MULTIMODAL LANGUAGE PROCESSING

Michael Johnston

Center for Human-Computer Communication
Oregon Graduate Institute
johnston@cse.ogi.edu

ABSTRACT

Multimodal interfaces enable more natural and effective human-computer interaction by providing multiple channels through which input or output may pass. In order to realize their full potential, they need to support not just input from multiple modes, but synchronized integration of semantic content from different modes. This paper describes a multimodal language processing architecture which allows for declarative statement of multimodal integration strategies in a unification-based grammar formalism. The architecture is currently deployed in a working system enabling interaction with dynamic maps using speech and pen, but the approach is more general and supports a wide variety of other potential multimodal interfaces.

1. INTRODUCTION

Interaction between humans and machines is often limited by the restriction of communication to a single mode. Multimodal interfaces overcome this limitation by providing multiple channels for input and/or output. Our focus here is on multimodal input, specifically pen/voice interaction with dynamic maps. Recent empirical results (Oviatt 1996) have shown significant task performance and user preference advantages for multimodal interfaces in comparison to unimodal spoken interfaces for map-based tasks.

Multimodal interfaces pose significant challenges for natural language processing, which has typically been concerned with parsing and understanding of input in a single (spoken or typed) mode. How can natural language be parsed and understood when it is distributed across a number of input modes? How can grammars be defined so they can describe content realized in different modes? In this paper, I show how techniques and representations from natural language processing can be applied to the development of multimodal language processing capabilities.

The approach described supports pen/voice input to interactive maps as part of the QuickSet system (Cohen et al. 1997). Users interact with a map displayed on a portable wireless pen computer. They can draw directly on the map with a pen and simultaneously issue spoken commands. For example, in Figure 1 the user has just drawn an area and said 'FLOOD ZONE' in order to annotate the map with the position of a flood zone.

The multimodal language processing architecture is distributed and consists of a number of agents which communicate through an agent architecture (Cohen et al. 1994). Incoming speech signals and electronic ink received by the user interface client (Figure 1) are passed on to speech and gesture recognition agents respectively, each of which generates an N-best list of potential recognitions with associated probabilities. These are then assigned interpretations by natural language and gesture interpretation agents. These interpretations are then passed on to a multimodal integrator agent which finds potential multimodal combinations and selects the command to be executed. Our previous work on multimodal integration (Johnston et al. 1997) advocated the use of typed feature structures (Carpenter 1992) as a common meaning representation for speech and gesture. In that work, integration was modeled as a cross product unification of feature structures assigned to speech and gesture. While that approach overcomes many of the limitations of previous multimodal systems and it supports a broad and useful class of multimodal systems, it does not scale well to support multigesture utterances, complex unimodal gestures, or other modes and combinations of modes (Johnston 1998). In order to address these limitations we have developed an approach to multimodal integration which utilizes a multidimensional chart parser (Johnston 1998). This approach draws on work in visual parsing (Wittenburg et al. 1991). Elements of multimodal input are treated as terminal edges by the parser. They are combined together in accordance with a unification-based multimodal grammar.

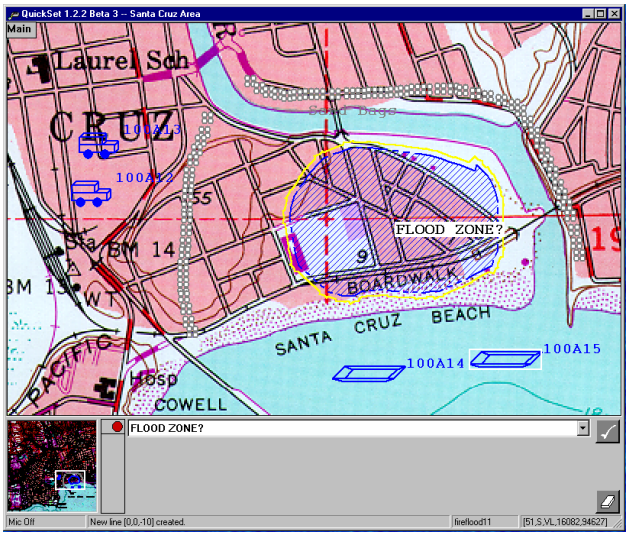

Figure 1: QuickSet user interface

2. UNIFICATION-BASED MULTIMODAL GRAMMARS

Our approach to the representation of multimodal grammars draws on unification-based approaches to syntax and semantics such as Head-driven phrase structure grammar (HPSG) (Pollard

and Sag 1994). Spoken phrases and pen gestures are assigned typed feature structures by the natural language and gesture interpretation agents respectively. For example, 'FLOOD ZONE ' is assigned the representation in Figure 2 and an area gesture the representation in Figure 3.

```
[ cat : located_command
  content : [ fsTYPE : create_area
              object : [ fsTYPE : area_obj
                         type : flood_zone ]
              location : [fsTYPE : area] ]
  modality : speech
  time : interval(...)
  prob : 0.67 ]
```

Figure 2: 'Flood zone' feature structure

```
[ cat : spatial_gesture
  content : [ fsTYPE : area
              coordlist : [latlon(..,..),...latlon(..,..)] ]
  modality : gesture
  time : interval(...)
  prob : 0.89 ]
```

Figure 3: Area gesture feature structure

In addition to providing a representation for the edges, typed feature structures are used to represent combination rules. These rules take the form of feature structure schemata. They describe the structure of multimodal utterances. The schema in Figure 4 describes the basic integration strategy for combination of speech and gesture. The basic, and only, integration strategy of Johnston et al 1997, is now just one rule among many. The rule states that a located spoken command (e.g. 'FLOOD ZONE') can combine with a spatial gesture (e.g. area), if the spoken command's location feature unifies with the semantic content of the gesture. The feature structure representation is augmented with functional constraints and these are used to further constraint integration. Constraints require certain spatial and temporal relationships to hold between combining edges. Complex constraints can be formulated using the logical operators $\wedge$, $\vee$, and $\Rightarrow$. The first constraint in Figure 4 requires that the time of the speech [7] must overlap or come within four seconds of the time of the gesture [10] (Oviatt et al 1997).

2.1 Multimodal Subcategorization

The rule schema outlined in the previous section enables simple combinations of speech and gesture. To handle more complex multimodal utterances such as cases where speech combines with several gestures (Figure 5) a subcategorization mechanism is employed.

```
[ lhs : [ cat : command
          content : [1]
          modality : [3]
          time : [4]
          prob : [5] ]
  rhs : [ dtr1 : [ cat : located_command
                   content : [1][location : [2]]
                   modality : [6]
                   time : [7]
                   prob : [8] ]
          dtr2 : [ cat : spatial_gesture
                   content : [2]
                   modality : [9]
                   time : [10]
                   prob : [11] ] ]
  constraints : { overlap([7],[10]) ∨ follow([7],[10],timeval(4,0))
                  total_time([7],[10],[4])
                  combine_prob([8],[11],[5])
                  assign_modality([6],[9],[3]) } ]
```

Figure 4: Basic multimodal integration rule schema

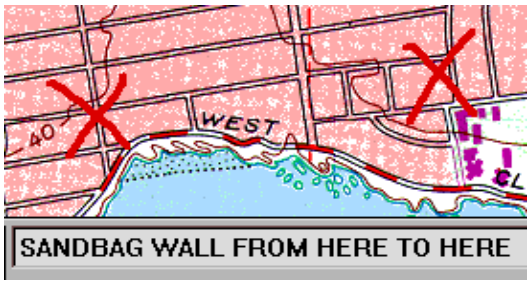

Figure 5: Multigesture utterance

This draws on lexicalist treatments of verb complementation such as that developed in HPSG. Just as a verb subcategorizes for a series of complements, an element of multimodal input can be thought of as subcategorizing for the gestures or other components which it needs to combine with. The spoken command 'SANDBAG WALL FROM HERE TO HERE' is assigned the representation in Figure 6. The list of subcategorized elements is encoded in a first/rest structure. This spoken phrase subcategorizes for two cross gestures, which provide the start and end of the wall respectively. This representation is processed by general combinatory rule schemata which combine edges with the elements they subcategorize for. The specific temporal constraints on combinations such as these cannot be specified in the general combination rules. Instead, these constraints are specified in a constraints: feature at each level of the first/rest structure. In this case, the first gesture needs to overlap or precede the speech by up to three seconds, and the second gesture is required to follow the first within five seconds.

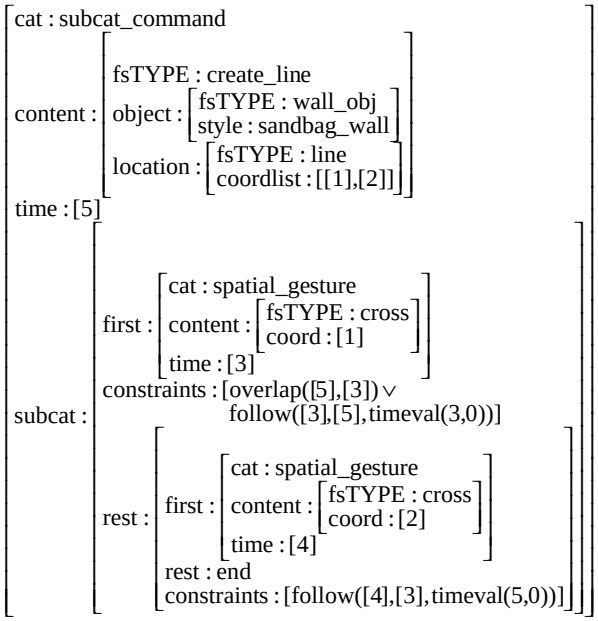

Figure 6: 'SANDBAG WALL FROM HERE TO HERE'

An important advantage of the use of a grammar for multimodal utterances is that grammars can be built up which allow the different parts of a command to be expressed in a variety of modes. For example, Figure 7 illustrates a unimodal gestural command for indicating a movement. The arrow specifies the extent of the move, while the times at the base and head specify the arrival and departure times.

Figure 7: Movement command

If the subcategorization associated with this command is specified at a semantic level it can support a range of multimodal and unimodal utterances. The arrow can subcategorize for two time specifications and allow them to be expressed in either speech or gesture. Spatial and temporal constraints can be still be placed on the combining elements through the use of conditional constraints. In this example case, if the times are expressed in gesture, conditional constraints will require the departure time to be close to the base of the arrow and the arrival time to be close to the head. This is achieved using constraints of the following form: *is([4],gesture)* ⇒ *close_to([5],[6])*.

2.2 Constructional Meaning

While many multimodal utterances are best described using subcategorization, others are better described as constructions; that is, specific rule schemata which assign a meaning to combining elements. For example, the rule schema in Figure 8 indicates that an area gesture containing and 'F" gesture can be interpreted as a command to create a flood zone annotation.

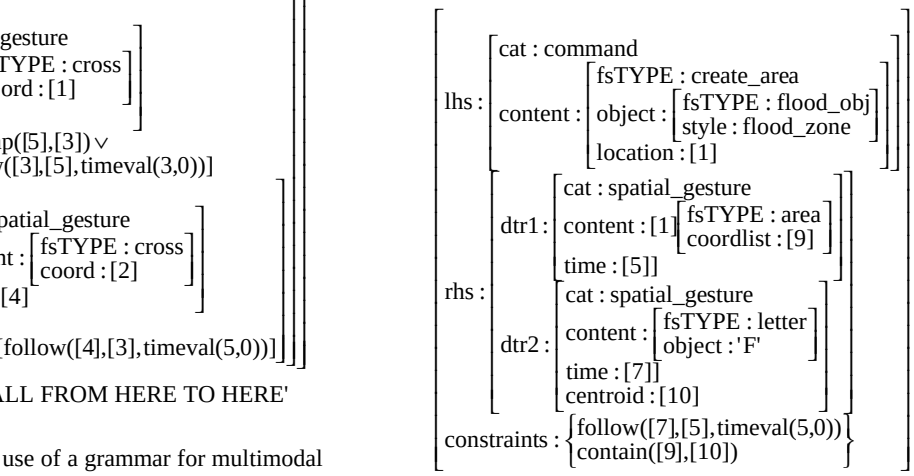

Figure 8: Unimodal flood zone construction

3. MULTIMODAL DISCOURSE

In addition to complex unimodal and multimodal commands, the chart and grammar representation are used to support more extended multimodal discourse. For example, in map-based tasks users frequently need to create a number of entities of the same type. This is achieved using *multiple* commands. For example, in order to indicate the locations of a number of flood zones, the user can say 'MULTIPLE FLOOD ZONES' and then draw any number of areas in succession. Each following gesture results in the creation of another area. Multiple commands are treated as persistent edges in the chart. 'MULTIPLE FLOOD ZONES' is assigned an interpretation much the same as 'FLOOD ZONE' but it is typed as a persistent edge and persistent edges are not removed from the chart when the resulting command is executed. Persistent edges are assigned a timeout feature which indicates how long they can persist. This time is pushed forward every time another gesture is drawn so that multiple commands can persist for as long as the user continues to draw further gestures.

When users are creating a number of different entities of the same type, those entities may be accompanied by further spoken phrases. For example, 'FLOOD ZONES HERE AND HERE' <area gesture> <area gesture> 'AND HERE' <area gesture>. In order to support commands which are distributed in this way certain commands are assigned what I will call *child edges*. These are edges which are released onto the chart as a result of a command being executed. The child edge is indicated in a child: feature. In the example in Figure 9 'FLOOD ZONES HERE AND HERE' subcategorizes for two area gestures. The child edge subcategorizes for two elements: a here command, such as 'HERE' or 'AND HERE', and further area gesture. If

the user says 'AND HERE' and draws another area within thirty seconds of 'FLOOD ZONES HERE AND HERE' then another area will be created. The child edge mechanism assigns the child feature to the child edge itself so it turn releases a child edge so the interaction can continue further.

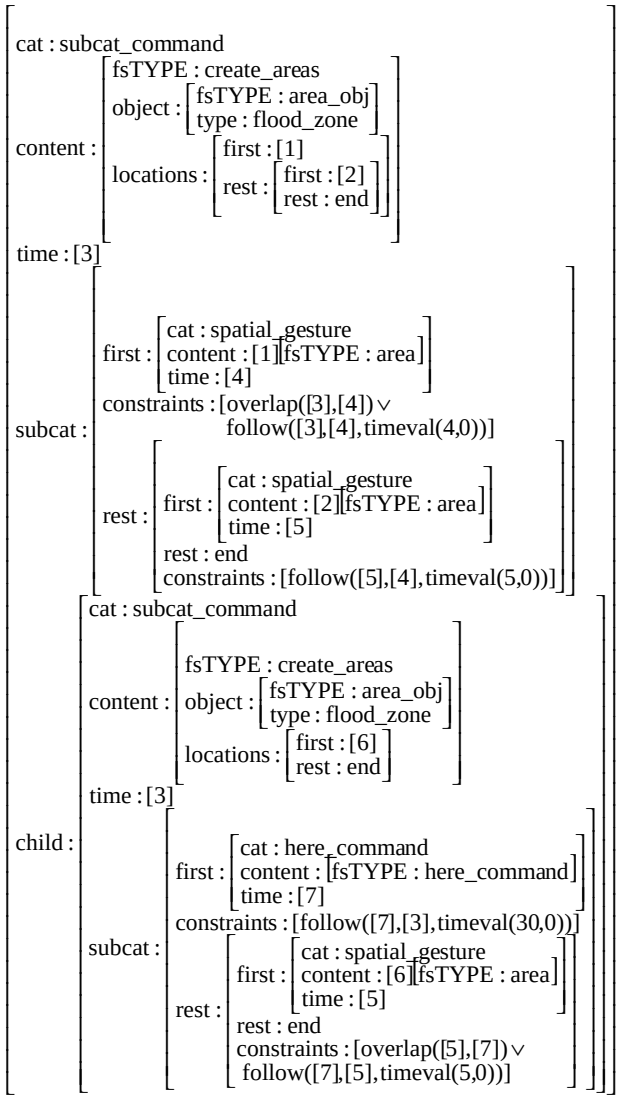

Figure 9: 'FLOOD ZONE HERE AND HERE'

4. CONCLUSION

The approach to multimodal language processing described here applies techniques from natural language processing to the support of next generation multimodal interfaces. Typed feature structures are used as common meaning representation for speech and gesture. Multimodal integration strategies and stated declaratively in a unification-based grammar formalism, which is used by a multidimensional multimodal parser to process and interpret multimodal input. A broad range of multimodal utterances are supported, including combination of speech with multiple gestures and complex unimodal gestures. The approach is sufficiently general to support alternative input modes and combinations of modes. Unlike speech-driven systems, such as that of Neal and Shapiro 1991 and others, this approach is *fully-multimodal* in that all elements of the content of a command can originate in either mode. The use of unification-based grammars facilitates integration of the approach with contemporary work in natural language processing, where feature structure formalisms are commonplace. Declarative statement of multimodal integration strategies enables rapid prototyping and iterative development of multimodal systems.

5. ACKNOWLEDGEMENTS

This research is supported by the Information Technology and Information Systems offices of DARPA under contract DABT63-95-C-007 and by the CHCC member consortium.

6. REFERENCES

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

2. 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.

3. Cohen, P. R., M. Johnston, D. McGee, S. L. Oviatt, J. A. Pittman, I. Smith, L. Chen, and J. Clow. 1997. "QuickSet: Multimodal interaction for distributed applications". In Proceedings of the Fifth ACM International Multimedia Conference. 31-40.

4. Johnston, M. 1998. "Unification-based Multimodal Parsing". In *Proceedings of the 17th International Conference on Computational Linguistics and the 36th Annual Meeting of the Association for Computational Linguistics*.

5. Neal, J. G., and S. C. Shapiro. 1991. "Intelligent multi-media interface technology". In J. W. Sullivan and S. W. Tyler (eds.) *Intelligent User Interfaces*, ACM Press, Addison Wesley, New York, 45-68.

6. Oviatt, S. L. 1996. "Multimodal interfaces for dynamic interactive maps," *Proceedings of CHI'96 Human Factors in Computing Systems*.

7. Oviatt, S. L., A. DeAngeli, and K. Kuhn. 1997. "Integration and synchronization of input modes during multimodal human-computer interaction". In *Proceedings of CHI'97*, 415-422.

8. Pollard, C. and I. Sag. 1994. *Head-driven phrase structure grammar*. University of Chicago Press.

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

10. Wittenburg, K., L. Weitzman, and J. Talley. 1991. "Unification-based grammars and tabular parsing for graphical languages". *Journal of Visual Languages and Computing* 2:347-370.