent Programs and their Executions. In *Proceedings of IEEE Workshop on Visual Languages*, pp. 7-14, October 4-6, Skokie, Illinois, USA.

Meyer, B. 1992. Pictures Depicting Pictures: On the Specification of Visual Languages by Visual Grammars. In *Proceedings of IEEE Workshop on Visual Languages*, pp. 41-47, September 15-18, Seattle, Washington, USA.

Nagl, M. 1983. Bibliography on Graph Rewriting Systems (Graph Grammars). *Bulletin of European Association of Theoretical Computer Science (Austria)* 20:114-148.

Pineda, L. 1992. Reference, Synthesis and Constraint Satisfaction. *Eurographics* 11:C333-C344.

Rekers, J. 1994. On the Use of Graph Grammars for Defining the Syntax of Graphical Languages. In *Proceedings of Colloquium on Graph Transformation and its Application in Computer Science*, Palma de Mallorca, Spain, March 1994.

Rosenfeld, A. 1990. Array, Tree, and Graph Grammars. In H. Bunke and A. Sanfeliu (eds.), *Syntactic and Structural Pattern Recognition: Theory and Applications*, World Scientific, Singapore.

Shaw, A.C. 1969. A Formal Picture Description Scheme as a Basis for Picture Processing Systems. *Information and Control* 14:9-52.

Weimer, D., and S.K. Ganapathy. 1992. Interaction Techniques Using Hand Tracking and Speech Recognition. In M. H. Blattner and R. B. Dannenberg (eds.), *Multimedia Interface Design*, ACM Press, pp. 109-126.

Weitzman, L., and Wittenburg, K. 1993. Relational Grammars for Interactive Design. In *Proceedings of IEEE Symposium on Visual Languages*, pp. 4-11, August 24-27, Bergen, Norway.

Weitzman, L., and K. Wittenburg. 1994. Automatic Generation of Multimedia Documents Using Relational Grammars. In *Proceedings of ACM Multimedia 94*, pp. 443-451, October 15-20, San Francisco, California, USA.

Wittenburg, K., Weitzman, L. and Talley, J. 1991. Unification-Based Grammars and Tabular Parsing for Graphical Languages, *Journal of Visual Languages and Computing* 2:347-370.

Wittenburg, K. 1992. Earley-style Parsing for Relational Grammars. In Proceedings of IEEE Workshop on Visual Languages, pp.192-199, Sept. 15-18, Seattle, Wa.

Wittenburg, K. 1993. Adventures in Multidimensional Parsing: Cycles and Disorders. In Proceedings of the Third International Workshop on Parsing Technology, pp. 333-348, Tilburg, Netherlands and Durbuy, Belgium, August 1993.

Submitted Papers

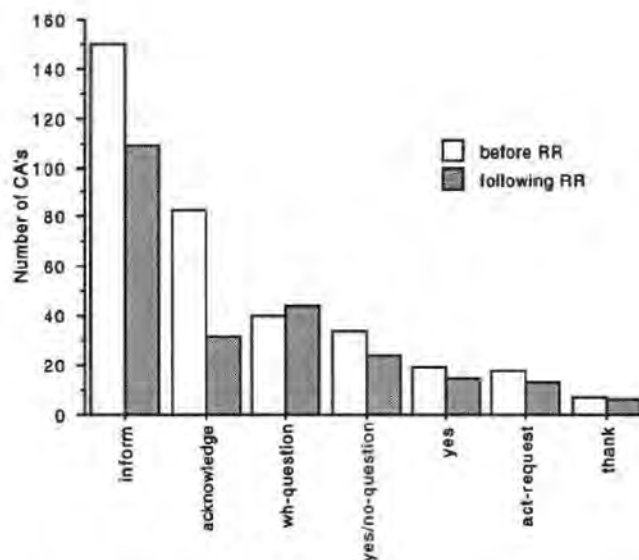


Figure 1: Number of major Communicative Acts used before and after RR

3 Results

Unless noted in the discussions below, figures show combined results for MM and telephone conditions.

3.1 Linguistic accommodations

We investigated a variety of levels at which subjects made linguistic adjustments to their post-RR utterances: discourse and syntactic structure, lexical choice and number, and disfluency and speaking rates.

3.1.1 Discourse strategies

Pre- and post-RR utterances were analyzed by hand for discourse units. The Communicative Acts (CA) in each utterance were labelled using the bilingual set of CA labels developed at ATR [5]. CA's are roughly equivalent to speech acts, and capture the illocutionary intent of a phrase or clause. Labels are assigned based on the surface structure of the utterance. The frequencies of the most common CA's are given in Figure 1. CA's which appeared rarely (once or twice in the utterances of only one or two speakers, for example) are not included in this discussion.

Reductions in frequency are observable for all CA's except WH- QUESTION. The percent of decrease for INFORM, YES/NO QUESTION, and ACT-REQUEST are roughly the same, between 27 and 29%, with that for YES slightly lower at 21%. The slight increase in the frequency of WH-QUESTION is due to a discourse strategy apparent in the data: some

subjects replaced a YES/NO QUESTION or a series of YES/NO QUESTION's with a single WH-QUESTION, as the client (C) did in example (1):

(1) C: But that says "Keage." Is "Keage" "Keitsu?" Are they the same?

WOZ: Please repeat

C: [um] I don't see "Keitsu" on the window. Where is "Keitsu?"

Note the much higher rate of decrease for ACKNOWLEDGE (Figure 1). There was a strong tendency for subjects to eliminate from their speech short acknowledging phrases such as "OK," "great," and the like (see "Structural clarification" below). However, this decrease is significant only in the telephone mode ($p < 0.05$).

Although ACKNOWLEDGE was the only CA to show significant effects of modality, there were some interesting intersubject effects for YES and YES/NO QUESTION. While the frequency of use of these CA's varied significantly by subject in the utterances *before* a RR, that variation was not significant in the *responses* to RR's. That is, although subject behavior was significantly different (with respect to these communicative acts; $p < 0.05$) in the utterances before RR's, it was much more consistent in responses to RR's².

3.1.2 Structural clarification

Utterances occurring before and after RR's were analyzed by hand for their syntactic structure and wording. In the course of the analysis, a number of distinct strategies for structurally modifying utterances became apparent. The three major strategies (in order of frequency) involved:

- eliminating short, idiomatic acknowledging structures, such as "OK," "all right," and "I see"
- eliminating clauses
- and changing lexical items (often phrasal idioms) so that their meaning was clearer.

Three secondary strategies, employed more or less equally frequently, involved:

- reducing the complexity of an utterance structure by simplifying the syntax
- reducing the complexity of a structure by eliminating adjuncts
- and amplifying a lexical item to make it more easily understood by providing a more specific or complete reference.

Example (2) illustrates the first three, most common, strategies:

(2) C: All right. I also will need to have a hotel reservation. Can you give me a hotel reservation please?

²The frequency of use of INFORM also showed a tendency to vary in this way. However, because INFORM is such an integral part of conveying information, the variation among subjects with respect to use of INFORM was not quite significant before RR's. It was much less significant, however, after RR's.

WOZ: Please repeat

C: I would like to get a hotel reservation please

In this case, the client eliminates the "all right" (an acknowledging phrase), drops the second clause, and changes the choice of lexical items in the first clause from "I will need to have" to "I would like to get," a slightly less indirect, slightly more transparent way to make the request for a hotel reservation. In example (3), the client eliminates a clause, and also amplifies "it" to "the bus ride":

(3) C: [im] [ah] How many stops is that and how long does it take?

WOZ: Please repeat

C: [um] How long is the bus ride?

In example (4), the subject simplifies the grammatical structure by changing a conjoined yes-no question (one clause of which itself contains a conjunction) into a single wh-question; in example (5), simplification has been achieved by dropping two adjunct appositive clauses:

(4) C: [ah] (is it thi³) Is it a straight walk or should I take a taxi or bus?

WOZ: Please repeat

C: (what's the) What's the best way to get there?

(5) C: [ah] (can I s) Can I make a reservation for an economy hotel, a cheap hotel, inexpensive hotel?

WOZ: Please repeat

C: [ah] I want to stay in a inexpensive hotel

The strategies described above are listed to the left of the dark vertical line in Figure 2. Subjects also employed strategies which would seem to be counter-productive to enhancing understanding. They sometimes added clauses, acknowledgement idioms, or adjuncts, made the meanings of phrases more opaque, or changed simple syntactic structures into more complicated ones. However, these strategies were employed significantly less frequently than the strategies described above. The frequencies for these less productive strategies are displayed to the right of the dark vertical line in Figure 2.

There were no significant differences in usage dependent upon communication modality.

3.1.3 Number of words

As a natural byproduct of using simpler or fewer syntactic structures, subjects reduced the number of words they used in post-RR utterances. This reduction (Figure 3) is not statistically significant. This is not surprising; subjects were constrained by the task to convey and request certain information and could not reduce their use of words beyond the threshold required to accomplish this task.

³"thi" is the transcription for "the" pronounced with a long "e" sound.

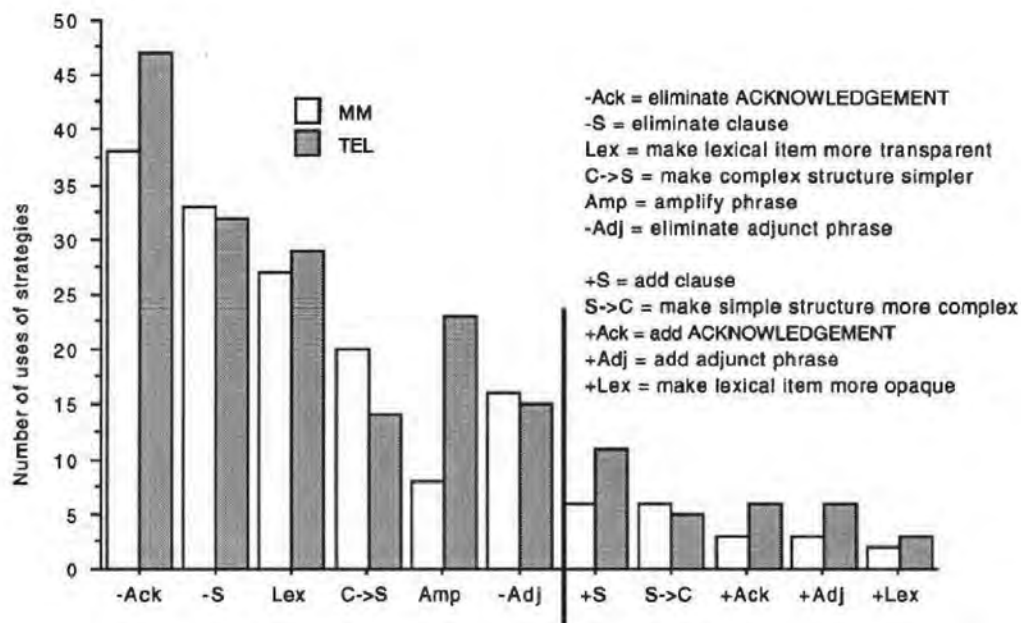


Figure 2: Frequency of use of syntactic clarification strategies

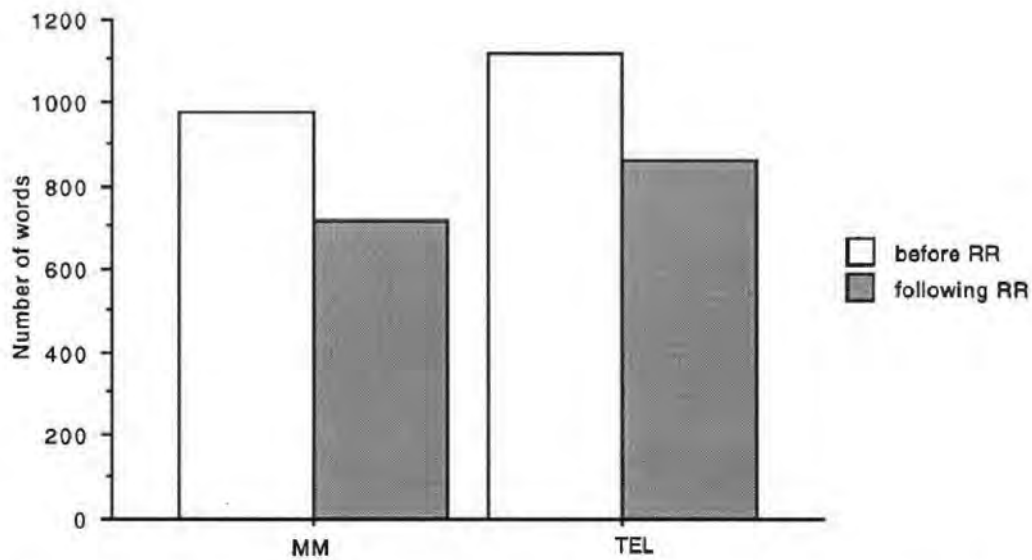


Figure 3: Number of words used before and after RR's in telephone and multimedia conditions

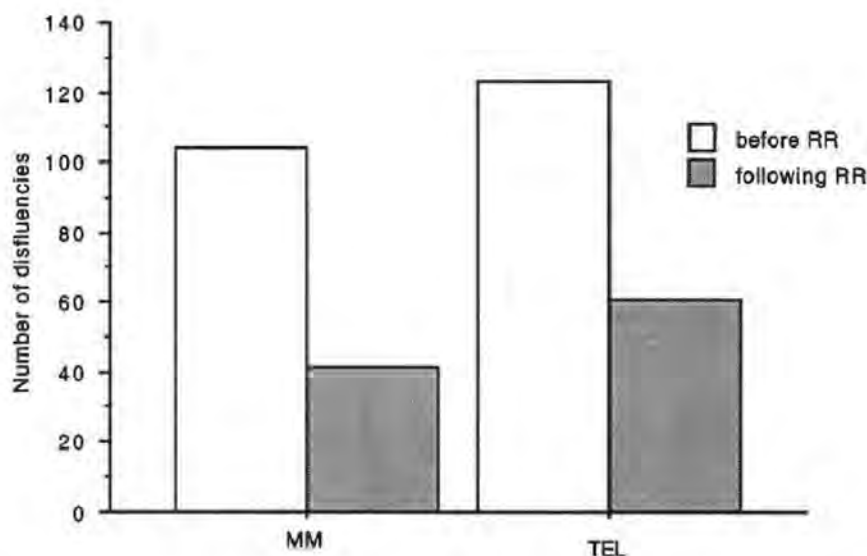


Figure 4: Number of disfluencies before and after RR in telephone and multimedia conditions

Although there were no significant differences across subjects, the same intersubject trend that was observed above for CA's is evident here. While subjects did vary significantly ($p < 0.05$) in the number of words used in pre-RR utterances, they did not vary significantly in the number of words used in post-RR utterances.

3.1.4 Lexical choice

As we conjectured, subjects showed a strong tendency to repeat the lexical items used in pre-RR utterances when constructing clarification utterances after RR's. Individual subjects repeated a minimum of 23% and a maximum of 80% of the words in their pre-RR utterances, with an average repetition rate of 50%. There were no significant differences dependent upon mode.

3.1.5 Disfluency

Disfluencies are defined as the filled pauses and false starts uttered by a speaker. Speakers significantly decreased the number of disfluencies they uttered when making clarification (post-RR) utterances (Figure 4). There were no modal effects.

There was the same intersubject effect for number of disfluencies as was observed above for CA's and number of words. While the differences in numbers of disfluencies were significant across subjects in the pre-RR cases ($p < 0.05$), those differences were not significant across subjects in the post-RR cases.

3.1.6 Speaking rate

Measurements for speaking rate were quite crude and revealed no modal differences. However, speaking rates tended to slow in the post-RR utterances, and showed the same sort of inter-subject differences as those observed above; while speakers differed significantly in speaking rate before the RR, they did not in their responses to the RR.

3.2 Media use

During the MM condition of the experiment, subjects were able to type in a text window at all times, and they could draw on a map or type in the slots of a form during any time that these graphics were displayed. In previous experiments in EMMI (involving same-language and human-interpreted situations, but not Wizard of Oz), subjects rarely availed themselves of these options [6,7]. However, we hypothesized that the increased processing demands placed on them by the "machine translation" environment would encourage subjects to increase their use of these options as they attempted to make themselves understood. The overall use of keyboard and touchscreen media in this experiment, was, in fact, much higher than that in previous experiments [6,7,8]. Here we will report on the relationship between the use of these additional media and the incidence of RR's.

One of the ten subjects did not use any media other than speech. Three other subjects used non-speech media infrequently and with no apparent relation to RR's. The non-speech media use of the remaining six subjects, discussed below, seemed to bear some relationship to RR's. Our criteria for positing such a relationship is the presence of non-speech media use in either a response to a RR or in the next contribution after a response to a RR. The client's drawing in example (6) is an example of the former case; the client's typing in example (7) is an example of the latter. (In the examples below, italics mark the speech that was simultaneous with drawing.)

(6) C: /ls/ I see, and that's thi Maiyako Hotel?

WOZ: Please repeat

C: [ah] I see thi hotel circled. Is that thi Maiyako Hotel?

WOZ: Please repeat

C: [ah] I see the circle. [ah] What is the hotel that is also circled? *This hotel*. Is this the closest hotel?

(7) C: OK, can you book me a room for three nights, starting tonight?

WOZ: Please repeat

C: OK, I need a room for three nights. Can you book?

WOZ: hai, sanpaku, shitainodesuga, yoyaku dekimasuka?

A: hai, itsukara otomarini narimasuka?

WOZ: Yes, from what day will you stay?

Client then types days of arrival and departure

3.2.1 Use of map

As in previous experiments, both client and agent drew on the map as one way to communicate location and direction. Subject drawing took a number of forms. Frequently, subjects drew a line showing direction while they described the same direction in speech. Sometimes, their line drawing followed the relevant speech. Subjects also circled their location or attempted to mark their location with a single point⁴. (For an in-depth description of media use in this experiment, see [8].)

Three subjects used map drawing in response to RR's. Two of these subjects had only a small number of RR's in the direction-giving portion of the conversation, but both accompanied their speech with drawing in a significant number of their responses to those RR's (one out of one; two out of three). The third subject clearly depended upon drawing to help clarify his utterances; in six out of eight RR responses, he used drawing along with speech. A typical example follows:

(8) C1: /ls/ OK, I'm looking at the map. It looks like

WOZ: Please repeat

C2: [ah] I see the map. [ah] It looks like Kyoto Station. Where is thi

WOZ: Please repeat

C3: I see the map. How do I get to the Conference Center?

WOZ: chizu wo mite imasu. kokusai koryu senta madeno, annai wo onegaishitainodsuga

A: maaku-san wa, ima, kyoto ekino dono atarini imasuka?

WOZ: Mr. Mark⁵, where in Kyoto Station are you?

C4: I'm at thi Kintetsu Line. *I'm putting a mark where I'm standing*

WOZ: Please repeat

C5: I'm standing at *thi mark* near the Kintetsu Line

The subject deals with the first two RR's verbally; the information he wants to convey does not allow a graphic rendering. However, when he is asked a locational question after those RR's, he responds by making a mark on the map as he speaks the italicized portion of utterance C4. That is, although it was not possible to respond visually to the first two RR's, he could and did respond appropriately using the graphic medium to the question following those RR's. When he was asked to repeat this utterance as well, he continued to use the graphic medium in his response by drawing a circle around his mark as he said "thi mark."

A fourth subject showed a very clear and quite interesting use of drawing with respect to RR's. This subject used drawing extensively from the beginning of her conversation, and kept her hand near the monitor screen for most of the direction-giving portion of the conversation. Because she drew on the map a number of times, there were three occasions on which her drawing coincided with a pre-RR utterance. In every case, she took her hand *away* from the screen and refrained from drawing during the RR response.

⁴Eight out of ten subjects also gestured toward the screen, usually pointing, but sometimes describing a line, even though they were making no contact with the screen and, thus, were making no visible mark. These gestures often followed RR's.

⁵Not his real name.

3.2.2 Use of keyboard

In previous experiments in the EMMI environment, clients rarely used the keyboard [6,7]. However, in the WOZ experiment reported here, clients much more readily typed on the keyboard to convey information. Only three subjects did not use the keyboard at all.

Two subjects typed in all hotel reservation information once they began using the keyboard, (one subject even typed in requests and short acknowledgments like "I understand," and "thank you"). As a result, they used speech very little and completely avoided generating utterances which "the machine" would be unable to understand. Thus, it is difficult to assess the relationship between their use of the keyboard and RR's. Three other subjects also used the keyboard, but with no apparent relationship to RR's.

Two subjects showed behavior, which does, however, conform to our hypothesis about media use. One typed in information after RR's on three occasions. Another behaved similarly and then avoided further RR's by typing all remaining information. Example (7) above is a typical example of the use of the keyboard in response to RR's.

3.2.3 Use of video

Finally, recall that clients and agents could also see one another's faces in a video image in one corner of their monitors. We have noted before the total lack of use of this video image in previous experiments [9], perhaps because there is no eye contact (due to the position of the video cameras). In this experiment, however, three subjects did utilize the video medium. Two clients nodded to their agents to confirm cross-language information (such as the agent's spelling of the client's name). A third subject used the video in response to RR's. He was attempting to ask the agent to type some information to him, and he had been requested twice to "please repeat." After the second RR, he held his hands up to the camera and made typing motions while he asked again to have the information typed. (At that point, the agent complied.)

4 Discussion

4.1 Linguistic variables

Linguistic adjustments to RR's can be characterized as *reduction* and *convergence*. Subjects reduced the number of virtually all CA's used. Their syntactic adjustment strategies also tended toward reduction, i.e., the elimination of structural elements ranging from clauses to adjuncts to idiomatic expressions. There was also a trend to use fewer words in post-RR utterances.

Certainly the reduction in number of words and complexity of structure means less strain on an automatic language processing system. There were other trends which would also reduce the language processing burden. Lexical adjustments away from idiomatic phrases to more literal phrases could simplify language processing. Even the tendency to amplify phrases, while sometimes adding more lexical items or creating more complex structures, resolves

problems of ambiguity of reference (as in example (3) above). The reduction in disfluency and in speaking rate also results in a more easily processed language input.

Speakers did not only *reduce* aspects of their utterances after RR's, they also *converged* toward more similar language use. The lack of significant variation among subjects' post-RR utterances for certain CA's, number of words, disfluency and speaking rate suggests that the language behavior after RR's can be more easily and more productively modelled. The high rate of repetition of lexical items post-RR represents a similar trend toward reduction of variability, or convergence toward a consistent, predictable behavior.

Modality effects on linguistic adjustments were minimal. This seems to imply that subjects' linguistic adjustments are independent of the availability and use of modality options.

4.2 Modal variables

Subjects' use of non-speech options, being difficult to analyze numerically, are consequently difficult to interpret in the same way as linguistic adjustments. Note that when we discuss linguistic factors, we are discussing adjustments made to a message within a particular medium, i.e., speech⁶. Media use, on the other hand, involves replacing one modality with another (e.g., typing instead of speaking) or supplementing one modality with another (e.g., drawing concurrent with speaking). This, then, is one of the difficulties subjects experience in using the media available: they must either switch media or coordinate the use of one medium with another.

Speakers engage in the kind of purely oral conversation they used in the telephone condition, every day of their lives. In case of a lack of understanding on the part of an interlocutor, their linguistic options are well-known and their clarification strategies are familiar if not habitual, learned from prior verbal interaction with and observation of other speakers. Thus, it is perhaps not too surprising that we should find some general trends in the linguistic approaches used by subjects for resolving a lack of understanding.

However, in the novel MM conversational environment, not only are the options themselves new, but also speakers have had no experience observing others use different communication media in clarification. So it is to be expected that speakers should show wide variation in their approaches to utilizing non-speech options.

In general, the approaches to non-speech media use that we described above seemed to be motivated by two different assumptions. Five subjects apparently assumed that using non-speech options would only make matters worse. These are the subjects who used non-speech media infrequently if at all, and the one subject who *refrained* from using them in his post-RR utterances, even though, judging from his use of them earlier in the discourse, he seemed to think that non-speech media were generally useful.

The other five subjects attempted to use MM options to help them out of their communicational difficulties. The most heavily used modality for these subjects was the typewriting modality. Notice that this is the modality closest to speaking; it involves linguistic input

⁶Of course, it would be possible to compare messages across modalities, especially for the two subjects who used extensive typing in their conversations. We could compare their oral utterances with their (usually post-RR) typewritten utterances. This, however, has not yet been done.

which is familiar to the subjects, unlike the sort of visual input used in map drawing, for which they know no "grammar" or social conventions.

5 Conclusions and Directions

This work examines spontaneous adjustments speakers make when difficulties in communication with a "machine" are encountered, and the role that the use of multimedia systems plays in such cases.

The results regarding linguistic adjustments are encouraging. Even assuming that pre-RR utterances are ignored by a language processing system, post-RR utterances represent an improvement in the quality of input for such a system. Speakers do tend to make linguistic reductions that would lessen the burden on automatic speech processing: reductions in illocutionary force units and syntactic structures requiring processing, in number of words used, in disfluency and speaking rates, and in lexical variability.

But speakers go beyond simple reduction. They also tend to converge to a more consistent language behavior after difficulties in communication (i.e., requests for repetition) are encountered. This means that partial parsing or recognition results from a pre-RR utterance will have a number of predictable relations to the following utterance and thus can be used to enhance the processing of the post-RR utterance. Our next step in working with this data will be to incorporate these relations in a statistical language model for speech recognition, exploiting these relationships to improve performance.

On the other hand, very few of these linguistic results were in any way affected by the media through which the conversation took place. An examination of media use suggests that, since users are largely unfamiliar with non-speech options for (real-time) communication, their use of these options is dependent upon their own, individual, judgments rather than upon any generalized social conventions. The wide variety of ways of using non-speech media observed in the course of the experiment do not reveal any particular recurring, consistent pattern that could be exploited in enhancing the performance of automatic language processing systems.

We suggested that the results reported here have implications for the nature of effective constraints for a system processing spontaneous speech. Speakers should be encouraged to reduce the linguistic aspects of their utterances in ways in which they are already inclined to do so: by eliminating unnecessary phrases from their syntactic structures, reducing lexical variability and disfluencies, and slowing down their speech. Instructions to speak simply, clearly and slowly would make explicit the strategies that speakers employ spontaneously when faced with a difficult communication situation.

The next step, then, is to provide some sort of constraint upon media use. This constraint could be imposed in one of two ways, either by providing explicit instructions or by encouraging pre-existing "intuitive" strategies. Recall that, in this experiment, the primary phrase used by the Wizard to indicate lack of understanding was "please repeat." For certain types of language processing breakdown, the "machine" might be given the option to request the client explicitly to "please type" or "please draw." Pre-conversation instructions which contain even more specific injunctions, say, to type *all* hotel reservation information or to draw a circle on the map to indicate location, could also be included.

Ultimately, however, we would hope that constraints on media use will parallel those on language use. That is, as more and more people become experienced in the use of multimedia systems, it will be possible to draw on their intuitive, *media-related* responses to communication difficulties just as we propose to draw on the intuitive *linguistic* responses of the subjects in this experiment. One future experiment in EMMI will involve frequent users of multimedia systems, whose experience has supplied them with some internal model for efficient and effective use of non-speech options. By studying how these users respond to RR's, it will be possible to design media systems that encourage "natural" media-related responses to communication difficulties, and to build these designs into effective language processing systems employing multimedia technology.

Acknowledgements

The authors would like to thank Tsuyoshi Morimoto and Yasuhiro Yamazaki for their continued support.

References

- [1] E. Zoltan-Ford. How to get people to say and type what computers can understand. *International Journal of Man-Machine Studies* 34. 1991.
- [2] C. Boitet. Practical speech translation systems will integrate human expertise, multimodal communication, and interactive disambiguation. Proc. MTS-IV, Kobe, Japan. 1993.
- [3] H. Blanchon, K.H. Loken-Kim, L. Fais, and T. Morimoto. A pattern-based approach for interactive clarification of natural language utterances. Proc. Information Processing Society of Japan SIG-NL Workshop, Tokyo, Japan. May 25-26, 1995.
- [4] K.H. Loken-Kim, F. Yato, K. Kurihara, L. Fais, and R. Furukawa. EMMI-ATR environment for multi-modal interactions. ATR Technical Report TR-IT-0018. Kyoto, Japan: ATR Interpreting Telecommunications Research Laboratories. 1993.
- [5] M. Seligman, L. Fais, and M. Tomokiyo. A bilingual set of Communicative Act labels for spontaneous dialogues. ATR Technical Report TR-IT-0081. Kyoto, Japan: ATR Interpreting Telecommunications Research Laboratories. 1994.
- [6] Y.D. Park, K.H. Loken-Kim, and L. Fais. An experiment for telephone versus multimedia multimodal interpretation: Methods and subjects' behavior. ATR Technical Report TR-IT-0087. Kyoto, Japan: ATR Interpreting Telecommunications Research Laboratories. 1994.
- [7] K.H. Loken-Kim, F. Yato, L. Fais, and T. Morimoto. Linguistic and paralinguistic differences of telephone-only and multi-modal dialogues. Proc. ICSLP, Yokohama, September, 1994.
- [8] Y.D. Park, K.H. Loken-Kim, L. Fais, and S. Mizunashi. Analysis of gesture behavior in a multimedia/multimodal interpreting experiment; Human vs. Wizard of Oz interpretation method. ATR Technical Report TR-IT-0091. Kyoto, Japan: ATR Interpreting Telecommu-

nications Research Laboratories. 1995.

[9] L. Fais and K.H. Loken-Kim. Effects of mode on spontaneous English speech in EMMI. ATR Technical Report TR-IT-0059. Kyoto, Japan: ATR Interpreting Telecommunications Research Laboratories. 1994.

A Cooperative Approach for Multimodal Presentation Planning

Yi Han and Ingrid Zukerman

Department of Computer Science, Monash University
Clayton, Victoria 3168, Australia
email: hanyi@bruce.cs.monash.edu.au

Abstract

A multimodal presentation planning mechanism must take into consideration the structure of the discourse and the restrictions imposed by partial plans generated in the early stages of the planning process. The latter requirement demands that the planning mechanism be able to transfer plan constraints from one level of planning to the next and to modify partial plans locally at each level. In this paper, we introduce a multi-agent planning mechanism based on the blackboard architecture that satisfies these requirements, and we describe the constraint propagation and agent negotiation processes activated by our mechanism.

Key words: multimodal presentation, presentation planning, blackboard architecture, constraint satisfaction.

1 Introduction

An essential requirement of a multimodal presentation planning system is that it be able to convey the overall structure of the discourse [Arens, Hovy and van Mulken 1993]. In addition, we postulate that such a system must consider the constraints imposed by previously generated presentation plans and it must be able to perform local plan re-organization. The constraints handled by a multimodal presentation planner result from two main sources: (1) inter-modal relations, and (2) limited resources.

Inter-modal relations – Different portions in the discourse play different roles, such as supporting, contradicting or contrasting with other portions. These relations often impose constraints on the modalities presenting the different portions of discourse. For example, if two portions are contrasted with each other, they should be presented in the same mode.

Limited resources – A modality is not eligible to present a discourse component if its specific presentation requirements exceed the available resources. Thus, alternatives may be rejected by the presentation planner owing to limited consumable resources, such as time and space. For example, two information items that are contrasted with each other must be visible on the same screen. This requirement restricts the space consumption of the presentations generated for these items.

The processes used for mode-specific presentation planning are independent of each other in the sense that individual functions and algorithms are applied to generate mode-specific

presentations. However, these processes are inter-related because they convey the relationships between different portions of the discourse. Hence, a mode-specific generator must be able to re-organize its plan in response to the actions of another generator. For instance, consider a table containing an icon entry. If the entry expands due to an increase in the width of its column, the icon generator must re-calculate the display position of the icon, so that the relative position of the icon remains unchanged.

A more demanding type of local re-organization is required when a planned presentation exceeds a limited resource or violates the overall quality requirements of the discourse. An example of the former occurs when the size of a table exceeds the space limit. In this case, the table generator may (1) remove rows which present non-essential information, as long as the communicative goal is still achieved; and/or (2) merge the information presented in several columns and select a modality capable of presenting the resulting composite information. For example, a vector may be used to convey the magnitude and direction of a force. A table with many columns illustrates a situation where the presentation violates the quality requirements of the discourse. Such a presentation is not acceptable due to the high density of the information being presented. In this case, like above, columns can be merged to reduce the density of the presentation. In both cases, the simpler type of local re-organization described in the previous paragraph must be applied to re-calculate the display positions of the individual entries.

In this paper we describe a multi-agent mechanism for planning multimodal presentations based on the blackboard architecture. Our mechanism uses a hierarchical presentation planning process to generate presentations that convey the structure of the discourse. It propagates design constraints from one level of a plan hierarchy to the next, and allows local plan modifications as long as the change does not violate the design constraints. This mechanism has been implemented in a system for the generation of multimodal presentations that convey abstract concepts in high-school Physics. The implementation is carried out in CLOS (Common Lisp Object System) and Garnet (a Toolkit for GUI design).

2 Related Research

Two types of system architecture are used for multimodal presentation planning in existing systems. The first type of architecture is the top-down structure used in COMET [Feiner and Mckeown 1990, McKeown *et al.* 1992]. COMET first determines the communicative goals and the information to be presented, and allocates a presentation modality, viz text or graphics, based on a rhetorical schema. This schema-based planning approach plans multimodal presentations during discourse planning, hence feedback from mode-specific generators is not considered. In addition, all the means of mode integration are pre-defined in COMET.

The second type of architecture is the mixed top-down and bottom-up structure used in WIP [Rist and André 1992, Wahlster *et al.* 1993] and in the system described in [Arens, Hovy and van Mulken 1993, Hovy and Arens 1993]. WIP has distinct planning processes for text and graphic presentations, and applies a 2-step process for presentation planning. Firstly, a presentation planner expands communicative goals into a hierarchy of communicative acts with a top-down method. Secondly, the text generator and graphics generator select communicative acts for realization according to their abilities using a bottom-up method. WIP's layout manager then automatically arranges layout components of different modalities into an efficient and expressive format by solving graphic constraints representing semantic and

pragmatic relationships between different discourse components [Graf 1992]. WIP is more flexible than COMET, since modalities are selected on the basis of presentation plans, and negotiations between the layout manager and the presentation planner are allowed during the planning process. However, it does not support communication between mode-specific generators. This communication becomes increasingly important when multimodal presentation systems support several different modalities and layout formats.

In the system described in [Arens, Hovy and van Mulken 1993, Hovy and Arens 1993], discourse planning and presentation planning are implemented as two reactive planning processes. However, rather than working on the same plan as done in WIP, the discourse planning process generates discourse structures, and then the presentation planning process transforms them into presentation structures by applying mode allocation rules. An advantage of this approach is that it considers the overall discourse structure of human-computer communication.

Like the system described in [Hovy and Arens 1993], our system generates presentation structures from the discourse structures determined by a discourse planner, hence the presentation structures reflect the overall structure of the discourse. However, instead of using a single presentation planner, our system uses several mode-specific presentation planning agents and a mechanism to support negotiations between these agents. As a result, the interaction between these agents is flexible, and new mode-specific processes may be easily incorporated into the system. In addition, this approach supports the consideration of planning strategies concerned with the use of consumable resources, e.g., time and screen-space.

3 The Multi-agent Architecture

Our presentation planning mechanism propagates the requirements of existing plans and allows local re-organization when constructing presentation structures. Our mechanism is based on the blackboard system architecture, where multiple knowledge sources post partial solutions on a blackboard while solving a problem [Engelmore and Morgan 1988]. However, our system uses dynamic agents instead of the static knowledge sources of the blackboard system because it needs the ability to activate an agent when a particular task is to be performed, and remove this agent when its job is finished or when an alternative agent has completed the job in a superior way.

In our implementation, there is an agent for each modality supported by the system and an agent which handles the overall discourse structure. We have selected a *hierarchical blackboard architecture*, where agents are dynamically organized into hierarchical groups during the presentation planning process on the basis of the task decomposition. That is, an agent may employ other autonomous agents to do the required subtasks. The agent who hires other agents is called the *master agent*, while agents who work for the master agent are called *server agents*. The master agent and its server agents form a group. A blackboard is bound to each group of agents to handle the communication between them.

Figure 1 illustrates a set of rhetorical devices which are part of the input to the presentation planning system. These rhetorical devices are generated by a discourse planner such as that described in [Zukerman and McConachy 1993] to convey the magnitude of a force. The first Assertion states that the magnitude of a force represents how large the force is, and the second one states that magnitude is measured in Newtons. The Instantiations illustrate the amount of force required to move some commonplace objects.

The presentation planner takes into consideration the following attributes of the informa-

| |
|---|
| Assert [magnitude(force) = how-much(force)] |
| Assert [magnitude(force) measured-in Newtons] |
| Instantiate |
| Inst ₁ [magnitude(force) measured-in Newtons] (lift apple) |
| Inst ₂ [magnitude(force) measured-in Newtons] (push box) |

Figure 1: Discourse Structure that Conveys the Magnitude of Forces

tion to be presented in order to determine the modality of a presentation: (1) the *dimension* of the information set (e.g., 1D or 2D); (2) the *dimensional focus* of the information set, which indicates which elements of the information should be presented along each dimensional axis; (3) the *importance* of the items in the information set (*t* or *nil*); and (4) the *discourse relations* between the information items (e.g., Contradiction or Comparison). These attributes are given as input to the system together with the structure of the discourse and the information to be presented. These attributes can be generated by a discourse planner, but at present they are hand-coded. Figure 2 contains a refinement of Instantiations 1 and 2 in Figure 1, where these attributes have been filled in as follows: (1) the dimension of the information set is 2D (*action* and *force-applied*); (2) the dimensional focus specifies that objects acted upon by actions are the focus of the *action* dimension, and that the magnitude of a force is the focus of the *force-applied* dimension; (3) both Instantiations are considered important for the presentation (importance is *t*); and (4) the discourse relation between the information items is Comparison. In addition to these attributes, information characteristics of the individual concepts to be presented, such as dimension, continuity (discrete or continuous) and information type (numerical, description or name), are required in order to render these concepts. These characteristics are obtained from a knowledge base where the individual concepts are stored.

One of the features of our architecture is that it does not coerce the (possibly unmotivated) selection of a single modality for presenting a given piece of information. Rather, it allows several potentially suitable presentation agents to work in parallel on the presentation of the intended information, and eventually selects a particular modality on the basis of its resource consumption and restrictions imposed by previous plans or by applying selection heuristics. In our example, the system initially determines that the Instantiations can be conveyed by means of text, a table or an image, and the Assertions can be conveyed through text. In principle, this determination can be made by applying modality selection rules which take into consideration the discourse attributes in Figure 2. However, since the focus of our research is on the system architecture, at present our modality selection process consists of a simple procedure which returns several candidate presentation modalities that have been hard-coded for different types of input. In future, the rules described in [Arens, Hovy and Vossers 1993] will be adapted to return several presentation modalities (rather than a single modality) and to take into consideration a perceiver's ability, e.g., non-textual modalities are suitable for perceivers with a low level of literacy.

Figure 3 illustrates the agent construction process for presenting the discourse in Figure 2. The presentation planning agent invokes a text agent, a table agent and an image agent to generate the Instantiations (first layer of Figure 3). If one of these presentations, say the image, requires too much space, it is eliminated. When the presentation task is decomposed further, all three agents can hire other agents to perform subtasks, e.g., the table agent hires server agents to present the entries of the table (left branch of Figure 3). In this particular

```

utterance-structure: (instantiate action force-applied)
    aspect: magnitude
    concept: force
    set-dimension: 2D
internal-relations: (comparison action force-applied)
    dimensional-focus: ((action object)
        (force-applied magnitude))

Instantiation1:
    ACTION:
    OPERATION: pick-up
        info-type: description
        dimension: 2D
        continuity: continuous
    OBJECT: apple
        info-type: name
        dimension: 0D
        continuity: discrete
    FORCE-APPLIED:
    MAGNITUDE: 0.49
        info-type: numerical
        dimension: 0D
        continuity: discrete
    UNIT: Newton
        info-type: name
        dimension: 0D
        continuity: discrete

importance: t

Instantiation2:
    ACTION:
    OPERATION: push
        info-type: description
        dimension: 2D
        continuity: continuous
    OBJECT: box
        info-type: name
        dimension: 0D
        continuity: discrete
    FORCE-APPLIED:
    MAGNITUDE: 98
        info-type: numerical
        dimension: 0D
        continuity: discrete
    UNIT: Newton
        info-type: name
        dimension: 0D
        continuity: discrete

importance: t

```

Figure 2: Refinement of the Sample Instantiations

example, the agent group headed by the table agent includes the number agent (to present the magnitude of a force), the icon agent (to present the objects in the actions) and the text agent (to present the table headings and also the objects in the actions). An instance of an agent is created for each subtask. Hence, there are two instances of the icon agent and the text agent, one for presenting an apple and one for presenting a box, and there are two additional instances of the text agent for presenting the table headings. Thus, the icon agent and the text agent compete for the presentation of the apple and the box. However, they collaborate on the presentation of the entire table, since the icon agent is working on the presentation of two entries and the text agent is working on the headings.

4 Blackboard Events and Communication Primitives

Agents share partial presentation plans on a blackboard and communicate with each other through blackboard events. As stated above, a local blackboard is bound to each agent group formed during the task decomposition. Agents within a group read from the local blackboard plan requirements propagated from the previous level and partial plans generated by other agents in the same group, and then generate their own partial plans which satisfy these requirements.

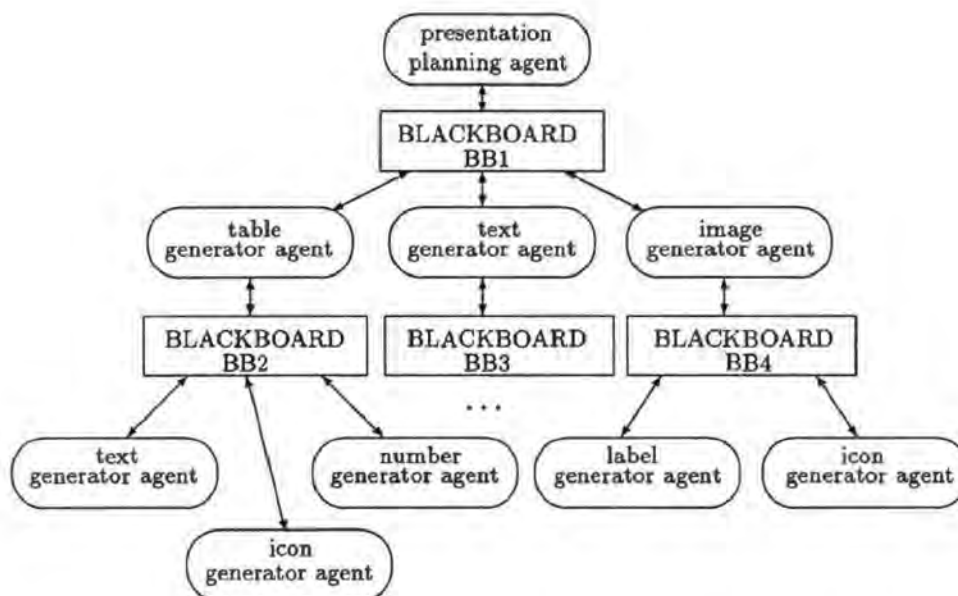


Figure 3: Agent Construction in a Multi-agent Planning Mechanism

4.1 Blackboard Events

All blackboard events have one sender agent and either one receiver agent or a group of receiver agents. We identify three types of events: *normal event*, *urgent event* and *announcement*. Normal events are messages sent from one agent to another. They are collected in the event queue maintained by local blackboards. In contrast, announcements are messages broadcast by one agent to all the agents in its group. They are forwarded immediately (without staying in the event queue) to all the agents in the group regardless of whether these agents are waiting for an event or working on a plan. Urgent events transfer important messages from one agent to another. They are forwarded to the receiver in the same way as announcements.

A message carried by a normal event may be either a request or a response to a request. An agent picks up a normal event from an event queue, and its event handlers determine its reaction to the event. An agent is able to send different requests to different agents and check their responses with respect to each request. For instance, if the table agent in the above example wishes to ask an icon agent to reduce the size of the icon it generated, the table agent will generate an *asking-event* which contains this request (Figure 4). When the icon agent picks up this event, its event handler will try to reduce the icon's size in its presentation plan. If this modification fails due to the absence of smaller icons, the icon agent will send a *rejection-event* to the table agent on the same request. Otherwise, it will send an *OK-event*.

Announcements and urgent events carry messages which require an agent's immediate attention. They interrupt the process being carried out by an agent, and force the agent to handle these events. An example of an announcement is *time-up*, which indicates that a period of real time has elapsed. When a time-up announcement is sent, all agents stop planning to handle this announcement, which requires them to display the best presentation plan generated so far. Time-up announcements are generated by an alarm process which is set up for a particular amount of time by the system at the beginning of the presentation planning process. An example of an urgent event is *cancel-request*. It indicates that the

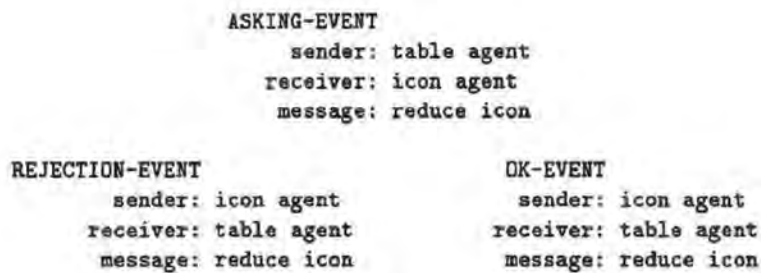


Figure 4: Events Related to a Request

master agent is no longer interested in the display being generated by the receiving agents, and that they should abort their presentation planning processes.

4.2 Communication Primitives

The system provides agents with a communication primitive called *get-normal-event* which returns a normal event sent to this agent. If such an event is not detected, the agent's process is skipped. Two planning strategies are implemented on top of this primitive.

- **Wait-all-responses.** This strategy is used by an agent if its planning process cannot proceed unless all the requests sent out by this agent are satisfied. This strategy is implemented by calling *get-normal-event* for the current process performed by the agent and an event handler to handle the event detected by *get-normal-event*. When this event has been processed, the agent waits for the next normal event. This process terminates when the agent receives a *rejection-event* response to one of its requests or an *OK-event* response to all of its requests.
- **Wait-any-response.** This strategy is used if an agent requires one of its requests to be satisfied in order to carry on with its planning process. The implementation of this strategy is similar to that of *wait-all-responses*, however, this process terminates when the agent receives an *OK-event* response to any of its requests or a *rejection-event* response to all of its requests.

Communication primitives are also provided for each type of urgent event and announcement. These primitives are called automatically after an urgent event or an announcement is created. The primitives forward an event to its receiver, select an event handler for the receiver according to the event type (e.g., *time-up*), and activate this event handler prior to the agent's normal activation when the agent is scheduled to be run. As a result, the agent is forced to interrupt its normal process and handle the received event.

4.3 System Concurrency

Agents work concurrently in our system. They may work on one partial plan, like the table agent and its server agents, or on competing plans, like the table agent and the image agent (Figure 3). As a result of concurrency the system must handle blackboard access by different agents, unexpected termination of an agent, and cancellation of requests.

Blackboard Access. Since a group of agents share the information on a local blackboard, more than one agent may try to write on the blackboard at the same time. To solve this

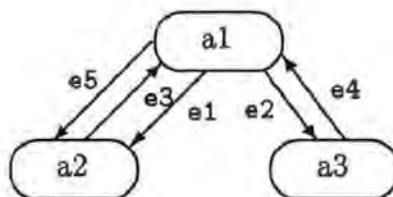


Figure 5: Example of Message Passing

| | |
|---------------------------------|---------------------------------|
| ASKING-EVENT e1 | ASKING-EVENT e2 |
| sender: a1 | sender: a1 |
| receiver: a2 | receiver: a3 |
| message: present instantiations | message: present instantiations |
| ASKING-EVENT e3 | OK-EVENT e4 |
| sender: a2 | sender: a3 |
| receiver: a1 | receiver: a1 |
| message: more screen space | message: present instantiations |
| REMOVE-AGENT-EVENT e5 | |
| sender: a1 | |
| receiver: a2 | |
| message: terminate a2 | |

Figure 6: Events Causing Termination

problem, the system provides a lock on each blackboard. An agent must acquire the lock before it writes to a blackboard, and it releases the lock when it has finished writing. If a blackboard is locked when an agent is trying to write on it, the agent must wait until the lock is released.

Unexpected Termination. An agent may be terminated by its master agent before it has completed its planning process. This happens, for example, when another agent completes a competing presentation plan before the agent in question. When an agent is terminated, the system recursively terminates all the server agents created by this agent, and clears any messages from this agent to its master agent. Figure 5 illustrates the message passing sequence in such a situation, where *a1* is the presentation planning agent, *a2* is the table agent, and *a3* is the image agent. The events appear in Figure 6. The presentation planning agent asks the table agent and the image agent to present two Instantiations (events *e1* and *e2* respectively). It then receives a request for more screen space from the table agent (event *e3*), but this demand exceeds the available screen space. As a result, *a1* decides to display the information by means of an image and to terminate the table agent (*e5*). As an urgent event, *e5* interrupts the planning process of the table agent. The event handler of *remove-agent-event* for this agent then clears the messages sent to its master agent, and sends a *remove-agent-event* to each server agent before it terminates itself.

Request Cancellation. A request sent to an agent may be canceled before the agent completes the process that addresses this request. In order to handle a cancellation, the agent needs to recover the plan that was current before the request, and clean up all the messages it sent out while processing the request. For instance, consider a situation where the table agent wants to enlarge a table of two rows and two columns which contains an icon in each entry of

| | |
|---------------------------------|------------------------|
| ASKING-EVENT e1 | ASKING-EVENT e2 |
| sender: a1 | sender: a1 |
| receiver: a2 | receiver: a3 |
| message: enlarge IconA | message: enlarge IconB |
| | |
| DK-EVENT e3 | REJECTION-EVENT e4 |
| sender: a2 | sender: a3 |
| receiver: a1 | receiver: a1 |
| message: enlarge IconA | message: enlarge IconB |
| | |
| CANCEL-REQUEST-EVENT e5 | |
| sender: a1 | |
| receiver: a2 | |
| message: cancel "enlarge IconA" | |

Figure 7: Events Causing a Request Cancellation

the first column and text in the entries of the second column. This may be done by enlarging the icons in the first column, and/or enlarging the font of the text in the second column. The message passing sequence for the first choice is illustrated in Figure 5, where a1 is the table agent, and a2 and a3 are two icon agents presenting the icons in the first column. The events appear in Figure 7. The table agent asks the two icon agents to enlarge their presentations (event e1 and event e2). This request is accepted by a2 (event e3), but rejected by a3 (event e4). Because of the rejection from a3, the choice is dropped even though a2 has no objection to it. The table agent then creates a cancel-request-event (e5) and proceeds to consider the second choice. Event e5 interrupts a2 and causes a2 to abort its plan and recover its initial status.

5 Plan Representation

A presentation structure generated by the system is composed of *segments* and *segment containers* distributed hierarchically on local blackboards. A segment defines a mode-specific display which presents atomic information, i.e., information which is not decomposable. It determines parameters such as the font, color and position of the display. A segment container includes a list of elements which in turn can be either segments or segment containers. A segment container describes the display arrangement of a list of segments as required by the discourse relations between the discourse components that yield these segments.

Figure 8 illustrates a segment container which stores the parameters defined by the table agent in our example. These parameters are: the content to be conveyed in each entry (stored in *segment-list*), the number of rows and columns, the height of the different rows and width of the different columns, and the alignment of each entry in each row and column. In this segment container, the table agent sets up two columns since there are two elements in the dimensional focus (Figure 2). In addition, the optional slot corresponding to the column heading has been filled by the table agent. The table agent then asks the icon agent, the number agent and the text agent to generate entry segments and display them in the positions specified in the *row-alignment* and *column-alignment* slots. These server agents share on the local blackboard information generated by their master agent. They use the values of the entries *columns*, *rows*, *column-width* and *row-height* to calculate the relative

```

modality: table
  columns: 2
  column-width: (70 60)
  column-separator: solid-line
  column-alignment: (center center)
  column-heading: ((action object)
                  (force-applied magnitude))
  rows: 2
  row-height: (70 70) ;pixels
  row-separator: solid-line
  row-alignment: (center center)
  row-heading: nil
  segment-list: ((object (action inst1)) (magnitude (force-applied inst1))
                (object (action inst2)) (magnitude (force-applied inst2)))
  space-estimation: (130 180) ; pixels x pixels
  time-estimation: 4 ; time-unit
modality-constraints: ((same-mode-same-column) (fixed-mode heading text)
                      (banned-mode entry (table image bar-chart line-chart)))
space-constraints: ((same-column-same-width) (same-row-same-height)
                   (keep-minimum-entry-margin) (keep-maximum-entry-margin))

```

Figure 8: A Segment Container Generated by the Table Agent

positions of the entry segments they generate. The display position of the table is calculated from information provided by the presentation planning agent (who selected a table as a presentation mode). Figure 9 shows a presentation generated by the system for this example.

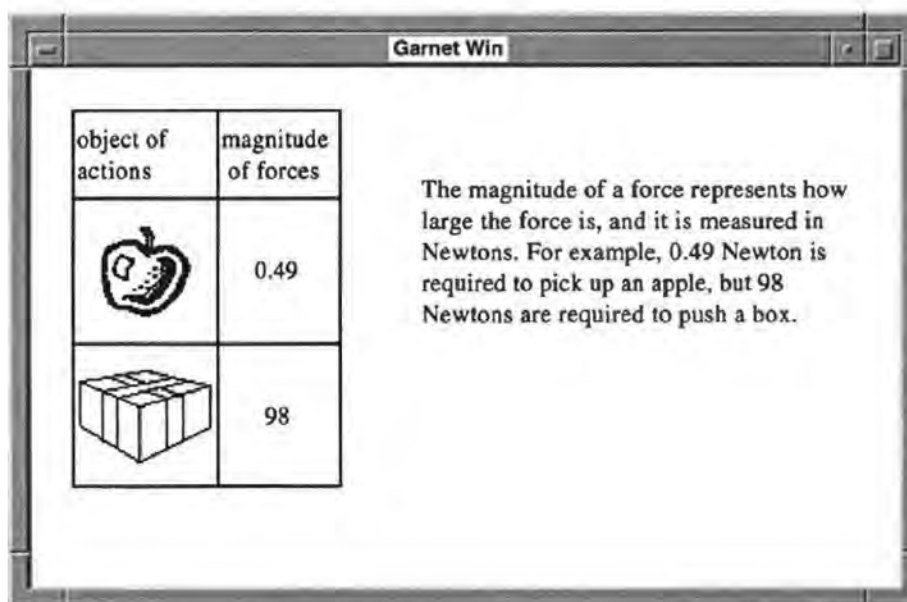


Figure 9: Sample Multimodal Presentation

6 Constraint Propagation and Agent Negotiation

The slots in a segment or a segment container generated by mode-specific agents contain variables, and the relations between mode-specific presentation plans are represented as constraints over slots which belong to individual segments or segment containers. Each agent in our system is responsible for the instantiation of a subset of variables, i.e., variables associated with the slots in its mode-specific plan. A solution, i.e., a multimodal presentation plan, is

found when all the variables are instantiated and all the constraints are satisfied.

We consider two types of constraints: (1) *Modality Consistency* – which restricts the modalities for presenting certain components; and (2) *Space Consistency* – which restricts the space that component segments can occupy. Constraints have two types of strength: *required* and *preferred*. The required constraints must be satisfied by all segments and segment containers. The preferred constraints may remain unsatisfied, but the system endeavours to satisfy as many preferred constraints as possible.

Space constraints are numeric, while modality constraints are non-numeric. The required modality constraints generated by a table agent demand that the same modality be used for presenting all the entries in a column and that table headings be textual (Figure 8). Further, they provide a list of modalities that cannot be used for presenting the entries in the table. The required space constraints generated by a table agent demand that entries in the same row have the same height, entries in the same column have the same width, and that an entry segment be displayed in a position that ensures a minimum margin from the borders of its entry (Figure 8). Related preferred constraints restrict the maximum width of the margins surrounding the presentation segments in table entries, so that a segment is not too small for its entry. Space estimations from the server agents are used to evaluate whether these constraints are likely to be satisfied. Table 1 illustrates the constraints which pertain to the width of the first column in Figure 9. We refer to the entries in this column as **entry₁** and **entry₂**, and to the segments in these entries as **segment₁** and **segment₂** respectively.

Constraints are created by a master agent when it generates its segment container. All the constraints are stored in the local blackboard, so that the information may be shared by the group of server agents. Therefore, when the master agent assigns values to its segment container, the server agents in its agent group will know the requirements placed on their partial plans. A server agent can then add its own constraints if it is the master agent of another group of agents. Hence, during the planning process, requirements of an existing plan are transferred to server agents by means of constraint propagation. These constraints ensure that each component segment satisfies the requirements of the overall discourse. For instance, if the presentation planning agent in Figure 3 wants the table to be displayed to the left of the text in the same window, it will create constraints on the width and height of the table and the text in relation to the window size. These constraints are shown in Table 2. The constraints *cn₇* and *cn₉* are then propagated to the table agent, and hence restrict the expansion of the table.

Since constraints are distributed in the plan hierarchy, the constraint satisfaction problem is considered a Distributed Constraint Satisfaction Problem, where a solution is found through agent cooperation. At this stage, constraint satisfaction of space constraints is implemented. However, constraint satisfaction of modality constraints is not implemented yet. Hence, modality selection in the system is hard-coded rather than being conducted through constraint satisfaction.

6.1 Constraint Propagation

In order for agents to be able to satisfy distributive constraints cooperatively, each agent in a group must (1) have a collection of all the constraints pertaining to its variables; (2) be able to read the values of another agent's variables, but be unable to modify the values of these variables; and (3) be able to inform another agent of its expectations regarding the values that can be assigned to the variables of this other agent, and also to inform this

| Name | Constraint | Strength |
|------------------------|--|-----------|
| <i>cn</i> ₁ | $\text{width}(\text{column}_1) = \text{width}(\text{entry}_1)$ | required |
| <i>cn</i> ₂ | $\text{width}(\text{column}_1) = \text{width}(\text{entry}_2)$ | required |
| <i>cn</i> ₃ | $\text{width}(\text{entry}_1) \geq \text{width}(\text{segment}_1) + \text{margin}(\text{minimum})$ | required |
| <i>cn</i> ₄ | $\text{width}(\text{entry}_2) \geq \text{width}(\text{segment}_2) + \text{margin}(\text{minimum})$ | required |
| <i>cn</i> ₅ | $\text{width}(\text{entry}_1) \leq \text{width}(\text{segment}_1) + \text{margin}(\text{maximum})$ | preferred |
| <i>cn</i> ₆ | $\text{width}(\text{entry}_2) \leq \text{width}(\text{segment}_2) + \text{margin}(\text{maximum})$ | preferred |

Table 1: Sample Constraints of a Column

| Name | Constraint | Strength |
|------------------------|---|----------|
| <i>cn</i> ₇ | $\text{height}(\text{window}) \geq \text{height}(\text{table}) + \text{margin}(\text{minimum})$ | required |
| <i>cn</i> ₈ | $\text{height}(\text{window}) \geq \text{height}(\text{text}) + \text{margin}(\text{minimum})$ | required |
| <i>cn</i> ₉ | $\text{width}(\text{window}) \geq \text{width}(\text{table}) + \text{width}(\text{text}) + \text{margin}(\text{minimum})$ | required |

Table 2: Constraints of Table Size and Text Size

other agent whether it considers the values assigned to these variables satisfactory. In our system, these requirements are satisfied because agents propagate constraints from one level in a plan hierarchy to the next through channels. A channel contains a set of constraints used to calculate the variables that pertain to an agent's plan from the variables of another agent's plan. Thus, an agent can collect other agents' expectations of the values assigned to its variables, and determine whether the values assigned to the variables of other agents satisfy its constraints.

To illustrate the constraint propagation process, let us consider the constraints concerned with the width of the first column in Figure 9, i.e., *cn*₁-*cn*₆ in Table 1. Among these constraints, *cn*₁ and *cn*₃ form a channel since the right hand side of *cn*₁ contains the variable on the left hand side of *cn*₃. The table agent and the icon agent can negotiate over $\text{width}(\text{column}_1)$ and $\text{width}(\text{segment}_1)$ via this channel. Other channels can be formed from *cn*₁ and *cn*₅, *cn*₂ and *cn*₄, and *cn*₂ and *cn*₆. Channels are updated by local blackboards (BB2 of Figure 3 in this case) after a constraint is added to or removed from partial plans. Hence, an agent is able to keep track of all the constraints placed on its variables, and modify its presentation plan accordingly. For example, the table agent can set $\text{width}(\text{column}_1)$ to a particular value, and ask the icon agent to reduce or enlarge the size of segment_1 to fit in the column. The icon agent then calculates $\text{width}(\text{segment}_1)$ from the channels formed by *cn*₁-*cn*₃ and *cn*₁-*cn*₅. It in turn can reject this request and insist that the current size is the only size it can provide for the segment. To this effect, the icon agent sends a blackboard event to activate the table agent, and uses these two channels to transmit to the table agent a request for a new column width to fit segment_1 . If the table agent approves the request and sets a new value for $\text{width}(\text{column}_1)$, this value is transferred via the channels formed by *cn*₂-*cn*₄ and *cn*₂-*cn*₆ to the agent which is presenting segment_2 in the same column. A consequence of this mechanism is that an agent can modify indirectly a partial plan of another agent.

6.2 Agent Negotiation

During presentation planning agents initially select values for the variable slots of their partial plans so that all the required constraints in their blackboard are satisfied. If time allows,

| Name | Constraint | Strength |
|-----------|--|-----------|
| cn_{10} | $width(column_1) = width(entry_3)$ | required |
| cn_{11} | $width(entry_3) \geq width(segment_3) + margin(minimum)$ | required |
| cn_{12} | $width(entry_3) \leq width(segment_3) + margin(maximum)$ | preferred |

Table 3: Additional Constraints for a Column

agents look for plans that satisfy as many preferred constraints as possible.

As an example, the table agent starts from an initial plan in which all the entry segments are generated from default slot values, i.e., text and numbers are in default font, and icons are in default size as they are created. In the initial plan, the table agent selects values for the column width of each column and the row height of each row to fit the biggest segment. If there are m entries in $column_n$ and one segment in each entry, $width(column_n)$ is calculated from constraints such as those in Table 1 as follows:

$$width(column_n) = margin(minimum) + \max\{width(segment_1), \dots, width(segment_m)\}$$

This initial plan satisfies all the required constraints and can be displayed immediately if necessary. However, some preferred constraints, such as cn_5 and cn_6 , may remain unsatisfied, leading to a presentation that many users would find unacceptable. An example of such a display is a table that has a tiny icon inside a large entry. Such a table would have been initially generated if there was one large default segment in the column containing this icon. To improve a plan, the table agent modifies the column width for each column and the row height for each row so that the entry cells fit most segments. As a result, more preferred constraints are satisfied, but some required constraints may be violated. To address this problem, the master agent sends requests for modifications to the server agents in charge of the segments that are now in violation of the required constraints.

For instance, consider a column denoted $column_i$ with three entries, and assume that $margin(minimum) = 6$, $margin(maximum) = 22$, $width(segment_1) = 64$, $width(segment_2) = 48$ and $width(segment_3) = 32$. The constraints corresponding to the first two entries are shown in Table 1, and those corresponding to the third entry are shown in Table 3. The initial value for $width(column_i)$ is 70. In this case, constraints cn_1 - cn_6 , cn_{10} - cn_{11} are satisfied. If the table agent had set $width(column_i)$ to 54, then an additional constraint, cn_{12} , would be satisfied. However, in this case, the required constraint cn_3 would be violated. In an attempt to satisfy the required constraint, the table agent sends a request to the server agent which is presenting $segment_1$, asking the server agent to reduce the size of this segment. If the table agent receives an *OK-event* response to this request, the plan is improved since all the required constraints are satisfied, as well as an additional preferred constraint. If the server agent is unable to satisfy the request, negotiation between this agent and the table agent is possible by means of blackboard events.

This procedure does not always produce a better plan, since it may result in the violation of previously satisfied constraints. For instance, in the case of a table, in addition to the constraints which pertain to the width of its columns, there are similar constraints which affect the height of its rows. If the table agent requests the text agent to reduce the width of a piece of text, and the text agent is able to comply, this may cause either an increase or a reduction in the height of the text. An increase in height takes place when we move words from one line to the next, while a reduction takes place if a smaller font is used. Because of

this, when an agent reduces a segment to satisfy a preferred constraint on `width(segmenti)`, a constraint on `height(segmenti)` may be violated. Such a situation may also be encountered when the table agent asks an icon agent to enlarge an icon. Since both the width and the height of an icon are increased when the icon is enlarged, constraints pertaining to the height of this icon may be violated. If a required constraint is violated, then the agents engage in a negotiation process where the icon agent asks the table agent to increase the height of the row which contains the icon. If an *OK-event* is received in regard to this request, the icon agent creates an *OK-event* to respond to the previous request sent by the table agent. Otherwise, the previous request is rejected. If a preferred constraint is violated, the process of modifying other entries to satisfy more preferred constraints continues. Upon completion of these modifications, the table agent evaluates the resulting table plan in terms of the number of preferred constraints that are satisfied. The new plan replaces the previous plan if it satisfies more preferred constraints. This process continues until it is time to display the table.

As the negotiation over a variable may introduce a new negotiation process regarding another variable, the master agent must sort out the order of variables for constraint satisfaction in order to avoid endless negotiations with its server agents. For example, segments in the same column of a table are generated by the same type of agent. This is required by the constraint which demands that the same modality be used for all the entries in a column. Hence, the table agent adjusts the width of each column before the height of each row, because segments of the same modality are more likely to be of uniform size. When the table agent is trying to reduce the width of a column, requests from its server agents in regard to modifying the height of a row will be accepted if the constraints placed on the height of the table are satisfied. In contrast, when the table agent is trying to reduce the height of a row, a request demanding an increase of the width of the column in question will be refused.

7 Conclusion

Multimodal presentation planning must take into account both the overall discourse structure of the communication process and the requirements that existing plans place on the plan refinement process. The hierarchical presentation planning process used in our multi-agent planning architecture satisfies the former requirement, and the constraint propagation and negotiation processes satisfy the latter requirement. A prototype system which demonstrates these ideas has been implemented, and an extension of this system to incorporate new modalities is currently being implemented.

Proposals for future research concern a number of issues. The modality selection in the current system is hard-coded, and hence not flexible. A mechanism for modality selection needs to be developed. An improved user model is required for the system to be able to describe precisely the interests and abilities of perceivers. Although the current system does not plan multimodal presentations in reaction to perceivers' behaviour, it is possible in the future to extend this approach to the planning of multimodal interaction. Interactive objects can be generated in display windows with the Garnet Toolkit, however, new agents will have to be designed for interactive objects so that requests from users can be translated into events, and sent to the agents concerned with these events. New event handlers and planning strategies will have to be implemented for each mode-specific agent in reaction to the events generated by these interactive objects.

Acknowledgment

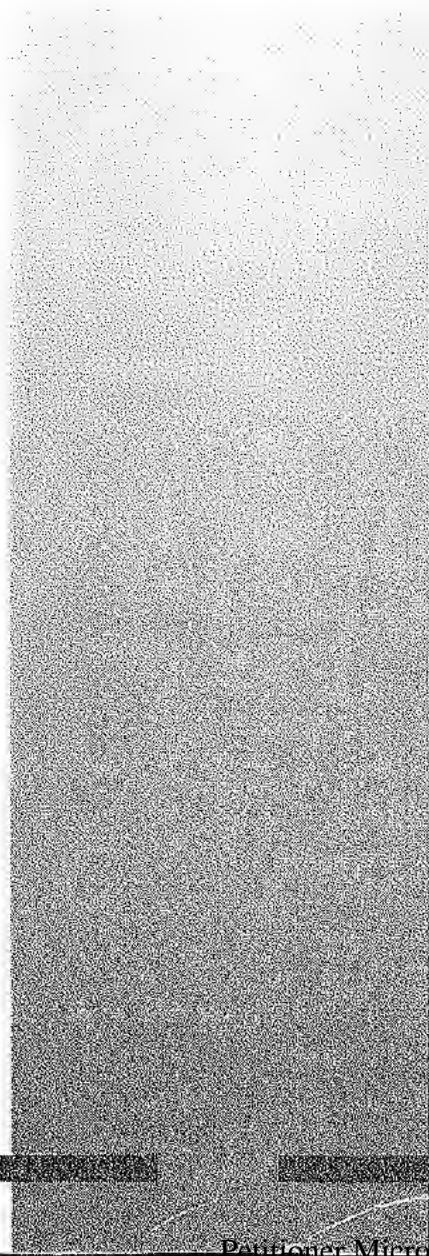
This research was supported in part by a research grant from the Faculty of Computing and Information Technology and by a Small ARC grant. The authors would like to thank Tun Heng Chiang for his work on the implementation of the display modules.

References

- [Arens, Hovy and van Mulken 1993] Y. Arens, E. Hovy and S. van Mulken, Structure and Rules in Automated Multimedia Presentation Planning. *IJCAI-93 Proceedings, the Thirteenth International Joint Conference on Artificial Intelligence* (Chambery, France, 1993), p. 1253-1259.
- [Arens, Hovy and Vossers 1993] Y. Arens, E. Hovy and M. Vossers, On the Knowledge Underlying Multimedia Presentations. *Intelligent Multimedia Interfaces*, ed. M. T. Maybury (AAAI Press, California, 1993).
- [Engelmore and Morgan 1988] R. S. Engelmore and A. J. Morgan, *Blackboard Systems* (Addison-Wesley Publishing Company, New York, 1988).
- [Graf 1992] W. Graf, Constraint-based Graphical Layout of Multimodal Presentations. *AVI'92 Proceedings - the International Workshop on Advanced Visual Interfaces* (Rome, 1992), p. 365-385.
- [Hovy and Arens 1993] E. Hovy and Y. Arens, The Planning Paradigm Required for Automated Multimedia Presentation Planning. *Proceedings of the AAAI Fall Symposium* (North Carolina, 1993).
- [Feiner and McKeown 1990] S. Feiner and K. McKeown, Coordinating Text and Graphics in Explanation Generation. *AAAI-90 Proceedings, the Eighth National Conference on Artificial Intelligence* (Boston, Massachusetts, 1990), p. 442-449.
- [McKeown et al. 1992] K. McKeown, S. K. Feiner, J. Robin, D. D. Seligmann and M. Tanenblatt, Generating Cross-References for Multimedia Explanation. *AAAI-92 Proceedings, the Tenth National Conference on Artificial Intelligence* (San Jose, California, 1992), p. 9-16.
- [Rist and André 1992] T. Rist and E. André, Incorporating Graphics Design and Realization into the Multimodal Presentation System WIP. *Proceedings of the International Workshop on Advanced Visual Interfaces* (Rome, Italy, 1992), p. 1-14.
- [Wahlster et al. 1993] W. Wahlster, E. André, W. Finkler, H. Profitlich and T. Rist, Plan-based Integration of Natural Language and Graphics Generation. *Artificial Intelligence* **63(1-2)** (1993), p. 387-427.
- [Zukerman and McConachy 1993] I. Zukerman and R. McConachy, Generating Concise Discourse that Addresses a User's Inferences. *IJCAI-93 Proceedings, the Thirteenth International Joint Conference on Artificial Intelligence* (Chambery, France, 1993), p. 1202-1207.

SamenwerkingsOrganen Brabantse Universiteiten

Proceedings of the
International Conference on
Cooperative Multimodal
Communication CMC/95
Part I



Katholieke Universiteit Brabant - Gebouw Y.102
Postbus 90153
5000 LE Tilburg

Technische Universiteit Eindhoven - Bestuursgeb
Postbus 513
5600 MB Eindhoven

Proceedings of the
International Conference on
Cooperative Multimodal
Communication CMC/95
Part I

Eindhoven, May 24-26, 1995

Harry Bunt, Robbert-Jan Beun & Tijn Borghuis (eds.)

Institut für Mathematische Maschinen
und Textverarbeitung (Informatik)
Postfach 3, 91058 Erlangen
BIBLIOTHEK



UER028023265496

CIP GEGEVENS KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Harry Bunt, Robbert-Jan Beun & Tijn Borghuis

Proceedings of the International Conference on Cooperative
Multimodal Communication, Eindhoven, May 24-26, 1995

ISBN 90-9008315-4

trefw.: mens-machine communicatie, multimedia, user-interfaces

Preface

Communication is a bidirectional activity that comes naturally in multimodal form, involving both verbal and nonverbal, vocal, visual, tactile and other means of interaction. Natural communication is also cooperative, in that the participants make an effort to understand each other, and act in a way that takes each other's goals and purposes into account, for instance helping a dialogue partner to obtain relevant information.

Technical developments increasingly allow the realization of human-computer interfaces where more sophisticated forms of visual and auditory, verbal and nonverbal information are used by the computer and where the user is allowed a greater variety of forms of expression. Two crucial aspects of natural communication are, however, still conspicuously absent in existing user interfaces:

- real cooperation from the part of the computer, based on a good understanding of the user's wants;
- true multimodality in the sense of fully integrated, simultaneous use of several modalities to convey a complex message.

As a result, human-computer communication is generally felt to be only marginally cooperative, and to be unnatural and primitive, compared to natural human communication.

The present conference aims at contributing to improving the state of the art in cooperative multimodal human-computer communication, bringing together researchers involved in the design, implementation, and application of forms of cooperative human-computer communication where natural language (typed or spoken) is used in combination with other modalities, such as visual feedback and direct manipulation. The conference focuses on formal, computational, and user aspects of building cooperative multimodal dialogue systems, with the following topics being identified in the call for papers:

- cooperativity in multimodal dialogue
- natural language semantics in a multimodal context
- formal and computational models of dialogue context
- incremental knowledge representation and dialogue
- interacting with visual domain representations
- collaborative problem solving
- constraint-based approaches to animation and visual modelling
- effective use of different interactive modalities
- modelling temporal aspects of multimodal communication
- type theory and natural language interpretation

In response to the call for papers, we have received submissions from all over the world (Europe, North America, Asia, Australia), from which the programme committee has selected 17 for paper presentation and 8 for poster presentation at the conference. In addition, the conference features a presentation of the multimodal DenK-project, which has provided the inspiration for organizing this conference, and invited papers by Mark Maybury, Wolfgang Wahlster, Bonnie Webber and Kent Wittenburg.

I would like to use this occasion to thank the members of the programme committee for reviewing the submitted contributions for the conference, and the members of the organizing committee plus the staff at the Institute for Perception Research IPO, which hosts the conference, for all their efforts to make the conference run smoothly. Particular thanks are due to the Samenwerkingsorgaan Brabantse Universiteiten (the organization for cooperation between the universities in the province of Brabant, i.e. the universities of Tilburg and Eindhoven), and to the Royal Dutch Academy of Sciences (KNAW) for their financial support.

Harry Bunt
Program Committee chairman.

Program Committee

Harry Bunt (chair)

Norman Badler

Walther von Hahn

Hans Kamp

Joseph Mariani

Paul Mc Kevitt

Kees van Overveld

Donia Scott

Bonnie Webber

Jeroen Groenendijk

Dieter Huber

John Lee

Mark Maybury

Rob Nederpelt

Ray Perrault

Wolfgang Wahlster

Kent Wittenburg

Organizing Committee

Robbert-Jan Beun (chair)

Tijn Borghuis

Harry Bunt

Rob Nederpelt

Marianné Wagemans

Sponsorship

Koninklijke Nederlandse Akademie van Wetenschappen (KNAW)

Samenwerkingsorgaan Brabantse Universiteiten (SOBU)

ACL Special Interest Group in Multimedia (SIGMEDIA)

Table of Contents

PART I

Invited Papers

| | |
|---|----|
| Toward Cooperative Multimedia Interaction (abstract) | 3 |
| <i>Mark T. Maybury</i> | |
| Instructing Animated Agents: Viewing Language in Behavioral Terms | 5 |
| <i>Bonnie Webber</i> | |
| Visual Language Parsing: If I Had a Hammer... .. | 17 |
| <i>Kent Wittenburg</i> | |

Submitted Papers

| | |
|---|-----|
| Contexts in Dialogue | 37 |
| <i>Tijn Borghuis</i> | |
| Management of Non-Standard Devices for Multimodal User Interfaces under UNIX/X11 | 49 |
| <i>Patrick Bourdot, Mike Krus and Rachid Gherbi</i> | |
| The Role of Multimodal Communication in Cooperation and Intention Recognition: The Case of Air Traffic Control | 63 |
| <i>Marie-Christine Bressolle, Bernard Pavard and Marcel Leroux</i> | |
| Cooperative Multimodal Communication in the DenK Project | 79 |
| <i>Harry Bunt, René Ahn, Robbert-Jan Beun, Tijn Borghuis and Kees van Overveld</i> | |
| Multimodal Maps: An Agent Based Approach | 103 |
| <i>Adam Cheyer and Luc Julia</i> | |
| Object Reference During Task-related Terminal Dialogues | 115 |
| <i>Anita Cremers</i> | |
| Speakers' Responses to Requests for Repetition in a Multimedia Language Processing Environment | 129 |
| <i>Laurel Fais, Kyung-ho Loken-Kim and Young-Duk Park</i> | |
| A Cooperative Approach for Multimodal Presentation Planning | 145 |
| <i>Yi Han and Ingrid Zukerman</i> | |

PART II

| | |
|--|-----|
| Studies into Full Integration of Language and Action | 161 |
| <i>Carla Huls and Edwin Bos</i> | |
| Referent Identification Requests in Multimodal Dialogues | 175 |
| <i>Tsuneaki Kato and Yukiko I. Nakano</i> | |
| Anaphora in Multimodal Discourse | 193 |
| <i>John Lee and Keith Stenning</i> | |
| Towards Adequate Representation Technologies for Multimodal Interfaces . | 207 |
| <i>Jean Claude Martin, Remko Veldman and Dominique Béroule</i> | |
| Designing a Multimedia Interface for Operators Assembling Circuit Boards | 225 |
| <i>Fergal McCaffery, Michael McTear and Maureen Murphy</i> | |
| Synthesizing Cooperative Conversation | 237 |
| <i>Catherine Pelachaud, Justine Cassell, Norman Badler, Mark Steedman, Scott Prevost and Matthew Stone</i> | |
| Topic Management in Information Dialogues | 257 |
| <i>Mieke Rats</i> | |
| An Approach to Solving the Symbol Grounding Problem: Neural Networks for Object Naming and Retrieval | 273 |
| <i>N.J. Sales and R.G. Evans</i> | |
| Modeling and Processing of the Oral and Tactile Activities in the Georal Tactile System | 287 |
| <i>J. Sioux, M. Guyomard, F. Multon and C. Remondeau</i> | |
| Poster presentations | |
| The Generalized Display Processor: a Platform for Real-time Interactive Computer Animation | 299 |
| <i>Gino van Bergen and Kees van Overveld</i> | |
| Communication and Co-ordination Difficulties During Interactive Tele-Teaching in the Independent Problem-Solving Format | 301 |
| <i>Martin, Colbert</i> | |
| Automatic Generation of Statistical Graphics | 303 |
| <i>Massimo Fasciano and Guy Lapalme</i> | |

| | |
|---|-----|
| Computer-Aided Negotiation Support in Hypermedia Multi-Agent Systems | 307 |
| <i>Igor V. Kotenko and Dmitry L. Krechman</i> | |
| Multimodal Dialogue Semantics Against a Dynamic World Model | 311 |
| <i>Susann Luperfoy and David Duff</i> | |
| Two Basic Orientations of Subject in World and in Human-Computer Communication | 315 |
| <i>Olga I. Marchenko</i> | |
| Internalized Contexts in NL Semantics | 317 |
| <i>Wlodek Zadrozny</i> | |
| Author Index | 321 |
| Subject Index | 323 |

Multimodal Maps: An Agent-based Approach

Adam Cheyer and Luc Julia

SRI International
333 Ravenswood Ave
Menlo Park, CA 94025 - USA

Abstract

In this paper, we discuss how multiple input modalities may be combined to produce more natural user interfaces. To illustrate this technique, we present a prototype map-based application for a travel planning domain. The application is distinguished by a synergistic combination of handwriting, gesture and speech modalities; access to existing data sources including the World Wide Web; and a mobile handheld interface. To implement the described application, a hierarchical distributed network of heterogeneous software agents was augmented by appropriate functionality for developing synergistic multimodal applications.

Key words: Multimodal Interface, Agent Architecture, Distributed Artificial Intelligence.

1 Introduction

As computer systems become more powerful and complex, efforts to make computer interfaces more simple and natural become increasingly important. Natural interfaces should be designed to facilitate communication in ways people are already accustomed to using. Such interfaces allow users to concentrate on the tasks they are trying to accomplish, not worry about what they must do to control the interface.

In this paper, we begin by discussing what input modalities humans are comfortable using when interacting with computers, and how these modalities should best be combined in order to produce natural interfaces. In section three, we present a prototype map-based application for the travel planning domain which uses a synergistic combination of several input modalities. Section four describes the agent-based approach we used to implement the application and the work on which it is based. In section five, we summarize our conclusions and future directions.

2 Natural Input

2.1 Input Modalities

Direct manipulation interface technologies are currently the most widely used techniques for creating user interfaces. Through the use of menus and a graphical user interface, users are presented with sets of discrete actions and the objects on which to perform them. Pointing

devices such as a mouse facilitate selection of an object or action, and drag and drop techniques allow items to be moved or combined with other entities or actions.

With the addition of electronic pen devices, gestural drawings add a new dimension direct manipulation interfaces. Gestures allow users to communicate a surprisingly wide range of meaningful requests with a few simple strokes. Research has shown that multiple gestures can be combined to form dialog, with rules of temporal grouping overriding temporal sequencing [22]. Gestural commands are particularly applicable to graphical or editing type tasks.

Direct manipulation interactions possess many desirable qualities: communication is generally fast and concise; input techniques are easy to learn and remember; the user has a good idea about what can be accomplished, as the visual presentation of the available actions is generally easily accessible. However, direct manipulation suffers from limitations when trying to access or describe entities which are not or can not be visualized by the user.

Limitations of direct manipulation style interfaces can be addressed by another interface technology, that of natural language interfaces. Natural language interfaces excel in describing entities that are not currently displayed on the monitor, in specifying temporal relations between entities or actions, and in identifying members of sets. These strengths are exactly the weaknesses of direct manipulation interfaces, and concurrently, the weaknesses of natural language interfaces (ambiguity, conceptual coverage, etc.) can be overcome by the strengths of direct manipulation.

Natural language content can be entered through different input modalities, including typing, handwriting, and speech. It is important to note that, while the same textual content can be provided by the three modalities, each modality has widely varying properties.

- Spoken language is the modality used first and foremost in human-human interactive problem solving [4]. Speech is an extremely fast medium, several times faster than typing or handwriting. In addition, speech input contains content that is not present in other forms of natural language input, such as prosody, tone and characteristics of the speaker (age, sex, accent).
- Typing is the most common way of entering information into a computer, because it is reasonably fast, very accurate, and requires no computational resources.
- Handwriting has been shown to be useful for certain types of tasks, such as performing numerical calculations and manipulating names which are difficult to pronounce [18, 19]. Because of its relatively slow production rate, handwriting may induce users to produce different types of input than is generated by spoken language; abbreviations, symbols and non-grammatical patterns may be expected to be more prevalent amid written input.

2.2 Combination of Modalities

As noted in the previous section, direct manipulation and natural language seem to be very complementary modalities. It is therefore not surprising that a number of multimodal systems combine the two.

Notable among such systems is the Cohen's Shoptalk system [6], a prototype manufacturing and decision-support system that aids in tasks such as quality assurance monitoring, and production scheduling. The natural language module of Shoptalk is based on the Chat-85

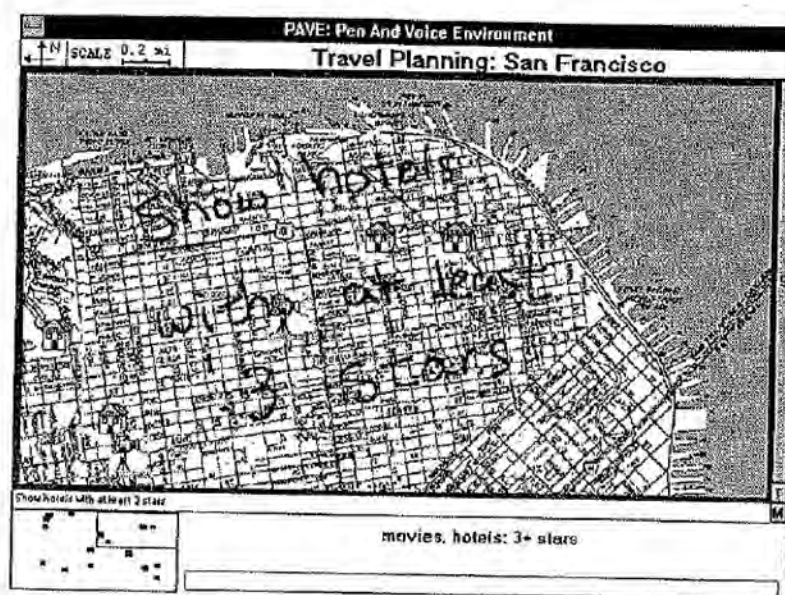


Figure 1: Multimodal Application for Travel Planning

natural language system [25] and is particularly good at handling time, tense, and temporal reasoning.

A number of systems have focused on combining the speed of speech with the reference provided by direct manipulation of a mouse pointer. Such systems include the XTRA system [1], CUBRICON [15], the PAC-Amodeus model [16], and TAPAGE [9].

XTRA and CUBRICON are both systems that combine complex spoken input with mouse clicks, using several knowledge sources for reference identification. CUBRICON's domain is a map-based task, making it similar to the application developed in this paper. However, the two are different in that CUBRICON can only use direct manipulation to indicate a specific item, whereas our system produces a richer mixing of modalities by adding both gestural and written language as input modalities.

The PAC-Amodeus systems such as VoicePaint and Notebook allow the user to synergistically combine vocal or mouse-click commands when interacting with notes or graphical objects. However, due to the selected domains, the natural language input is very simple, generally of the style "Insert a note here."

TAPAGE is another system that allows true synergistic combination of spoken input with direct manipulation. Like PAC-Amodeus, TAPAGE's domain provides only simple linguistic input. However, TAPAGE uses a pen-based interface instead of a mouse, allowing gestural commands. TAPAGE, selected as a building block for our map application, will be described more in detail in section 4.2.

Other interesting work regarding the simultaneous combination of handgestures and gaze can be found in [2, 13].

3 A Multimodal Map Application

In this section, we will describe a prototype map-based application for a travel planning domain. In order to provide the most natural user interface possible, the system permits the

user to simultaneously combine direct manipulation, gestural drawings, handwritten, typed and spoken natural language. When designing the system, other criteria were considered as well:

- The user interface must be light and fast enough to run on a handheld PDA while able to access applications and data that may require a more powerful machine.
- Existing commercial or research natural language and speech recognition systems should be used.
- Through the multimodal interface, a user must be able to transparently access a wide variety of data sources, including information stored in HTML form on the World Wide Web.

As illustrated in Figure 1, the user is presented with a pen sensitive map display on which drawn gestures and written natural language statements may be combined with spoken input. As opposed to a static paper map, the location, resolution, and content presented by the map change, according to the requests of the user. Objects of interest, such as restaurants, movie theaters, hotels, tourist sites, municipal buildings, etc. are displayed as icons. The user may ask the map to perform various actions. For example :

- *distance calculation* : e.g. "How far is the hotel from Fisherman's Wharf?"
- *object location* : e.g. "Where is the nearest post office?"
- *filtering* : e.g. "Display the French restaurants within 1 mile of this hotel."
- *information retrieval* : e.g. "Show me all available information about Alcatraz."

The application also makes use of multimodal (multimedia) output as well as input: video, text, sound and voice can all be combined when presenting an answer to a query.

During input, requests can be entered using gestures (see Figure 2 for sample gestures), handwriting, voice, or a combination of pen and voice. For instance, in order to calculate the distance between two points on the map, a command may be issued using the following:

- *gesture*, by simply drawing a line between the two points of interest.
- *voice*, by speaking "What is the distance from the post office to the hotel?"
- *handwriting*, by writing "dist p.o. to hotel?"
- *synergistic combination of pen and voice*, by speaking "What is the distance from here to this hotel?" while simultaneously indicating the specified locations by pointing or circling.

Notice that in our example of synergistic combination of pen and voice, the arguments to the verb "distance" can be specified before, at the same time, or shortly after the vocalization of the request to calculate the distance. If a user's request is ambiguous or underspecified, the system will wait several seconds and then issue a prompt requesting additional information.

The user interface runs on pen-equipped PC's or a Dauphin handheld PDA ([7]) using either a microphone or a telephone for voice input. The interface is connected either by

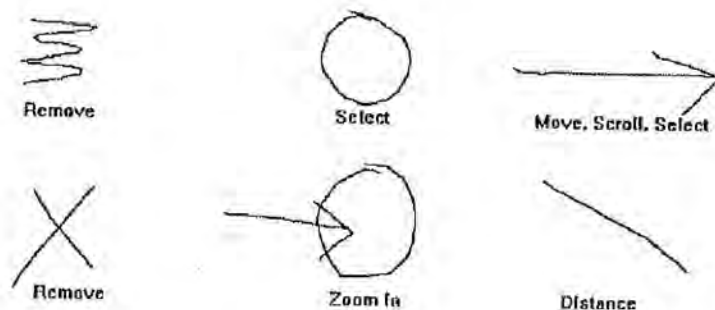


Figure 2: Sample gestures

modem or ethernet to a server machine which will manage database access, natural language processing and speech recognition for the application. The result is a mobile system that provides a synergistic pen/voice interface to remote databases.

In general, the speed of the system is quite acceptable. For gestural commands, which are handled locally on the user interface machine, a response is produced in less than one second. For handwritten commands, the time to recognize the handwriting, process the English query, access a database and begin to display the results on the user interface is less than three seconds (assuming an ethernet connection, and good network and database response). Solutions to verbal commands are displayed in three to five seconds after the end of speech has been detected; partial feedback indicating the current status of the speech recognition is provided earlier.

4 Approach

In order to implement the application described in the previous section, we chose to augment a proven agent-based architecture with functionalities developed for a synergistically multimodal application. The result is a flexible methodology for designing and implementing distributed multimodal applications.

4.1 Building Blocks

4.1.1 Open Agent Architecture

The Open Agent Architecture (OAA) [5] provides a framework for coordinating a society of agents which interact to solve problems for the user. Through the use of agents, the OAA provides distributed access to commercial applications, such as mail systems, calendar programs, databases, etc.

The Open Agent Architecture possesses several properties which make it a good candidate for our needs:

- An Interagent Communication Language (ICL) and Query Protocol have been developed, allowing agents to communicate among themselves. Agents can run on different platforms and be implemented in a variety of programming languages.
- Several natural language systems have been integrated into the OAA which convert English into the Interagent Communication Language. In addition, a speech recognition

agent has been developed to provide transparent access to the Corona speech recognition system.

- The agent architecture has been used to provide natural language and agent access to various heterogeneous data and knowledge sources.
- Agent interaction is very fine-grained. The architecture was designed so that a number of agents can work together, when appropriate in parallel, to produce fast responses to queries.

The architecture for the OAA, based loosely on Schwartz's FLiPSiDE system[23], uses a hierarchical configuration where client agents connect to a "facilitator" server. Facilitators provide content-based message routing, global data management, and process coordination for their set of connected agents. Facilitators can, in turn, be connected as clients of other facilitators. Each facilitator records the published functionality of their sub-agents, and when queries arrive in Interagent Communication Language form, they are responsible for breaking apart any complex queries and for distributing goals to the appropriate agents. An agent solving a goal may require supporting information and the agent architecture provides numerous means of requesting data from other agents or from the user.

Among the assortment of agent architectures, the Open Agent Architecture can be most closely compared to work by the ARPA knowledge sharing community [10]. The OAA's query protocol, Interagent Communication Language and Facilitator mechanisms have similar instantiations in the SHADE project, in the form of KQML, KIF and various independent capability matchmakers. Other agent architectures, such as General Magic's Telescript [11], MASCOs [20], or the CORBA distributed object approach [17] do not provide as fully developed mechanisms for interagent communication and delegation.

The Open Agent Architecture provides capability for accessing distributed knowledge sources through natural language and voice, but it is lacking integration with a synergistic multimodal interface.

4.1.2 TAPAGE

TAPAGE (edition de Tableaux par la Parole et la Geste) is a synergistic pen/voice system for designing and correcting tables.

To capture signals emitted during a user's interaction, TAPAGE integrates a set of modality agents, each responsible for a very specialized kind of signal [9]. The modality agents are connected to an "interpret agent" which is responsible for combining the inputs across all modalities to form a valid command for the application. The interpret agent receives filtered results from the modality agents, sorts the information into the correct fields, performs type-checking on the arguments, and prompts the user for any missing information, according to the model of the interaction. The interpret agent is also responsible for merging the data streams sent by the modality agents, and for resolving ambiguities among them, based on its knowledge of the application's internal state. Another function of the interpret agent is to produce reflexes: reflexes are actions output at the interface level without involving the functional core of the application.

The TAPAGE system can accept multimodal input, but it is not a distributed system; its functional core is fixed. In TAPAGE, the set of linguistic input is limited to a *verb object argument* format.

4.2 Synthesis

In the Open Agent Architecture, agents are distributed entities that can run on different machines, and communicate together to solve a task for the user. In TAPAGE, agents are used to provide streams of input to a central interpret process, responsible for merging incoming data. A generalization of these two types of agents could be :

Macro Agents: contain some knowledge and ability to reason about a domain, and can answer or make queries to other macro agents using the Interagent Communication Language.

Micro Agents: are responsible for handling a single input or output data stream, either filtering the signal to or from a hierarchically superior "interpret" agent.

The network architecture that we used was hierarchical at two resolutions - micro agents are connected to a superior macro agent, and macro agents are connected in turn to a facilitator agent. In both cases, a server is responsible for the supervision of its client sub-agents.

In order to describe our implementation, we will first give a description of each agent used in our application and then illustrate the flow of communication among agents produced by a user's request.

Speech Recognition (SR) Agent: The SR agent provides a mapping from the Interagent Communication Language to the API for the Decipher (Corona) speech recognition system [4], a continuous speech speaker independent recognizer based on Hidden Markov Model technology. This macro agent is also responsible for supervising a child micro agent whose task is to control the speech data stream. The SR agent can provide feedback to an interface agent about the current status and progress of the micro agent (e.g. "listening", "end of speech detected", etc.) This agent is written in C.

Natural Language (NL) Parser Agent: translates English expressions into the Interagent Communication Language (ICL). For a more complete description of the ICL, see [5]. The NL agent we selected for our application is the simplest of those integrated into the OAA. It is written in Prolog using Definite Clause Grammars, and supports a distributed vocabulary; each agent dynamically adds word definitions as it connects to the network. A current project is underway to integrate the Gemini natural language system [4], a robust bottom up parser and semantic interpreter specifically designed for use in Spoken Language Understanding projects.

Database Agents: Database agents can reside at local or remote locations and can be grouped hierarchically according to content. Micro agents can be connected to database agents to monitor relevant positions or events in real time. In our travel planning application, database agents provide maps for each city, as well as icons, vocabulary and information about available hotels, restaurants, movies, theaters, municipal buildings and tourist attractions. Three types of databases were used: Prolog databases, X.500 hierarchical databases, and data loaded automatically by scanning HTML pages from the World Wide Web (WWW). In one instance, a local newspaper provides weekly updates to its Mosaic-accessible list of current movie times and reviews, as well as adding several new restaurant reviews to a growing collection; this information is extracted by an HTML reading database agent and made accessible to the agent architecture. Descriptions and addresses of new restaurants are presented to the user on request, and the user can choose to add them to the permanent database by specifying positional coordinates on the map (eg. "add this new restaurant here"), information lacking in the WWW database.

Reference Resolution Agent: This agent is responsible for merging requests arriving in parallel from different modalities, and for controlling interactions between the user interface

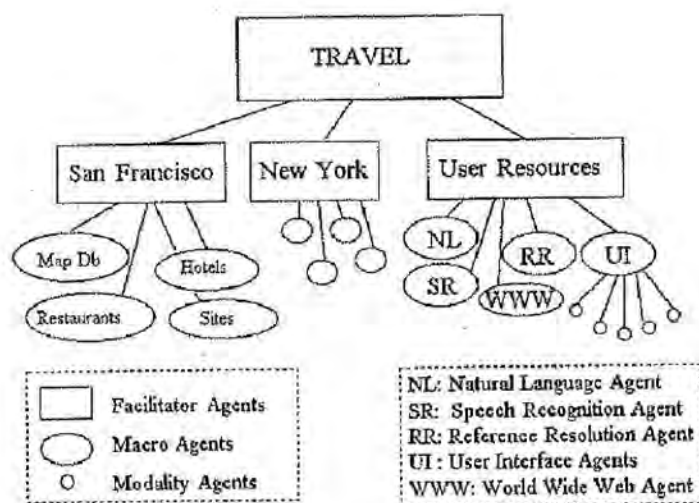


Figure 3: Agent Architecture for Map Application

agent, database agents and modality agents. In this implementation, the reference resolution agent is domain specific: knowledge is encoded as to what actions must be performed to resolve each possible type of ICL request in its particular domain. For a given ICL logical form, the agent can verify argument types, supply default values, and resolve argument references. Some argument references are descriptive ("How far is it to the hotel on Emerson Street?"); in this case, a domain agent will try to resolve the definite reference by sending database agent requests. Other references, particularly when contextual or deictic, are resolved by the user interface agent ("What are the rates for this hotel?"). Once arguments to a query have been resolved, this agent coordinates the actions and calculations necessary to produce the result of the request.

Interface Agent: This macro agent is responsible for managing what is currently being displayed to the user, and for accepting the user's multimodal input. The Interface Agent also coordinates client modality agents and resolves ambiguities among them: handwriting and gestures are interpreted locally by micro agents and combined with results from the speech recognition agent, running on a remote speech server. The handwriting micro-agent interfaces with the Microsoft PenWindows API and accesses a handwriting recognizer by CIC Corporation. The gesture micro-agent accesses recognition algorithms developed for TAPAGE.

An important task for the interface agent is to record which objects of each type are currently salient, in order to resolve contextual references such as "the hotel" or "where I was before." Deictic references are resolved by gestural or direct manipulation commands. If no such indication is currently specified, the user interface agent waits long enough to give the user an opportunity to supply the value, and then prompts the user for it.

We shall now give an example of the distributed interaction of agents for a specific query. In the following example, all communication among agents passes transparently through a facilitator agent in an undirected fashion; this process is left out of the description for brevity.

1. A user speaks: "How far is the restaurant from this hotel?"

2. The speech recognition agent monitors the status and results from its micro agent, sending feedback received by the user interface agent. When the string is recognized, a translation is requested.
3. The English request is received by the NL agent and translated into ICL form.
4. The reference resolution agent (RR) receives the ICL distance request containing one definite and one deictic reference and asks for resolution of these references.
5. The interface agent uses contextual structures to find what "the restaurant" refers to, and waits for the user to make a gesture indicating "the hotel", issuing prompts if necessary.
6. When the references have been resolved, the domain agent (RR) sends database requests asking for the coordinates of the items in question. It then calculates the distance according to the scale of the currently displayed map, and requests the user interface to produce output displaying the result of the calculation.

5 Conclusions

By augmenting an existing agent-based architecture with concepts necessary for synergistic multimodal input, we were able to rapidly develop a map-based application for a travel planning task. The resulting application has met our initial requirements: a mobile, synergistic pen/voice interface providing good natural language access to heterogeneous distributed knowledge sources. The approach used was general and should provide a for developing synergistic multimodal applications for other domains.

The system described here is one of the first that accepts commands made of synergistic combinations of spoken language, handwriting and gestural input. This fusion of modalities can produce more complex interactions than in many systems and the prototype application will serve as a testbed for acquiring a better understanding of multimodal input.


In the near future, we will continue to verify and extend our approach by building other multimodal applications. We are interested in generalizing the methodology even further; work has already begun on an agent-building tool which will simplify and automate many of the details of developing new agents and domains.

References

- [1] Allegayer, J, Jansen-Winkel, R., Reddig, C. and Reithinger, N. "Bidirectional use of knowledge in the multi-modal NL access system XTRA". In Proceedings of IJCAI-89, Detroit, pp. 1492-1497.
- [2] Bolt, R. "Put that there: Voice and Gesture at the Graphic Interface". Computer Graphics, 14(3), 1980, pp. 262-270.
- [3] Bellik, Y. and Teil, D. "Les types de multimodalites", In Proc. IIM'92 (Paris), pp. 22-28.
- [4] Cohen, M., Murveit, H., Bernstein, J., Price, P., Weintraub, M., "The DECIPHER Speech Recognition System". 1990 IEEE ICASSP, pp. 77-80.

- [5] Cohen, P.R., Cheyer, A., Wang, M. and Baeg, S.C. "An Open Agent Architecture". In Proc. AAAI'94 - SA (Stanford), pp. 1-8.
- [6] Cohen, P. "The role of natural language in a multimodal interface". Proceedings of UIST'92, 143-149.
- [7] Dauphin DTR-1 User's Manual, Dauphin Technology, Inc. 337 E. Butterfield Rd., Suite 900, Lombard, Ill 60148.
- [8] Dowding, J., Gawron, J.M., Appelt, D., Bear, J., Cherny, L., Moore, B. and Moran D., "Gemini: A natural language system for spoken-language understanding", Technical Note 527, AI Center, SRI International, 333 Ravenswood Ave., Menlo Park, CA 94025, April 1993.
- [9] Faure, C. and Julia, L. "An Agent-Based Architecture for a Multimodal Interface". In Proc. AAAI'94 - IM4S (Stanford), pp. 82-86.
- [10] Genesereth, M. and Singh, N.P. "A knowledge sharing approach to software interoperation". Computer Science Department, Stanford University, unpublished ms., 1994.
- [11] General Magic, Inc., "Telescript Product Documentation", 1995.
- [12] Julia, L. and Faure, C. "A Multimodal Interface for Incremental Graphic Document Design". HCI International '93, Orlando.
- [13] Koons, D.B., Sparrell, C.J., and Thorisson, K.R. "Integrating Simultaneous Input from Speech, Gaze and Hand Gestures". In *Intelligent Multimedia Interfaces*, Edited by Mark Maybury, Menlo Park, CA, AAAI Press, 1993.
- [14] Maybury, M.T. (ed.), *Intelligent Multimedia Interfaces*, AAAI Press/MIT Press: Menlo Park, Ca, 1993.
- [15] Neal, J.G., and Shapiro, S.C. "Intelligent Multi-media Interface Technology". In *Intelligent User Interfaces*, Edited by J. Sullivan and S. Tyler, Addison-Wesley Pub. Co., Reading, MA, 1991.
- [16] Nigay, L. and Coutaz, J. "A Design Space for Multimodal Systems: Concurrent Processing and Data Fusion". In Proc. InterCHI'93 (Amsterdam), ACM Press, pp. 172-178.
- [17] Object Management Group, "The Common Object Request Broker: Architecture and Specification", OMG Document Number 91.12.1, December 1991.
- [18] Oviatt, S. "Toward Empirically-Based Design of Multimodal Dialogue Systems". In Proc. AAAI'94 - IM4S (Stanford), pp. 30-36.
- [19] Oviatt, S. and Olsen, E. "Integration Themes in Multimodal Human-Computer Interaction". Proceedings of ICSLP'94, Yokohama, pp. 551-554.
- [20] Park, S.K., Choi J.M., Myeong-Wuk J., Lee G.L., and Lim Y.H. "MASCOS : A Multi-Agent System as the Computer Secretary". Submitted for publication.
- [21] Pfaff, G. and Ten Hagen, P.J.W. *Seeheim workshop on User Interface Management Systems* (Berlin), Springer- Verlag.

- [22] Rhyne J. "Dialogue Management for Gestural Interfaces". *Computer Graphics*, 21(2), 1987, pp. 137-142.
- [23] Schwartz, D.G. "Cooperating heterogeneous systems: A blackboard-based meta approach". Technical Report 93-112, Center for Automation and Intelligent Systems Research, Case Western Reserve University, Cleveland Ohio, April 1993. Unpublished PhD. thesis.
- [24] Sullivan, J. and Tyler, S. (eds.), *Intelligent User Interfaces*, Addison-Wesley Pub. Co., Reading, MA, 1991.
- [25] Warren, D. and Pereira, F., "An Efficient Easily Adaptable System for Interpreting Natural Language Queries", in *American Journal of Computational Linguistics*, 8(3), 1982, pp. 110-123.
- [26] Wauchope, K., "Eucalyptus: Integrating Natural Language with a Graphical User Interface." Naval Research Laboratory Technical Report NRL/FR/5510-94-9711, in press, 1994.

[Zurück zur Trefferliste](#)
   
Katalog UB Erlangen-Nürnberg (1/3)
[\[« »\]](#)
Weitersuchen & Fernleihe
[>> Weitersuchen \(Fernleihe\)](#)
[>> Suchanfrage ändern](#)

Speichern in:

 [Speichern](#)

Anzeige:

[Einzelbände](#)
FAUdok

Über den Aufsatzlieferdienst FAUdok (siehe "Bestellung/Verfügbarkeit") können sich Studierende und Beschäftigte der Universität Erlangen-Nürnberg Aufsätze aus Zeitschriften und Büchern per E-Mail zuschicken lassen. Bestellung, Lieferfristen und Standorte, die an FAUdok teilnehmen

Proceedings of the International Conference on Cooperative Multimodal Communication : CMC 95 ; Eindhoven, May 24 - 26, 1995

CMC 95 ; Eindhoven, May 24 - 26, 1995


Erscheinungsort: Tilburg

Verlag: Katholieke Univ. Brabant [u.a.]

ISBN: 9090083154

Schlagwort: Mensch-Maschine-Kommunikation / Kongress / Eindhoven <1995>

| Exemplare | Bestellung/Verfügbarkeit | mehr zum Titel |
|--|--------------------------|----------------|
| <p>Titel: Proceedings of the International Conference on Cooperative Multimodal Communication</p> <p>Zusatz zum Titel: CMC 95 ; Eindhoven, May 24 - 26, 1995</p> <p>Von: Harry Bunt ... (eds.)</p> <p>Institution: International Conference on Cooperative Multimodal Communication (1, 1995, Eindhoven)</p> <p>Erscheinungsort: Tilburg</p> <p>Verlag: Katholieke Univ. Brabant [u.a.]</p> <p>Reihe: Samenwerkings Orgaan Brabantse Universiteiten</p> <p>ISBN: 9090083154</p> <p>Schlagwort: Mensch-Maschine-Kommunikation / Kongress / Eindhoven <1995></p> <p>ID Verbundkatalog: BV010328174</p> | | |

 [Online-Auskunft: Fragen Sie uns!](#)
[Universitätsbibliothek Erlangen-Nürnberg | Kontakt](#)

 Ansicht: [Klassisch](#) | [Mobil](#)

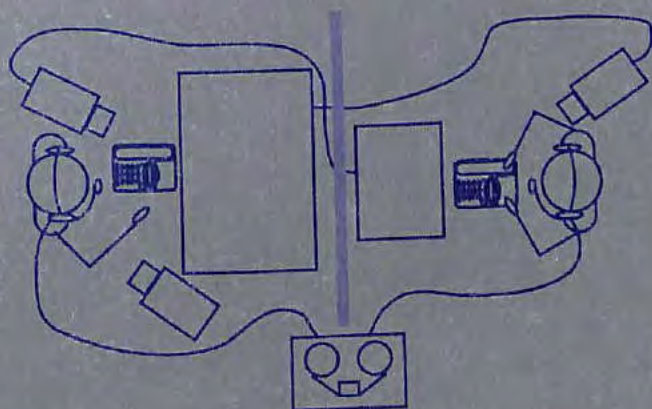
Lecture Notes in Artificial Intelligence 1374

Subseries of Lecture Notes in Computer Science

Harry Bunt Robert-Jan Beun
Tijn Borghuis (Eds.)

Multimodal Human-Computer Communication

Systems, Techniques, and Experiments



Springer



Fig. 1. Multimodal application for travel planning

- The user interface must be light and fast enough to run on a handheld PDA while able to access applications and data that may require a more powerful machine.
- Existing commercial or research natural language and speech recognition systems should be used.
- Through the multimodal interface, a user must be able to transparently access a wide variety of data sources, including information stored in HTML form on the World Wide Web.

As illustrated in Fig. 1, the user is presented with a pen sensitive map display on which drawn gestures and written natural language statements may be combined with spoken input. As opposed to a static paper map, the location, resolution, and content presented by the map change, according to the requests of the user. Objects of interest, such as restaurants, movie theaters, hotels, tourist sites, municipal buildings, etc. are displayed as icons. The user may ask the map to perform various actions. For example :

- *distance calculation* : e.g. "How far is the hotel from Fisherman's Wharf?"
- *object location* : e.g. "Where is the nearest post office?"
- *filtering* : e.g. "Display the French restaurants within 1 mile of this hotel."
- *information retrieval* : e.g. "Show me all available information about Alcatraz."

The application also makes use of multimodal (multimedia) output as well as input: video, text, sound and voice can all be combined when presenting an answer to a query.

During input, requests can be entered using gestures (see Fig. 2 for sample gestures), handwriting, voice, or a combination of pen and voice. For instance, in order to calculate the distance between two points on the map, a command may be issued using the following:

- *gesture*, by simply drawing a line between the two points of interest.
- *voice*, by speaking "What is the distance from the post office to the hotel?"
- *handwriting*, by writing "dist p.o. to hotel?"
- *synergistic combination of pen and voice*, by speaking "What is the distance from here to this hotel?" while simultaneously indicating the specified locations by pointing or circling.

Notice that in our example of synergistic combination of pen and voice, the arguments to the verb "distance" can be specified before, at the same time, or shortly after the vocalization of the request to calculate the distance. If a user's request is ambiguous or underspecified, the system will wait several seconds and then issue a prompt requesting additional information.

The user interface runs on pen-equipped PC's or a Dauphin handheld PDA (Dauphin, DTR-1 User's Manual) using either a microphone or a telephone for voice input. The interface is connected either by modem or ethernet to a server machine which will manage database access, natural language processing and speech recognition for the application. The result is a mobile system that provides a synergistic pen/voice interface to remote databases.

In general, the speed of the system is quite acceptable. For gestural commands, which are handled locally on the user interface machine, a response is produced in less than one second. For handwritten commands, the time to recognize the handwriting, process the English query, access a database and begin to display the results on the user interface is less than three seconds (assuming an ethernet connection, and good network and database response). Solutions to verbal commands are displayed in three to five seconds after the end of speech has been detected; partial feedback indicating the current status of the speech recognition is provided earlier.

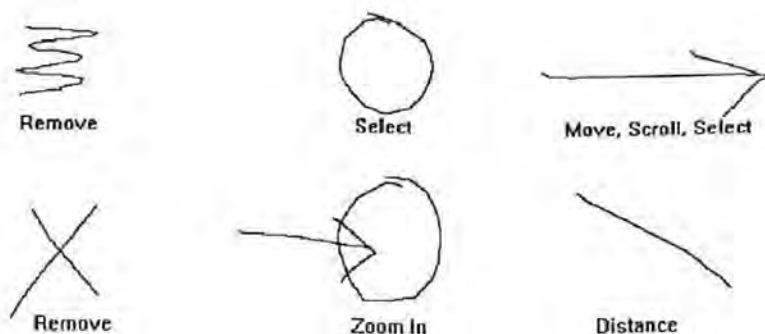


Fig. 2. Sample gestures

4 Approach

In order to implement the application described in the previous section, we chose to augment a proven agent- based architecture with functionalities developed for a synergistically multimodal application. The result is a flexible methodology for designing and implementing distributed multimodal applications.

4.1 Building Blocks

Open Agent Architecture. The Open Agent Architecture (OAA) (Cohen et al., 1994) provides a framework for coordinating a society of agents which interact to solve problems for the user. Through the use of agents, the OAA provides distributed access to commercial applications, such as mail systems, calendar programs, databases, etc.

The Open Agent Architecture possesses several properties which make it a good candidate for our needs:

- An Interagent Communication Language (ICL) and Query Protocol have been developed, allowing agents to communicate among themselves. Agents can run on different platforms and be implemented in a variety of programming languages.
- Several natural language systems have been integrated into the OAA which convert English into the Interagent Communication Language. In addition, a speech recognition agent has been developed to provide transparent access to the Corona speech recognition system.
- The agent architecture has been used to provide natural language and agent access to various heterogeneous data and knowledge sources.
- Agent interaction is very fine-grained. The architecture was designed so that a number of agents can work together, when appropriate in parallel, to produce fast responses to queries.

The architecture for the OAA, based loosely on Schwartz's FLiPSiDE system (Schwartz, 1993), uses a hierarchical configuration where client agents connect to a "facilitator" server. Facilitators provide content-based message routing, global data management, and process coordination for their set of connected agents. Facilitators can, in turn, be connected as clients of other facilitators. Each facilitator records the published functionality of their sub-agents, and when queries arrive in Interagent Communication Language form, they are responsible for breaking apart any complex queries and for distributing goals to the appropriate agents. An agent solving a goal may require supporting information and the agent architecture provides numerous means of requesting data from other agents or from the user.

Among the assortment of agent architectures, the Open Agent Architecture can be most closely compared to work by the ARPA knowledge sharing community (Genesereth and Singh, 1994). The OAA's query protocol, Interagent Communication Language and Facilitator mechanisms have similar instantiations in

the SHADE project, in the form of KQML, KIF and various independent capability matchmakers. Other agent architectures, such as General Magic's Telescript (General Magic, 1995), MASCOS (Park et al, submitted), or the CORBA distributed object approach (Object Management Group, 1991) do not provide as fully developed mechanisms for interagent communication and delegation.

The Open Agent Architecture provides capability for accessing distributed knowledge sources through natural language and voice, but it is lacking integration with a synergistic multimodal interface.

TAPAGE. TAPAGE (edition de Tableaux par la Parole et la Geste) is a synergistic pen/voice system for designing and correcting tables.

To capture signals emitted during a user's interaction, TAPAGE integrates a set of modality agents, each responsible for a very specialized kind of signal (Faure and Julia, 1994). The modality agents are connected to an 'interpret agent' which is responsible for combining the inputs across all modalities to form a valid command for the application. The interpret agent receives filtered results from the modality agents, sorts the information into the correct fields, performs type-checking on the arguments, and prompts the user for any missing information, according to the model of the interaction. The interpret agent is also responsible for merging the data streams sent by the modality agents, and for resolving ambiguities among them, based on its knowledge of the application's internal state. Another function of the interpret agent is to produce reflexes: reflexes are actions output at the interface level without involving the functional core of the application.

The TAPAGE system can accept multimodal input, but it is not a distributed system; its functional core is fixed. In TAPAGE, the set of linguistic input is limited to a *verb object argument* format.

4.2 Synthesis

In the Open Agent Architecture, agents are distributed entities that can run on different machines, and communicate together to solve a task for the user. In TAPAGE, agents are used to provide streams of input to a central interpret process, responsible for merging incoming data. A generalization of these two types of agents could be:

Macro Agents: contain some knowledge and ability to reason about a domain, and can answer or make queries to other macro agents using the Interagent Communication Language.

Micro Agents: are responsible for handling a single input or output data stream, either filtering the signal to or from a hierarchically superior 'interpret' agent.

The network architecture that we used was hierarchical at two resolutions: micro agents are connected to a superior macro agent, and macro agents are connected in turn to a facilitator agent. In both cases, a server is responsible for the supervision of its client sub-agents.

In order to describe our implementation, we will first give a description of each agent used in our application and then illustrate the flow of communication among agents produced by a user's request.

Speech Recognition (SR) Agent: The SR agent provides a mapping from the Interagent Communication Language to the API for the Decipher (Corona) speech recognition system (Cohen et al., 1990), a continuous speech speaker independent recognizer based on Hidden Markov Model technology. This macro agent is also responsible for supervising a child micro agent whose task is to control the speech data stream. The SR agent can provide feedback to an interface agent about the current status and progress of the micro agent (e.g. "listening", "end of speech detected", etc.) This agent is written in C.

Natural Language (NL) Parser Agent: translates English expressions into the Interagent Communication Language (ICL). For a more complete description of the ICL, see Cohen et al. (Cohen et al., 1994). The NL agent we selected for our application is the simplest of those integrated into the OAA. It is written in Prolog using Definite Clause Grammars, and supports a distributed vocabulary; each agent dynamically adds word definitions as it connects to the network. A current project is underway to integrate the Gemini natural language system (Cohen et al., 1990), a robust bottom up parser and semantic interpreter specifically designed for use in Spoken Language Understanding projects.

Database Agents: Database agents can reside at local or remote locations and can be grouped hierarchically according to content. Micro agents can be connected to database agents to monitor relevant positions or events in real time. In our travel planning application, database agents provide maps for each city, as well as icons, vocabulary and information about available hotels, restaurants, movies, theaters, municipal buildings and tourist attractions. Three types of databases were used: Prolog databases, X.500 hierarchical databases, and data loaded automatically by scanning HTML pages from the World Wide Web (WWW). In one instance, a local newspaper provides weekly updates to its Mosaic-accessible list of current movie times and reviews, as well as adding several new restaurant reviews to a growing collection; this information is extracted by an HTML reading database agent and made accessible to the agent architecture. Descriptions and addresses of new restaurants are presented to the user on request, and the user can choose to add them to the permanent database by specifying positional coordinates on the map (e.g. "add this new restaurant here"), information lacking in the WWW database.

Reference Resolution Agent: This agent is responsible for merging requests arriving in parallel from different modalities, and for controlling interactions between the user interface agent, database agents and modality agents. In this implementation, the reference resolution agent is domain specific: knowledge is encoded as to what actions must be performed to resolve each possible type of ICL request in its particular domain. For a given ICL logical form, the agent can verify argument types, supply default values, and resolve argument references. Some argument references are descriptive ("How far is it to the hotel on Emerson Street?"); in this case, a domain agent will try to resolve the definite reference by

sending database agent requests. Other references, particularly when contextual or deictic, are resolved by the user interface agent ("What are the rates for this hotel?"). Once arguments to a query have been resolved, this agent coordinates the actions and calculations necessary to produce the result of the request.

Interface Agent: This macro agent is responsible for managing what is currently being displayed to the user, and for accepting the user's multimodal input. The Interface Agent also coordinates client modality agents and resolves ambiguities among them: handwriting and gestures are interpreted locally by micro agents and combined with results from the speech recognition agent, running on a remote speech server. The handwriting micro-agent interfaces with the Microsoft PenWindows API and accesses a handwriting recognizer by CIC Corporation. The gesture micro-agent accesses recognition algorithms developed for TAPAGE.

An important task for the interface agent is to record which objects of each type are currently salient, in order to resolve contextual references such as "the hotel" or "where I was before." Deictic references are resolved by gestural or direct manipulation commands. If no such indication is currently specified, the user interface agent waits long enough to give the user an opportunity to supply the value, and then prompts the user for it.

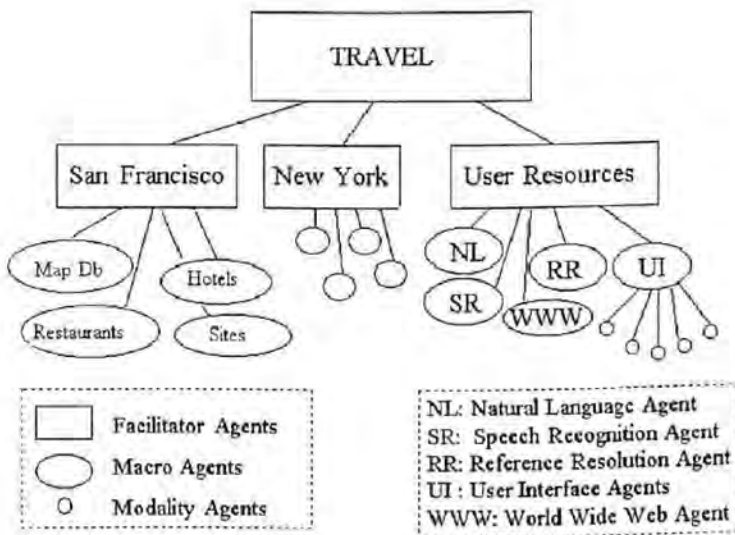


Fig. 3. Agent Architecture for Map Application

We shall now give an example of the distributed interaction of agents for a specific query. In the following example, all communication among agents passes

transparently through a facilitator agent in an undirected fashion; this process is left out of the description for brevity.

1. A user speaks: "How far is the restaurant from this hotel?"
2. The speech recognition agent monitors the status and results from its micro agent, sending feedback received by the user interface agent. When the string is recognized, a translation is requested.
3. The English request is received by the NL agent and translated into ICL form.
4. The reference resolution agent (RR) receives the ICL distance request containing one definite and one deictic reference and asks for resolution of these references.
5. The interface agent uses contextual structures to find what "the restaurant" refers to, and waits for the user to make a gesture indicating "the hotel", issuing prompts if necessary.
6. When the references have been resolved, the domain agent (RR) sends database requests asking for the coordinates of the items in question. It then calculates the distance according to the scale of the currently displayed map, and requests the user interface to produce output displaying the result of the calculation.

5 Conclusions

By augmenting an existing agent-based architecture with concepts necessary for synergistic multimodal input, we were able to rapidly develop a map-based application for a travel planning task. The resulting application has met our initial requirements: a mobile, synergistic pen/voice interface providing good natural language access to heterogeneous distributed knowledge sources. The approach used was general and should provide a for developing synergistic multimodal applications for other domains.

The system described here is one of the first that accepts commands made of synergistic combinations of spoken language, handwriting and gestural input. This fusion of modalities can produce more complex interactions than in many systems and the prototype application will serve as a testbed for acquiring a better understanding of multimodal input.

In the near future, we will continue to verify and extend our approach by building other multimodal applications. We are interested in generalizing the methodology even further; work has already begun on an agent-building tool which will simplify and automate many of the details of developing new agents and domains.

References

- Allegayer, J., Jansen-Winkel, R., Reddig, C. and Reithinger, N. (1989) Bidirectional use of knowledge in the multi-modal NL access system XTRA. In *Proceedings of IJCAI-89*, Detroit, pp. 1492-1497.