Unification-based Multimodal Parsing

Michael Johnston
Center for Human Computer Communication
Department of Computer Science and Engineering
Oregon Graduate Institute
P.O. Box 91000, Portland, OR 97291-1000
johnston@cse.ogi.edu

Abstract

In order to realize their full potential, multimodal systems need to support not just input from multiple modes, but also synchronized integration of modes. Johnston et al (1997) model this integration using a unification operation over typed feature structures. This is an effective solution for a broad class of systems, but limits multimodal utterances to combinations of a single spoken phrase with a single gesture. We show how the unification-based approach can be scaled up to provide a full multimodal grammar formalism. In conjunction with a multidimensional chart parser, this approach supports integration of multiple elements distributed across the spatial, temporal, and acoustic dimensions of multimodal interaction. Integration strategies are stated in a high level unification-based rule formalism supporting rapid prototyping and iterative development of multimodal systems.

1 Introduction

Multimodal interfaces enable more natural and efficient interaction between humans and machines by providing multiple channels through which input or output may pass. Our concern here is with multimodal input, such as interfaces which support simultaneous input from speech and pen. Such interfaces have clear task performance and user preference advantages over speech only interfaces, in particular for spatial tasks such as those involving maps (Oviatt 1996). Our focus here is on the integration of input from multiple modes and the role this plays in the segmentation and parsing of natural human input. In the examples given here, the modes are speech and pen, but the architecture described is more general in that it can support more than two input modes and modes of other types such as 3D gestural input.

Our multimodal interface technology is implemented in QuickSet (Cohen et al 1997), a working system which supports dynamic interaction with maps and other complex visual displays. The initial applications of QuickSet are: setting up and interacting with distributed simulations (Courtemanche and Cercanowicz 1995), logistics planning, and navigation in virtual worlds. The system is distributed; consisting of a series of agents (Figure 1) which communicate through a shared blackboard (Cohen et al 1994). It runs on both desktop and handheld PCs, communicating over wired and wireless LANs. The user interacts with a map displayed on a wireless hand-held unit (Figure 2).

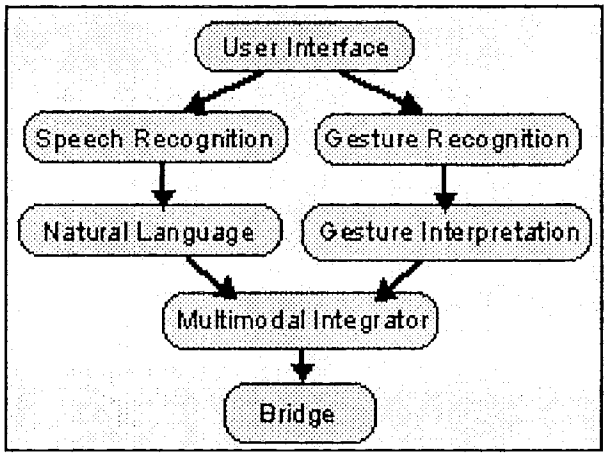

Figure 1: Multimodal Architecture

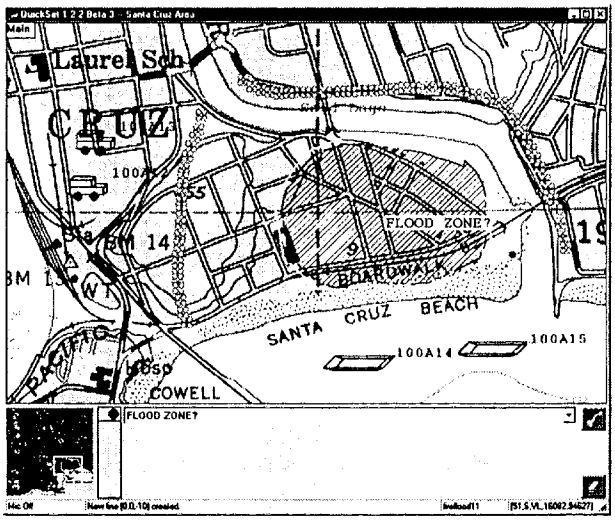

Figure 2: User Interface

They can draw directly on the map and simultaneously issue spoken commands. Different kinds of entities, lines, and areas may be created by drawing the appropriate spatial features and speaking their type; for example, drawing an area and saying '*flood zone*'. Orders may also be specified; for example, by drawing a line and saying '*helicopter follow this route*'. The speech signal is routed to an HMM-

based continuous speaker-independent recognizer. The electronic 'ink' is routed to a neural net-based gesture recognizer (Pittman 1991). Both generate N-best lists of potential recognition results with associated probabilities. These results are assigned semantic interpretations by natural language processing and gesture interpretation agents respectively. A multimodal integrator agent fields input from the natural language and gesture interpretation agents and selects the appropriate multimodal or unimodal commands to execute. These are passed on to a bridge agent which provides an API to the underlying applications the system is used to control.

In the approach to multimodal integration proposed by Johnston et al 1997, integration of spoken and gestural input is driven by a unification operation over typed feature structures (Carpenter 1992) representing the semantic contributions of the different modes. This approach overcomes the limitations of previous approaches in that it allows for a full range of gestural input beyond simple deictic pointing gestures. Unlike speech-driven systems (Bolt 1980, Neal and Shapiro 1991, Koons et al 1993, Wauchope 1994), it is *fully multimodal* in that all elements of the content of a command can be in either mode. Furthermore, compared to related frame-merging strategies (Vo and Wood 1996), it provides a well understood, generally applicable common meaning representation for the different modes and a formally well defined mechanism for multimodal integration. However, while this approach provides an efficient solution for a broad class of multimodal systems, there are significant limitations on the expressivity and generality of the approach.

A wide range of potential multimodal utterances fall outside the expressive potential of the previous architecture. Empirical studies of multimodal interaction (Oviatt 1996), utilizing wizard-of-oz techniques, have shown that when users are free to interact with any combination of speech and pen, a single spoken utterance maybe associated with more than one gesture. For example, a number of deictic pointing gestures may be associated with a single spoken utterance: '*calculate distance from here to here*', '*put that there*', '*move this team to here and prepare to rescue residents from this building*'. Speech may also be combined with a series of gestures of different types: the user circles a vehicle on the map, says '*follow this route*', and draws an arrow indicating the route to be followed.

In addition to more complex multipart multimodal utterances, unimodal gestural utterances may contain several component gestures which compose to yield a command. For example, to create an entity with a specific orientation, a user might draw the entity and then draw an arrow leading out from it (Figure 3 (a)). To specify a movement, the user might draw an arrow indicating the extent of the move and indicate departure and arrival times by writing expressions at the base and head (Figure 3 (b)). These

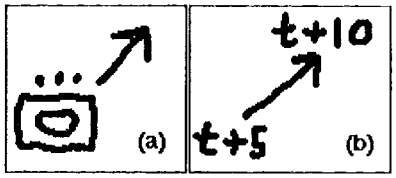

Figure 3: Complex Unimodal Gestures

are specific examples of the more general problem of visual parsing, which has been a focus of attention in research on visual programming and pen-based interfaces for the creation of complex graphical objects such as mathematical equations and flowcharts (Lakin 1986, Wittenburg et al 1991, Helm et al 1991, Crimi et al 1995).

The approach of Johnston et al 1997 also faces fundamental architectural problems. The multimodal integration strategy is hard-coded into the integration agent and there is no isolatable statement of the rules and constraints independent of the code itself. As the range of multimodal utterances supported is extended, it becomes essential that there be a declarative statement of the grammar of multimodal utterances, separate from the algorithms and mechanisms of parsing. This will enable system developers to describe integration strategies in a high level representation, facilitating rapid prototyping and iterative development of multimodal systems.

2 Parsing in Multidimensional Space

The integrator in Johnston et al 1997 does in essence parse input, but the resulting structures can only be unary or binary trees one level deep; unimodal spoken or gestural commands and multimodal combinations consisting of a single spoken element and a single gesture. In order to account for a broader range of multimodal expressions, a more general parsing mechanism is needed.

Chart parsing methods have proven effective for parsing strings and are commonplace in natural language processing (Kay 1980). Chart parsing involves population of a triangular matrix of well-formed constituents: $chart(i, j)$, where i and j are numbered vertices delimiting the start and end of the string. In its most basic formulation, chart parsing can be defined as follows, where $*$ is an operator which combines two constituents in accordance with the rules of the grammar.

$$chart(i,j) = \bigcup_{i<k<j} chart(i,k) * chart(k,j)$$

Crucially, this requires the combining constituents to be discrete and linearly ordered. However, multimodal input does not meet these requirements:

gestural input spans two (or three) spatial dimensions, there is an additional non-spatial acoustic dimension of speech, and both gesture and speech are distributed across the temporal dimension. Unlike words in a string, speech and gesture may overlap temporally, and there is no single dimension on which the input is linear and discrete. So then, how can we parse in this multidimensional space of speech and gesture? What is the rule for chart parsing in multi-dimensional space? Our formulation of multidimensional parsing for multimodal systems ($multichart$) is as follows.

$$multichart(X) = \cup\, multichart(Y) * multichart(Z)$$
$$where\ X = Y \cup Z, Y \cap Z = \emptyset, Y \neq \emptyset, Z \neq \emptyset$$

In place of numerical spans within a single dimension (e.g. $chart(3,5)$), edges in the multidimensional chart are identified by sets (e.g. $multichart(\{[s,4,2],[g,6,1]\})$) containing the identifiers(IDs) of the terminal input elements they contain. When two edges combine, the ID of the resulting edge is the union of their IDs. One constraint that linearity enforced, which we can still maintain, is that a given piece of input can only be used once within a single parse. This is captured by a requirement of non-intersection between the ID sets associated with edges being combined. This requirement is especially important since a single piece of spoken or gestural input may have multiple interpretations available in the chart. To prevent multiple interpretations of a single signal being used, they are assigned IDs which are identical with respect to the the non-intersection constraint. The multichart statement enumerates all the possible combinations that need to be considered given a set of inputs whose IDs are contained in a set X.

The multidimensional parsing algorithm (Figure 4) runs bottom-up from the input elements, building progressively larger constituents in accordance with the ruleset. An agenda is used to store edges to be processed. As a simplifying assumption, rules are assumed to be binary. It is straightforward to extend the approach to allow for non-binary rules using techniques from active chart parsing (Earley 1970), but this step is of limited value given the availability of multimodal subcategorization (Section 4).

```
while AGENDA ≠ [ ] do
    remove front edge from AGENDA
      and make it CURRENTEDGE
    for each EDGE, EDGE ∈ CHART
      if CURRENTEDGE ∩ EDGE = ∅
        find set NEWEDGES = ∪ (
            (∪ CURRENTEDGE * EDGE)
            (∪ EDGE * CURRENTEDGE))
        add NEWEDGES to end of AGENDA
    add CURRENTEDGE to CHART
```

Figure 4: Multichart Parsing Algorithm

For use in a multimodal interface, the multidimensional parsing algorithm needs to be embedded into the integration agent in such a way that input can be processed incrementally. Each new input received is handled as follows. First, to avoid unnecessary computation, stale edges are removed from the chart. A **timeout** feature indicates the shelf-life of an edge within the chart. Second, the interpretations of the new input are treated as terminal edges, placed on the agenda, and combined with edges in the chart in accordance with the algorithm above. Third, complete edges are identified and executed. Unlike the typical case in string parsing, the goal is not to find a single parse covering the whole chart; the chart may contain several complete non-overlapping edges which can be executed. These are assigned to a category *command* as described in the next section. The complete edges are ranked with respect to probability. These probabilities are a function of the recognition probabilities of the elements which make up the command. The combination of probabilities is specified using declarative constraints, as described in the next section. The most probable complete edge is executed first, and all edges it intersects with are removed from the chart. The next most probable complete edge remaining is then executed and the procedure continues until there are no complete edges left in the chart. This means that selection of higher probability complete edges eliminates overlapping complete edges of lower probability from the list of edges to be executed. Lastly, the new chart is stored. In ongoing work, we are exploring the introduction of other factors to the selection process. For example, sets of disjoint complete edges which parse all of the terminal edges in the chart should likely be preferred over those that do not.

Under certain circumstances, an edge can be used more than once. This capability supports multiple creation of entities. For example, the user can utter '*multiple helicopters*' *point point point point* in order to create a series of vehicles. This significantly speeds up the creation process and limits reliance on speech recognition. Multiple commands are persistent edges; they are not removed from the chart after they have participated in the formation of an executable command. They are assigned timeouts and are removed when their alloted time runs out. These 'self-destruct' timers are zeroed each time another entity is created, allowing creations to chain together.

3 Unification-based Multimodal Grammar Representation

Our grammar representation for multimodal expressions draws on unification-based approaches to syntax and semantics (Shieber 1986) such as Head-

driven phrase structure grammar (HPSG) (Pollard and Sag 1987,1994). Spoken phrases and pen gestures, which are the terminal elements of the multimodal parsing process, are referred to as *lexical edges*. They are assigned grammatical representations in the form of typed feature structures by the natural language and gesture interpretation agents respectively. For example, the spoken phrase '*helicopter*' is assigned the representation in Figure 5.

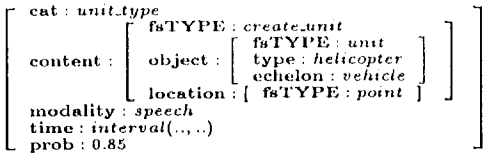

Figure 5: Spoken Input Edge

The **cat** feature indicates the basic category of the element, while **content** specifies the semantic content. In this case, it is a *create_unit* command in which the object to be created is a vehicle of type helicopter, and the location is required to be a point. The remaining features specify auxiliary information such as the modality, temporal interval, and probability associated with the edge. A point gesture has the representation in Figure 6.

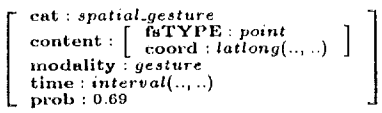

Figure 6: Point Gesture Edge

Multimodal grammar rules are productions of the form $LHS \rightarrow DTR1\ DTR2$ where LHS, $DTR1$, and $DTR2$ are feature structures of the form indicated above. Following HPSG, these are encoded as feature structure rule schemata. One advantage of this is that rule schemata can be hierarchically ordered, allowing for specific rules to inherit basic constraints from general rule schemata. The basic multimodal integration strategy of Johnston et al 1997 is now just one rule among many (Figure 7).

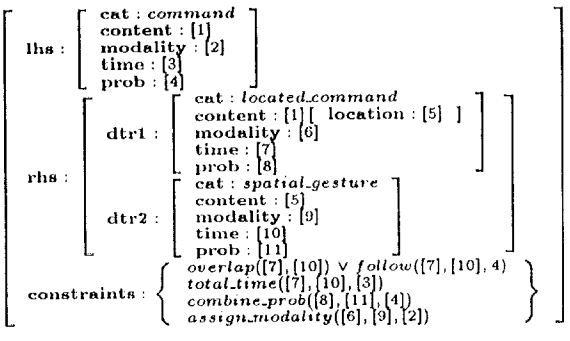

Figure 7: Basic Integration Rule Schema

The **lhs**,**dtr1**, and **dtr2** features correspond to LHS, $DTR1$, and $DTR2$ in the rule above. The **constraints** feature indicates an ordered series of constraints which must be satisfied in order for the rule to apply. Structure-sharing in the rule representation is used to impose constraints on the input feature structures, to construct the LHS category, and to instantiate the variables in the constraints. For example, in Figure 7, the basic constraint that the **location** of a located command such as '*helicopter*' needs to unify with the **content** of the gesture it combines with is captured by the structure-sharing tag [5]. This also instantiates the **location** of the resulting edge, whose **content** is inherited through tag [1].

The application of a rule involves unifying the two candidate edges for combination against **dtr1** and **dtr2**. Rules are indexed by their **cat** feature in order to avoid unnecessary unification. If the edges unify with **dtr1** and **dtr2**, then the constraints are checked. If they are satisfied then a new edge is created whose category is the value of **lhs** and whose ID set consists of the union of the ID sets assigned to the two input edges.

Constraints require certain temporal and spatial relationships to hold between edges. Complex constraints can be formed using the basic logical operators $\vee$, $\wedge$, and $\Rightarrow$. The temporal constraint in Figure 7, $overlap([7],[10]) \vee follow([7],[10],4)$, states that the time of the speech [7] must either overlap with or start within four seconds of the time of the gesture [10]. This temporal constraint is based on empirical investigation of multimodal interaction (Oviatt et al 1997). Spatial constraints are used for combinations of gestural inputs. For example, $close\_to(X,Y)$ requires two gestures to be a limited distance apart (See Figure 12 below) and $contact(X,Y)$ determines whether the regions occupied by two objects are in contact. The remaining constraints in Figure 7 do not constrain the inputs per se, rather they are used to calculate the **time**, **prob**, and **modality** features for the resulting edge. For example, the constraint $combine\_prob([8],[11],[4])$ is used to combine the probabilities of two inputs and assign a joint probability to the resulting edge. In this case, the input probabilities are multiplied. The $assign\_modality([6],[9],[2])$ constraint determines the modality of the resulting edge. Auxiliary features and constraints which are not directly relevant to the discussion will be omitted.

The constraints are interpreted using a prolog meta-interpreter. This basic back-tracking constraint satisfaction strategy is simplistic but adequate for current purposes. It could readily be substituted with a more sophisticated constraint solving strategy allowing for more interaction among constraints, default constraints, optimization among a series of constraints, and so on. The addition of functional constraints is common in HPSG and other unification grammar formalisms (Wittenburg 1993).

4 Multimodal Subcategorization

Given that multimodal grammar rules are required to be binary, how can the wide variety of commands in which speech combines with more than one gestural element be accounted for? The solution to this problem draws on the lexicalist treatment of complementation in HPSG. HPSG utilizes a sophisticated theory of subcategorization to account for the different complementation patterns that verbs and other lexical items require. Just as a verb subcategorizes for its complements, we can think of a lexical edge in the multimodal grammar as subcategorizing for the edges with which it needs to combine. For example, spoken inputs such as '*calculate distance from here to here*' and '*sandbag wall from here to here*' (Figure 8) result in edges which subcategorize for two gestures. Their multimodal subcategorization is specified in a list valued **subcat** feature, implemented using a recursive **first/rest** feature structure (Shieber 1986:27-32).

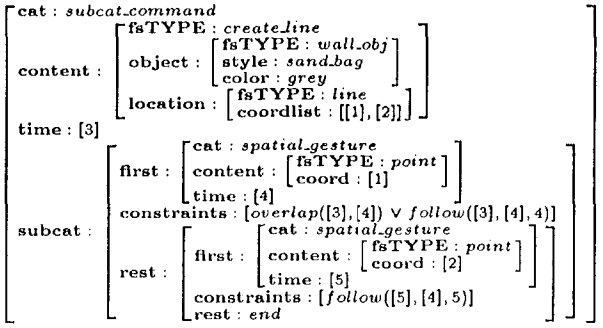

Figure 8: 'Sandbag wall from here to here'

The **cat** feature is *subcat_command*, indicating that this is an edge with an unsaturated subcategorization list. The **first/rest** structure indicates the two gestures the edge needs to combine with and terminates with **rest**: *end*. The temporal constraints on expressions such as these are specific to the expressions themselves and cannot be specified in the rule constraints. To support this, we allow for lexical edges to carry their own specific *lexical constraints*, which are held in a **constraints** feature at each level in the **subcat** list. In this case, the first gesture is constrained to overlap with the speech or come up to four seconds before it and the second gesture is required to follow the first gesture. Lexical constraints are inherited into the rule constraints in the combinatory schemata described below. Edges with **subcat** features are combined with other elements in the chart in accordance with general combinatory schemata. The first (Figure 9) applies to unsaturated edges which have more than one element on their **subcat** list. It unifies the first element of the **subcat** list with an element in the chart and builds a new edge of category *subcat_command* whose **subcat** list is the value of **rest**.

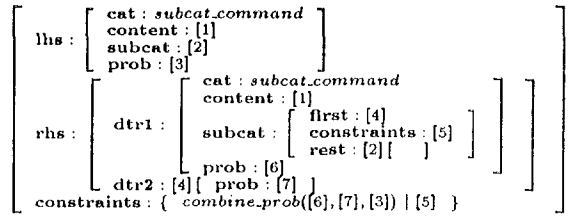

Figure 9: Subcat Combination Schema

The second schema (Figure 10) applies to unsaturated (**cat**: *subcat_command*) edges on whose **subcat** list only one element remains and generates saturated (**cat**: *command*) edges.

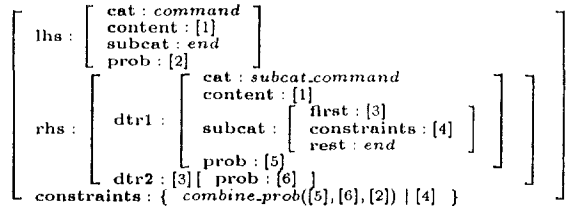

Figure 10: Subcat Termination Schema

This specification of combinatory information in the lexical edges constitutes a shift from rules to representations. The ruleset is simplified to a set of general schemata, and the lexical representation is extended to express combinatorics. However, there is still a need for rules beyond these general schemata in order to account for constructional meaning (Goldberg 1995) in multimodal input, specifically with respect to complex unimodal gestures.

5 Visual Parsing: Complex Gestures

In addition to combinations of speech with more than one gesture, the architecture supports unimodal gestural commands consisting of several independently recognized gestural components. For example, lines may be created using what we term *gestural diacritics*. If environmental noise or other factors make speaking the type of a line infeasible, it may be specified by drawing a simple gestural mark or word over a line gesture. To create a barbed wire, the user can draw a line specifying its spatial extent and then draw an alpha to indicate its type.

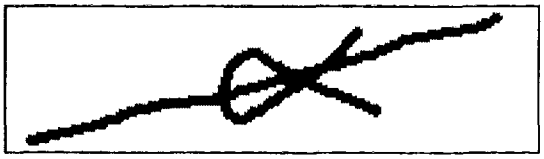

Figure 11: Complex Gesture for Barbed Wire

This gestural construction is licensed by the rule schema in Figure 12. It states that a line gesture

(**dtr1**) and an alpha gesture (**dtr2**) can be combined, resulting in a command to create a barbed wire. The location information is inherited from the line gesture. There is nothing inherent about alpha that makes it mean 'barbed wire'. That meaning is embodied only in its construction with a line gesture, which is captured in the rule schema. The *close_to* constraint requires that the centroid of the alpha be in proximity to the line.

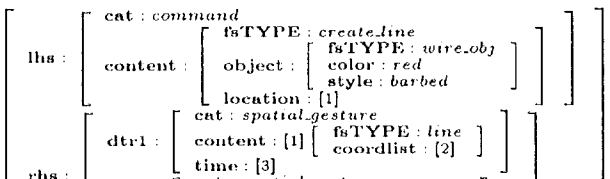

Figure 12: Rule Schema for Unimodal Barbed Wire

6 Conclusion

The multimodal language processing architecture presented here enables parsing and interpretation of natural human input distributed across two or three spatial dimensions, time, and the acoustic dimension of speech. Multimodal integration strategies are stated declaratively in a unification-based grammar formalism which is interpreted by an incremental multidimensional parser. We have shown how this architecture supports multimodal (pen/voice) interfaces to dynamic maps. It has been implemented and deployed as part of QuickSet (Cohen et al 1997) and operates in real time. A broad range of multimodal utterances are supported including combination of speech with multiple gestures and visual parsing of collections of gestures into complex unimodal commands. Combinatory information and constraints may be stated either in the lexical edges or in the rule schemata, allowing individual phenomena to be described in the way that best suits their nature. The architecture is sufficiently general to support other input modes and devices including 3D gestural input. The declarative statement of multimodal integration strategies enables rapid prototyping and iterative development of multimodal systems.

The system has undergone a form of pro-active evaluation in that its design is informed by detailed predictive modeling of how users interact multimodally, and incorporates the results of empirical studies of multimodal interaction (Oviatt 1996, Oviatt et al 1997). It is currently undergoing extensive user testing and evaluation (McGee et al 1998).

Previous work on grammars and parsing for multidimensional languages has focused on two dimensional graphical expressions such as mathematical equations, flowcharts, and visual programming languages. Lakin (1986) lays out many of the initial issues in parsing for two-dimensional drawings and utilizes specialized parsers implemented in LISP to parse specific graphical languages. Helm et al (1991) employ a grammatical framework, *constrained set grammars*, in which constituent structure rules are augmented with spatial constraints. Visual language parsers are build by translation of these rules into a constraint logic programming language. Crimi et al (1991) utilize a similar *relation grammar* formalism in which a sentence consists of a multiset of objects and relations among them. Their rules are also augmented with constraints and parsing is provided by a prolog axiomatization. Wittenburg et al (1991) employ a unification-based grammar formalism augmented with functional constraints (F-PATR, Wittenburg 1993), and a bottom-up, incremental, Earley-style (Earley 1970) tabular parsing algorithm.

All of these approaches face significant difficulties in terms of computational complexity. At worst, an exponential number of combinations of the input elements need to be considered, and the parse table may be of exponential size (Wittenburg et al 1991:365). Efficiency concerns drive Helm et al (1991:111) to adopt a *committed choice strategy* under which successfully applied productions cannot be backtracked over and complex negative and quantificational constraints are used to limit rule application. Wittenburg et al's parsing mechanism is directed by *expander relations* in the grammar formalism which filter out inappropriate combinations before they are considered. Wittenburg (1996) addresses the complexity issue by adding top-down predictive information to the parsing process.

This work is fundamentally different from all of these approaches in that it focuses on multimodal systems, and this has significant implications in terms of computational viability. The task differs greatly from parsing of mathematical equations, flowcharts, and other complex graphical expressions in that the number of elements to be parsed is far smaller. Empirical investigation (Oviatt 1996, Oviatt et al 1997) has shown that multimodal utterances rarely contain more than two or three elements. Each of those elements may have multiple interpretations, but the overall number of lexical edges remains sufficiently small to enable fast processing of all the potential combinations. Also, the intersection constraint on combining edges limits the impact of the multiple interpretations of each piece of input. The deployment of this architecture in an implemented system supporting real time spoken and gestural interaction with a dynamic map provides evidence of its computational viability for real tasks. Our approach is similar to Wittenburg et

al 1991 in its use of a unification-based grammar formalism augmented with functional constraints and a chart parser adapted for multidimensional spaces. Our approach differs in that, given the nature of the input, using spatial constraints and top-down predictive information to guide the parse is less of a concern, and as a result the parsing algorithm is significantly more straightforward and general.

The evolution of multimodal systems is following a trajectory which has parallels in the history of syntactic parsing. Initial approaches to multimodal integration were largely algorithmic in nature. The next stage is the formulation of declarative integration rules (phrase structure rules), then comes a shift from rules to representations (lexicalism, categorial and unification-based grammars). The approach outlined here is at representational stage, although rule schemata are still used for constructional meaning. The next phase, which syntax is undergoing, is the compilation of rules and representations back into fast, low-powered finite state devices (Roche and Schabes 1997). At this early stage in the development of multimodal systems, we need a high degree of flexibility. In the future, once it is clearer what needs to be accounted for, the next step will be to explore compilation of multimodal grammars into lower power devices.

Our primary areas of future research include refinement of the probability combination scheme for multimodal utterances, exploration of alternative constraint solving strategies, multiple inheritance for rule schemata, maintenance of multimodal dialogue history, and experimentation with 3D input and other combinations of modes.

References

Bolt, R. A. 1980. "Put-That-There":Voice and gesture at the graphics interface. *Computer Graphics*, 14.3:262–270.

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

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*, 1–8.

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.

Courtemanche, A. J., and A. Ceranowicz. 1995. ModSAF development status. In *Proceedings of the 5th Conference on Computer Generated Forces and Behavioral Representation*, 3–13.

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.

Earley, J. 1970. An efficient context-free parsing algorithm. *Communications of the ACM*, 13, 94–102.

Goldberg, A. 1995. *Constructions: A Construction Grammar Approach to Argument Structure.* University of Chicago Press, Chicago.

Helm, R., K. Marriott, and M. Odersky. 1991. Building visual language parsers. In *Proceedings of Conference on Human Factors in Computing Systems: CHI '91*, ACM Press, New York, 105–112.

Johnston, M., P. R. Cohen, D. McGee, S. L. Oviatt, J. A. Pittman, and 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*, 281-288.

Kay, M. 1980. Algorithm schemata and data structures in syntactic processing. In B. J. Grosz, K. S. Jones, and B. L. Webber (eds.) *Readings in Natural Language Processing*, Morgan Kaufmann, 1986, 35–70.

Koons, D. B., C. J. Sparrell, and K. R. Thorisson. 1993. Integrating simultaneous input from speech, gaze, and hand gestures. In M. T. Maybury (ed.) *Intelligent Multimedia Interfaces*, MIT Press, 257–276.

Lakin, F. 1986. Spatial parsing for visual languages. In S. K. Chang, T. Ichikawa, and P. A. Ligomenides (ed.s), *Visual Languages*. Plenum Press, 35–85.

McGee, D., P. R. Cohen, S. L. Oviatt. 1998. Confirmation in multimodal systems. In *Proceedings of 17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics.*

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

Oviatt, S. L. 1996. Multimodal interfaces for dynamic interactive maps. In *Proceedings of Conference on Human Factors in Computing Systems*, 95–102.

Oviatt, S. L., A. DeAngeli, and K. Kuhn. 1997. Integration and synchronization of input modes during multimodal human-computer interaction. In *Proceedings of Conference on Human Factors in Computing Systems*, 415–422.

Pittman, J. A. 1991. Recognizing handwritten text. In *Proceedings of Conference on Human Factors in Computing Systems: CHI '91.* 271–275.

Pollard, C. J., and I. A. Sag. 1987. *Information-based syntax and semantics: Volume I, Fundamentals.*, CSLI Lecture Notes Volume 13. CSLI, Stanford.

Pollard, Carl and Ivan Sag. 1994. *Head-driven phrase structure grammar.* University of Chicago Press. Chicago.

Roche, E. and Y. Schabes. 1997. *Finite state language processing.* MIT Press, Cambridge.

Shieber, S. M. 1986. *An Introduction to unification-based approaches to grammar.* CSLI Lecture Notes Volume 4. CSLI, Stanford.

Vo, M. T., and C. Wood. 1996. Building an application framework for speech and pen input integration in multimodal learning interfaces. In *Proceedings of ICASSP'96.*

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

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.

Wittenburg, K. L. 1993. F-PATR: Functional constraints for unification-based grammars. *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, 216–223.

Wittenburg, K. 1996. Predictive parsing for unordered relational languages. In H. Bunt and M. Tomita (eds.), *Recent Advances in Parsing Technologies*, Kluwer, Dordrecht, 385–407.