UNDERSTANDING MULTIMODAL INTERACTION BY EXPLOITING UNIFICATION AND INTEGRATION RULES

Håkan Johansson
Department of Computer and Information Science
Linköping University
Sweden
g_hakjo@ida.liu.se

Abstract

This paper presents a model for synergistic integration of multimodal speech and pen information. The model consists of an algorithm for matching and integrating interpretations of inputs from different modalities, as well as of a grammar that constrains integration. Integration proper is achieved by unifying feature structures. The integrator is part of a general framework for multimodal information systems with dialogue capabilities. Those parts of this framework that are relevant and affects the design of the integrator are also presented.

1. Introduction

In recent years, a number of studies have shown that interfaces that allow interaction through more than a single modality (e.g., speech) can empower users in their day-to-day interaction with computers (for a good up-to-date review, consult Oviatt et al., 2000). Carefully designed multimodal interfaces promise to make human-computer interaction more flexible, efficient, habitable, and natural. This is of most importance when it comes to walk-up-and-use systems, such as information kiosks. These are systems with which users do not interact on an everyday basis and therefore need to be designed to allow an intuitive interaction. An instance of such a system is a time table information system for the local bus and train transportations in a city and its surroundings. This kind of system is currently being developed at the Natural Language Laboratory (NLPLAB) at Linköping University.

The project aims to develop a publicly available time table information system capable of synergistic multimodal speech and pen interaction. In parallel to the development of the specific application, a general framework for multimodal information systems with dialogue capabilities, called MALIN, is set up. MALIN is an elaboration of an earlier architecture for unimodal typed natural language dialogue systems, and therefore this paper describes how the interpretation module has been expanded to handle multimodal speech and pen interaction. The primary focus of this paper is on how the problem of integrating, or fusing, the information received from the speech and pen modalities has been solved. This problem has previously been addressed by a number of researchers (cf. Neal et al. 1989; Wahlster, 1991; Koons, Sparrell, and Thorisson, 1993; Nigay and Coutaz, 1995; Johnston et al., 1997; Johnston, 1998; Johnston and Bangalore, 2000), but no single technique has become standard or even widely reused. This paper presents an approach to multimodal integration that falls somewhere between two of the earlier approaches, namely that of Johnston et al. (1997) and that of Johnston (1998). The integrator proposed consists of an algorithm for matching and integrating interpretations of input from different modalities, as well as an grammar formalism that constrains integration. The integration proper is performed by unification of feature structures.

2. MALIN

MALIN is an acronym for Multimodal Application of LINLIN. LINLIN, in turn, is an acronym for Linköping Natural Language Interface and is a general architecture for natural language interfaces capable of entertaining a coherent dialogue (Jönsson, 1997). MALIN is an extension of LINLIN and presents a general framework for multimodal dialogue applications. The framework consists of modules for interpreting and generating multimodal input and output, dialogue management, domain knowledge management, and user interface management (Dahlbäck et al., 1999). In the present paper, the focus is on the multimodal interpretation module.

The interpretation module consists of five separate parts: a speech recognizer, a natural language interpreter, a gesture recognizer, a gesture interpreter, and a multimodal integrator. The composition of these parts are illustrated in figure 1 below:

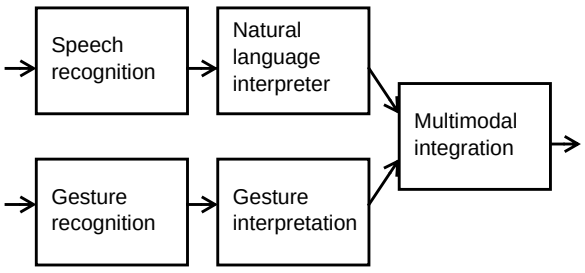

Figure 1: The interpretation module in the MALIN framework.

The architectural composition illustrated in figure 1 deviates very little from the typical multimodal interpre-

tation architecture discussed in Oviatt et al. (2000, p. 275). The only discrepancy between the two architectures is that the one shown above does not regard the dialogue context when integrating multimodal information. More specifically, integration, as proposed in the present paper, only occurs locally within a single interactional segment. References that remain unresolved after multimodal integration (e.g., references to previous dialogue contributions) are solved by the dialogue manager in the MALIN framework.

2.1 Speech recognizer

The speech recognizer, which is currently not available, is assumed to be a standard continuous, speaker-independent recognizer for Swedish. However, it is essential that the recognizer can provide information on the temporal onset and offset of the individual tokens in a recognized string. For example, if recognition results in the string ”depart from here”, the recognizer must provide temporal information about when ’depart’, ’from’, and ’here’ were uttered respectively. Currently, such speech recognizers exist for other languages than Swedish, and the lack of such a recognizer is a practical problem rather than a theoretical one.

2.2 Natural language interpreter

The natural language interpreter combines shallow and partial parsing (Strömbäck and Jönsson, 1998), which leads to a degree of flexibility suitable for spoken natural language. The interpreter is based on an extended PATR-II formalism. It has been extended to allow the occurrence of unknown words within phrases in order to avoid analysis to break down in the face of words that are not present in the lexicon. Furthermore, the parser environment has been extended to allow the application developer to specify which of the inactive edges that constitute the parse result, i.e., the result is a set of partial parses. The representational format is directed acyclic graph (DAG) representations of feature structures.

The temporal information provided by the speech recognizer is incorporated into the resulting partial parses. It is therefore required that the grammar is defined in such a way that all the information extracted from a single word or subphrase is represented within a single partial parse, rather than being distributed over several different partial parses. This is a limitation that has not been an issue in the current application, but could prove to be a problem in other applications. However, it is in many cases possible to construct the rules in the grammar such that this problem can be avoided.

2.3 Gesture recognizer

The gesture recognizer is triggered when the pen is pressed to the surface of the screen and terminates recognition when the pen is released from the screen. The recognition result is represented as a feature structure containing information on the ”touched” coordinates, as well as the onset and offset of a number of consecutive pen gestures. The only task the gesture recognizer has is to convert a continuous stream of information in a symbolic, non-interpreted, representation.

2.4 Gesture interpreter

The gesture interpreter further process the feature structure received from the gesture recognizer in that it receives an interpretation according to a specification of a gesture language. Different interpretations are assigned depending on where in the interface the gesture was made. The output of the gesture interpreter is a DAG.

3. Multimodal Integrator

As can be inferred from the previous sections, the multimodal integrator receives as input two feature structures, or more specifically, two DAG:s. The integrator process the DAG:s and attempts to integrate them in order to construct a coherent interpretation of a user’s actions.

3.1 Algorithm

The algorithm is quite straightforward. Informally, the algorithm maps every subDAG in one modality (i.e., the interpretation of input from one modality) to every subDAG in the other modality. In other words, the algorithm attempts to integrate every combination of subDAG:s. This means that the algorithm has to consider $S * G$ combinations, where S is the number of subDAG:s of the interpreted speech, and G is the number of subDAG:s in the interpreted gesture. In order for the combinations of subDAG:s to be integrated they have to match some rule in an ordered set of rules. These rules are ordered with regards to specificity, and therefore this is a case of conflict resolution by specificity (cf. Jackson, 1999, p. 86).

A few other points are worth noticing. First, what is considered for integration is subDAG:s along with the feature (or attribute) of which they are a value. Since we are considering only proper subDAG:s, every subDAG is the value of some feature. However, in the current natural language interpreter, cardinal numbers are used as features to iterate a number of similar structures. For instance, if the user specifies multiple locations, these are located by iteration under a feature *locations*. In the case where a subDAG is a value of a feature that is a cardinal number, the algorithm replaces this number with the feature name at the next shallower level.

Another point worth noticing is that in the current implementation, speech is considered to be the primary modality. This means that if the algorithm receives non-empty DAG:s but is unable to integrate them, the speech-DAG is returned. However, if the speech-DAG is empty and the gesture-DAG is not, the latter is returned.

3.2 Integration rules

As previously mentioned, the integration rules are an ordered set where each rule is a set of constraints on the combination of DAG:s to be integrated. For a rule to be applicable, all constraints must be satisfied. The con-

straints regard both the semantic and temporal information represented in the DAG:s.

3.3 Temporal constraints

When interacting with computers multimodally, users utilize a wide range of temporal integration patterns (cf. Oviatt et al., 1997). Therefore, the integration rules consists of constraints on these patterns. These constraints are placed on Δt_{onset} and Δt_{offset}. Δt_{onset} is calculated by subtracting the onset of gesture from the onset of speech. Δt_{offset} is similarly calculated by subtracting the offset of gesture form the offset of speech. The constraints are set by placing a maximum and a minimum value of Δt_{onset} and Δt_{offset} respectively, i.e., by setting an interval within which the values must range.

3.4 Semantic constraints

The semantics of a subDAG is taken to be the attribute of which it is a value (henceforth referred to as parent attribute) along with the attributes within the subDAG. The semantic constraints differs from the temporal in that they are optional. If the semantic constraints are left out, the integration process will be driven entirely by temporal information. This can work for some cases, but the risk is that absurd integrations will take place. For instance, if the user points to a location in the map while uttering a timepoint this information might be integrated.

The parent attribute is constrained by declaring which literal string it must be equal to. Constraints on the attributes contained within the subDAG on the other hand is more complex. These constraints can have the following three forms:

- <*attribute*> = *w*
- <*attribute*> = []
- <*attribute*> = ε

The symbol *w* denotes some string, [] denotes an empty dag, and ε denotes that no constraint is placed on the value (i.e., one only states that the attribute in question should be present).

3.5 Summary of constraints

The list below summarizes the different constraints that can be placed on the subDAG:s being compared:

- constraints on the intervals within which Δt_{onset} and Δt_{offset} must fall,
- constraints on the parent attributes, and
- constraints on the attributes contained within the respective subDAG:s.

4. Example Integration

It is now useful to turn to a more concrete example in order to better understand the integrator's functionality. The figures to which this section refers can be found on the last page of the paper.

The scenario is this: a user says ”Jag vill åka <u>där</u>ifrån till Resecentrum”. The underlining means that the user simultaneously pointed somewhere in the map-part of the interface. We will now step through the different parts of the integration process.

First the gesture is recognized and represented in the form seen in the left part of figure 2. This feature structure simply contains information on when and at which coordinate in the interface the gesture was made. This structure is then passed to the gesture interpreter, which realizes that the coordinate corresponds to some location in the map-part of the interface. Based on this it assumes that the user intended to indicate some physical location. The gesture interpreter does however not interpret exactly which location the user indicated. This is later done in the domain knowledge manager in the MALIN framework. Hence, if the user utters the name of one location while pointing to another, this is not resolved until the integrated result reaches the domain knowledge manager.

On the speech side, the utterance is first passed through the speech recognizer, which passes the string representation of the recognition result to the natural language interpreter. The parser outputs the DAG seen in figure 3.

The output of the gesture interpreter and the natural language interpreter is then passed to the integrator. Since neither of the DAG:s are empty initially, the algorithm proceeds to check all combinations of subDAG:s against the integration rules. The gesture-DAG contains three subDAG:s, while the speech-DAG contains seven. All in all, 21 combinations have to be checked for integration. Assume that the grammar contains the following rule:

- $(0.0 \leq \Delta t_{\text{onset}} \leq 1.0)$ AND
- $(2.0 \leq \Delta t_{\text{offset}} \leq 5.0)$ AND
- $(\text{parent\_attribute}_{\text{speech}}$ = ”locations”) AND
- $(\text{parent\_attribute}_{\text{gesture}}$ = ”locations) AND
- $(<\text{location}>_{\text{speech}} = [])$

The first and second constraints specify the intervals for Δt_{onset} and Δt_{offset} respectively. The first constraint declares that gesture must have the same onset as, or precede speech by up to most 1.0 time unit. The second constraint declares that the pen must be lifted from the touch screen somewhere between 2.0 and 5.0 time units before speech ends. Constraints four and five declares that both speech and gesture should have received an interpretation as being some form of location. Finally, constraint five declares that the speech-subDAG should contain an attribute 'location' that has an empty DAG as value, i.e., it should *not* have been interpreted as some *specific* location.

It should be clear in this simple example which parts of the DAG:s that are integrable, simply recall that cardinal numbers can not be parent attributes. The result of the integration can be seen in figure 4. Time stamp information has been dropped at this point since no other module in the MALIN framework makes use of it.

5. Discussion

This paper has presented a mechanism for integrating, or fusing, information received through different channels (i.e., speech and pen). However, some issues remain and deserve further elaboration. To some extent the four issues raised by Johnston et al. (1997) can serve as a useful ground for discussion.

First, the multimodal integrator has initially been designed to handle only simple deictic pointing gestures. However, this is not a fundamental limitation of the integrator. The choice to limit the integrator in this sense has been made because there is currently no gesture recognizer or gesture interpreter implemented. Therefore, minimal assumptions about these modules has been made.

Second, the integrator is to some extent speech-driven, i.e., in the presence of speech it tries to find elements in speech that can be integrated with gesture. However, the integration rules can be declared in such a way that the two modalities can stand on more equal ground. Furthermore, in the absence of speech, the interpreted gesture can constitute the entire interpretation.

Third, the multimodal integrator is based on a formally well defined and understood meaning representation formalism, i.e., feature structures. This makes the integrator more readily integrable with other parsers than if it utilized some novel technique.

Finally, the integrator is semi-formally well defined even though it lacks a full formal specification.

References

Dahlbäck, N., Flycht-Eriksson, A., Jönsson, A. and Qvarfordt, P. (1999). An Architecture for Multi-Modal Natural Dialogue Systems. In *Proceedings of ESCA Tutorial and Research Workshop (ETRW) on Interactive Dialogue in Multi-Modal Systems*, Germany.

Jackson, P. (1999). *Introduction to Expert Systems*. Addison Wesley Longman Ltd.

Koons, D. B., Sparrell, C. J. and Thorisson, K. R. (1993). Integrating Simultaneous Input from Speech, Gaze, and Hand Gestures. In M. T. Maybury (Ed.) *Intelligent Multimedia Interfaces*, pp. 243-261, Menlo Park, *CA*: *AAAI*/*MIT* Press.

Johnston, M. (1998). Unification-based Multimodal Parsing. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics*, pp. 624-630, Montreal, Quebec, Canada.

Johnston, M. and Bangalore, S. (2000). Finite-state Multimodal Parsing and Understanding. In *Proceedings of the 18th International Conference on Computational Linguistics*, Saarbrücken, Germany.

Johnston, M., Cohen, P. R., McGee, D., Oviatt, S. L., Pittman, J. A. and Smith, I. (1997). Unification-based Multimodal Integration. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pp. 281-288, Madrid, Spain.

Jönsson, A. (1997). A Model for Habitable and Efficient Dialogue Management for Natural Language Interaction. *Natural Language Engineering*, 3(2/3), pp. 103-122.

Neal, J. G., Thielman, C. Y., Dobes, Z., Haller, S. M. and Shapiro, S. C. (1989). Natural Language with Integrated Deictic and Graphic Gestures. In *Proceedings of the 1989 DARPA Workshop on Speech and Natural Language*, pp. 410-423.

Nigay, L. and Coutaz, J. (1995). A Generic Platform for Addressing the Multimodal Challenge. In *Proceedings of Conference on Human Factors in Computing Systems (CHI'95)*, pp. 98-105, Denver, Colorado.

Oviatt, S., Cohen, P., Wu, L., Vergo, J., Duncan, L., Suhm, B., Bers, J., Holzman, T., Winograd, T., Landay, J., Larson, J., and Ferro, D. (2000). Designing the User Interface for Multimodal Speech and Pen-based Gesture Applications: State-of-the-Art Systems and Future Research Directions. *Human-Computer Interaction*, 15(4), pp. 263-322.

Oviatt, S., DeAngeli, A., and Kuhn, K. (1997). Integration and Synchronization of Input Modes during Multimodal Human-Computer Interaction. In *Proceedings of Conference on Human Factors in Computing Systems*, pp. 415-422, Atlanta, Georgia.

Strömbäck, L. and Jönsson, A. (1998). Robust Interpretation for Spoken Dialogue Systems. In *Proceedings of the International Conference on Spoken Language Processing*, pp. 491-494, Sydney, Australia.

Wahlster, W. (1991). User Discourse Models for Multimodal Communication. In J. W. Sullivan and S. W. Tyler (Eds.) *Intelligent User Interfaces*, New York, *NY*: *ACM* Press.

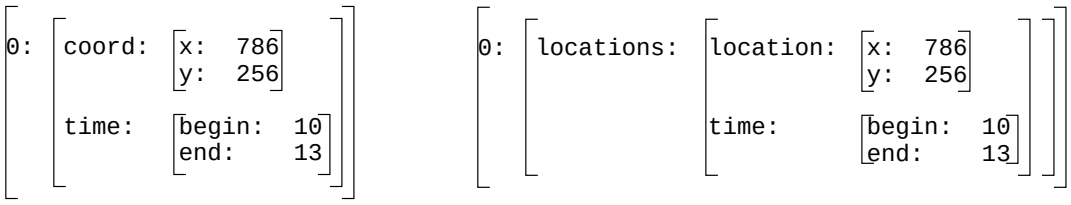

Figure 2: To the left a DAG representing a recognized gesture, and to the right its interpretation.

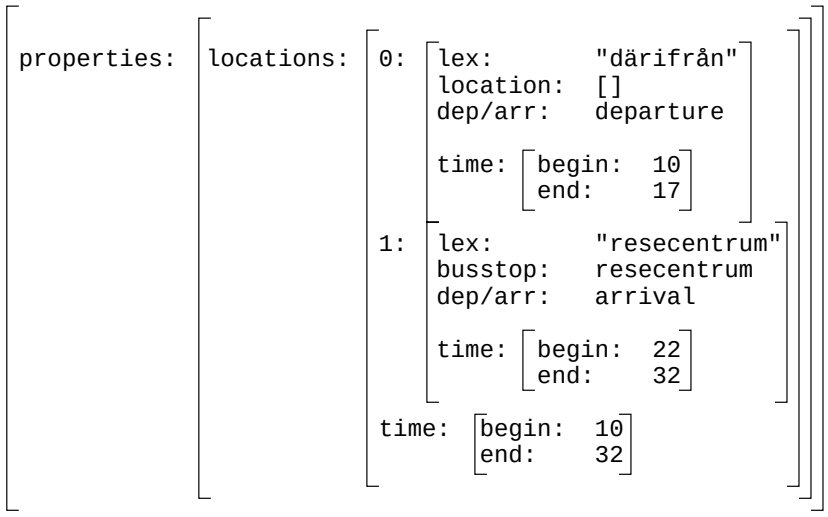

Figure 3: A DAG representing the interpretation of the spoken utterance ”Jag vill åka därifrån till Resecentrum”.

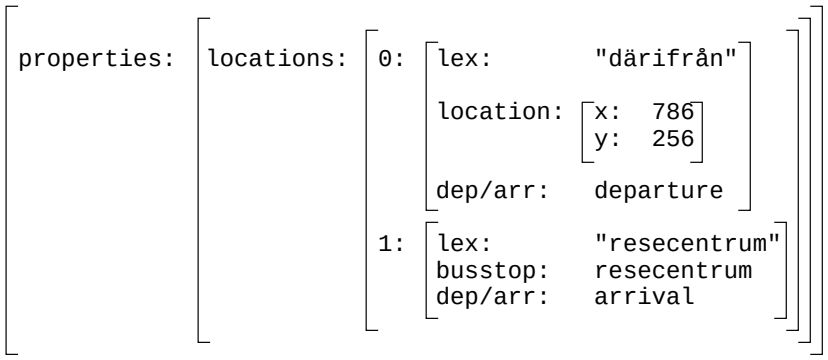

Figure 4: A DAG representing the result of integrating the DAG to the right in figure 2 and the DAG in figure 3.