A frame semantics for an IntelliMedia TourGuide

Paul Mc Kevitt* and Paul Dalsgaard
Center for PersonKommunikation (CPK)
Fredrik Bajers Vej 7-A2
Institute of Electronic Systems (IES)
Aalborg University
DK-9220, Aalborg
DENMARK, EU.
{pmck,pd}@cpk.auc.dk

Abstract

One of the most important issues in developing Intelligent MultiMedia (IntelliMedia) or the real-time computer processing, understanding and integration of perceptual input from speech, textual and visual sources is that of the semantics of communication between the various modules. We provide here such a semantics in terms of frames and give a worked example of how it can be used to process a sample query where the application is an IntelliMedia TourGuide giving information about building plans on an IntelliMedia Workbench. This is one application of our general CHAMELEON platform for performing IntelliMedia through integration of at least speech and images.

1 Introduction

The area of MultiMedia is growing rapidly internationally and it is clear that it has various meanings from various points of view. MultiMedia can be separated into at least two areas: (1) (traditional) MultiMedia and (2) Intelligent MultiMedia (*IntelliMedia*). The former area is the one that people think of as being MultiMedia, encompassing the display of text, voice, sound and video/graphics with possibly touch and virtual reality linked in. However, the computer has little or no understanding of the meaning of what it is presenting.

IntelliMedia, which involves the computer processing and understanding of perceptual input from speech, text and visual images and reacting to it is much more complex and involves research from Engineering, Computer Science and Cognitive Science (see Mc Kevitt 1995/96, 1997). This is the newest area of MultiMedia research which has seen an upsurge over the last two years and one where most universities internationally do not have expertise. Aalborg University, Denmark has initiated IntelliMedia 2000+ which involves research with the production of a number of real-time demonstrators showing examples of IntelliMedia applications and to establish a new Master's degree in IntelliMedia and a nation-wide MultiMedia Network which is concerned with technology transfer to industry. More details can be found on WWW: http://www.cpk.auc.dk/CPK/MMUI/.

Four research groups exist within the Faculty of Science and Technology in the Institute of Electronic Systems, each of them covering expertise which together is necessary for building up IntelliMedia systems. The four research groups are Computer Science (CS), Medical Informatics (MI), Laboratory of Image Analysis (LIA) and Center for PersonKommunikation (CPK) each of them contributing knowledge to platforms for specification, learning, integration and interactive applications, expert systems and decision taking, image/vision processing, and spoken language processing/sound localisation.

*Paul Mc Kevitt is also a British Engineering and Physical Sciences Research Council (EPSRC) Advanced Fellow at the University of Sheffield, England for five years under grant B/94/AF/1833 for the Integration of Natural Language, Speech and Vision Processing.

2 CHAMELEON

The four groups of IntelliMedia 2000+ are developing an IntelliMedia computing platform called CHAMELEON which will be general enough to be used for a number of different applications.

2.1 System architecture

The general architecture of CHAMELEON consists of the following modules:

Speech recognizer for recognizing input speech and mapping it into written text. The recognizer has the capability to determine intentions (query?, instruction!, declarative) of the user from pitch contour.

Speech synthesizer for providing speech output to give acknowledgements and verbal responses to queries and also for asking questions.

Gesture recognizer for determining the coordinates (X,Y) of a pointing gesture.

Image recognizer for conducting 3D image processing of objects on an IntelliMedia Workbench.

Dialogue/NLP module for parsing the output of the speech recognizer and determining a meaning representation. We use frames as a syntax for meaning representation. The dialogue model has a stored representation of the anticipated dialogue interaction with the user using a Dialogue Description Language (DDL) (see Baekgaard 1996, Dalsgaard and Baekgaard 1994, Larsen 1997).

Domain model for modelling the domain of application (e.g. Architectural Layout/Plans of building(s)/tenants)

Topsy, a synchronizer which detects and learns co-occurrences in the input and synchronizes output. Topsy uses learned co-occurrences to conduct reasoning and decision-taking over the frame semantics.

Microphone array for determining and localising on the coordinates (X,Y,Z) of a sound source.

Laser device for pointing to a specified location coordinates (X,Y) and in more advanced scenarios drawing route descriptions.

Blackboard which is a database of the recorded history of the MultiModal dialogue interaction in terms of frames. Any other module can read and write frames from/to the blackboard at any time.

CHAMELEON has a general architecture of communicating agent modules processing inputs and outputs from different modalities and each of which can be tailored to a number of application domains. CHAMELEON is being developed in both a top-down and bottom-up manner making sure it is general enough for multiple application domains but at the same time keeping particular domains in mind.

CHAMELEON demonstrates that existing software for (1) distributed processing and learning (see Manthey 1997a, 1997b), (2) decision taking, (3) image processing, (4) spoken dialogue processing (see Baekgaard 1996, Dalsgaard and Baekgaard 1994, Larsen 1997) and (5) microphone arrays (see Leth-Espensen and Lindberg 1996) can be interfaced to a single hub platform and act as communicating agent modules within it. The platform is independent of any particular application domain and the intention is to run it over different server machines. We are using programming languages C, C++, and Java for implementation. The software platform DACS (Distributed Applications Communication System) (see Fink et al. 1995) is used for integrating

Figure 1: IntelliMedia Workbench

An IntelliMedia Workbench has 2D architectural plans (or 3D model) on it. Initially we are working with 2D plans of a building at Aalborg University. Cameras are used to interpret the plans on the workbench and the user's pointing. At present the plans are not processed in real-time but are preprocessed. The system points using a laser pointing device and in more advanced scenarios will even give route descriptions for some destination. Microphone arrays perform sound localisation which aids speech processing. A computer monitor is linked in so that internet/WWW data about the building and the domain model, which has a database record of offices and their functionality/tenants, can be accessed.

We have implemented the image processing software, laser pointing drivers and domain model for this application and are currently tuning topsy and the spoken dialogue system for the application. The sample dialogue to be produced by the first prototype includes two of the most common phenomena which occur in IntelliMedia: (1) reference ("Who's office is this?") and (2) spatial relations ("Who's in the office beside him?"). Examples of other interesting problems to be solved as part of this application are resolution of ambiguity where a user says "Who's office is this?" but where the pointing-gesture is ambiguous since the person points sloppily between two rooms rather than into one. The system can then become active in the dialogue and ask the user for a clarification as to which office he/she means. Other interesting interactions are "Point to Paul's office" where there are two Pauls and again the system would ask a question.

2.3 Sample dialogue

We give here part of the sample dialogue which the first prototype of the system will process. The full sample dialogue is pregnant with phenomena such as instructions, questions, state-

ments/declaratives, ambiguity (CHAMELEON becomes active), ellipsis, reference (endophoric/ pronominal, exophoric/ pronominal), spatial relationships (beside), and learning.

```
USER:      Point to Hanne's office.
CHAMELEON: [points]
           This is Hanne's office.
USER:      Where is Jorgen's office?
CHAMELEON: [points]
           This is Jorgen's office.
USER:      Show me Paul's office.
CHAMELEON: [points(twice)]
           This is Paul Dalsgaard's office and
           this is Paul Mc Kevitt's office.
or/
CHAMELEON: There are two Paul's. Do you mean Paul Dalsgaard or Paul Mc Kevitt?
USER:      Paul Dalsgaard!
CHAMELEON: [points]
           This is Paul Dalsgaard's office.
```

3 Frame semantics

The frame language consists of three types of frames: (1) *input* (F-in), (2) *output* (F-out) and (3) *integration* (F-int). Input frames are those which come from modules conducting perceptual input, output frames are those produced by modules generating system output and integration frames are those which are processed by modules accessing the blackboard.

3.1 Input frames

An input frame (F-in) takes the general form:

[MODULE
 INPUT: input
 INTENTION: intention-type
 TIME: timestamp]

where MODULE is the module producing the frame, INPUT can be at least UTTERANCE or GESTURE, *input* is the utterance or gesture and intention-type includes different types of utterances and gestures. An utterance input frame is where intention-type can be at least (1) query?, (2) instruction! and (3) declarative. An example of an utterance input frame is:

[SPEECH
 UTTERANCE: (Point to Hanne's office)
 INTENTION: instruction!
 TIME: timestamp]

A gesture input frame is where intention-type can be at least (1) pointing!, (2) signal-1, (3) signal-2 where the meaning of signals are common knowledge between the user and system. An example of a gesture input frame is:

[GESTURE
 GESTURE: coordinates (3, 2)
 INTENTION: pointing
 TIME: timestamp]

3.2 Output frames

An output frame (F-out) takes the general form:

[MODULE
 INTENTION: intention-type
 OUTPUT: output
 TIME: timestamp]

where INTENTION is at least UTTERANCE or GESTURE, intention-type is the different types of utterance or gesture and output is the utterance or gesture. An utterance output frame is where intention-type is (1) query? (2) instruction!, and (3) declarative. An example utterance output frame is:

[SPEECH-SYNTHESIZER
 INTENTION: declarative
 UTTERANCE: (This is Hanne's office)
 TIME: timestamp]

A gesture output frame is where intention-type is (1) description (pointing), (2) description (signal-1), (3) description (signal-2) where signal meaning is common to user and system. An example utterance output frame is:

[LASER
 INTENTION: description (pointing)
 LOCATION: coordinates (5, 2)
 TIME: timestamp]

3.3 Integration frames

An integration frame (F-int) takes the general form:

[MODULE
 INTENTION: intention-type
 LOCATION: location
 OUTPUT: output
 TIME: timestamp]

where intention-type can be (1) query?, (2) instruction!, and (3) declarative, location is a specification of a location and OUTPUT is an UTTERANCE or GESTURE. An example utterance integration frame is:

[DIALOGUE/NLP
 INTENTION: description (pointing)
 LOCATION: office (owner Hanne) (coordinates (5, 2))
 UTTERANCE: (This is Hanne's office)
 TIME: timestamp]

We are currently investigating how to use frames to specify various types of reference and spatial relationships.

There are input and output gestures (G-in, G-out) and input and output utterances (U-in, U-out). Input modules are SPEECH-RECOGNIZER (U-in), IMAGE-GESTURE (G-in), and IMAGE-WORKBENCH (W-in). In our initial prototype the workbench images (2D building plans) are preprocessed by the system. Output modules are LASER (G-out) and SPEECH-SYNTHESIZER (U-out). Most modules give and take frames to/from the blackboard database and process them (F-int).

3.4 Giving an instruction

Here, we present all the steps and frames involved in processing an instruction "Point to Hanne's office" given to CHAMELEON. Although we show the various modules acting in a given sequence here, since CHAMELEON is intended to work in a completely distributed manner, then the modules processing and frames may not necessarily run in this order. The frames given are placed on the blackboard as they are produced and processed.

USER(U-in): Point to Hanne's office

PROCESSING(1):
SPEECH-RECOGNIZER:
(1) wakes up when it detects registering of U-in
(2) maps U-in into F-in
(3) places and registers F-in on blackboard:

FRAME(F-in)(1):
[SPEECH
UTTERANCE: (Point to Hanne's office)
INTENTION: instruction!
TIME: timestamp]

PROCESSING(2):
DIALOGUE/NLP:
(1) wakes up when it detects registering of F-in
(2) maps F-in into F-int
(3) places and registers F-int on blackboard:

FRAME(F-int)(1):
[DIALOGUE/NLP
INTENTION: instruction! (pointing)
LOCATION: office (owner Hanne) (coordinates (X, Y))
TIME: timestamp]

PROCESSING(3):
DOMAIN-MODEL:
(1) wakes up when it detects registering of F-int
(2) reads F-int and sees its from DIALOGUE/NLP
(3) produces updated F-int (coordinates)
(4) places and registers updated F-int on blackboard:

FRAME(F-int)(2):
[DOMAIN-MODEL
INTENTION: instruction! (pointing)
LOCATION: office (owner Hanne) (coordinates (5, 2))
TIME: timestamp]

PROCESSING(4):
DIALOGUE/NLP:
(1) wakes up when it detects registering of F-int
(2) reads F-int and sees it's from DOMAIN-MODEL
(3) produces updated F-int (intention + utterance)
(4) places and registers updated F-int on blackboard:

FRAME(F-int)(3):
[DIALOGUE/NLP
INTENTION: description (pointing)
LOCATION: office (owner Hanne) (coordinates (5, 2))
UTTERANCE: (This is Hanne's office)
TIME: timestamp]

PROCESSING(5):
LASER:
(1) wakes up when it detects registering of F-int
(2) reads F-int and sees it's from DOMAIN-MODEL
(3) produces F-out (pruning + registering)
(4) places and registers F-out on blackboard:

FRAME(F-out)(1):
[LASER
INTENTION: description (pointing)
LOCATION: coordinates (5, 2)
TIME: timestamp]

PROCESSING(6):
SPEECH-SYNTHESIZER:
(1) wakes up when it detects registering of F-int
(2) reads F-int and sees it's from DIALOGUE/NLP
(3) produces F-out (pruning + registering)
places and registers F-out on blackboard:

FRAME(F-out)(2):
[SPEECH-SYNTHESIZER
INTENTION: description
UTTERANCE: (This is Hanne's office)
TIME: timestamp]

PROCESSING(7):
TOPSY:
(1) wakes up when it detects registering of F-out and F-out
(2) reads F-out and F-out and sees they are from
LASER and SPEECH-SYNTHESIZER
(3) dials and fires LASER and SPEECH-SYNTHESIZER
in a rhythmic way (synchronized)
(1) LASER reads its own F-out and fires G-out
(2) SPEECH-SYNTHESIZER reads its own F-out and fires U-out

CHAMELEON(G-out): [points]
CHAMELEON(U-out): This is Hanne's office.

4 Conclusion

We have presented here a semantics for communication between various modules in our CHAMELEON platform being applied as an IntelliMedia TourGuide. The application is one where a system gives advice on the usage of a building and integrates speech and image processing, synchronization, and laser pointing technology. Frames are created by various modules and placed on a blackboard where they can be read, written and processed by other modules. We show that there are different types of frames in the semantics, the general form these take, and specific instances of them. A worked example with all associated frames and module interactions demonstrating an instruction ("Point to Hanne's office") is given. Future work will involve augmenting the frame semantics to handle more complex situations involving reference and spatial relations and testing various methods of communication and interaction between the modules.

Mobile computing aspects of the IntelliMedia TourGuide become evident if we consider the user walking in the building represented by the plans/model with a wearable computer (see Bruegge and Bennington 1996, Rudnicky et al. 1996, and Smailagic and Siewiorek 1996) and head-up display. Also, this research could eventually be incorporated into more advanced scenarios involving multiple speakers in a VideoConferencing environment say planning building and institution layout.

Intelligent MultiMedia will be important in the future of international computing and media development and IntelliMedia 2000+ at Aalborg University, Denmark brings together the necessary ingredients from research, teaching and links to industry to enable its successful implementation. Particularly, we have research groups in spoken dialogue processing, image processing, and radio communications which are the necessary features of this technology. Our IntelliMedia TourGuide application which focusses on giving help on building usage is an ideal one for testing integration of various modules.

Acknowledgements

We would like to acknowledge Tom Broendsted, Lars Bo Larsen, Mike Manthey, Thomas Moeslund, Kristian Olesen, and Helge Vad from IntelliMedia 2000+ who have provided various comments on the semantics and frames given here.

References

Baekgaard, Anders (1996) Dialogue Management in a Generic Dialogue System. In *Proceedings of the Eleventh Twente Workshop on Language Technology (TWLT), Dialogue Management in Natural Language Systems*, 123-132, Twente, The Netherlands.

Bruegge, Bernd and Ben Bennington (1996) Applications of wireless research to real industrial problems: applications of mobile computing and communication. In *IEEE Personal Communications*, 64-71, February.

Dalsgaard, Paul and A. Baekgaard (1994) Spoken Language Dialogue Systems. In *Prospects and Perspectives in Speech Technology: Proceedings in Artificial Intelligence*, Chr. Freksa (Ed.), 178-191, September. München, Germany: Infix.

Fink, G.A., N. Jungclaus, H. Ritter, and G. Sagerer (1995) A communication framework for heterogeneous distributed pattern analysis. In *Proc. International Conference on Algorithms and Architectures for Parallel Processing*, 881-890, Brisbane, Australia.

Larsen, L.B. (1996) Voice controlled home banking - objectives and experiences of the Esprit OVID project. In *Proceedings of IVTTA-96*, New Jersey, USA, September (IEEE 96-TH-8178).

Leth-Espensen, P. and B. Lindberg (1996) Separation of Speech Signals Using Eigenfiltering in a Dual Beamforming System. In *Proc. IEEE Nordic Signal Processing Symposium (NORSIG)*, Espoo, Finland, September, .

Manthey, Mike (1997a) *Distributed computation, the twisted isomorphism, and auto-poiesis.* Technical Report R-97-5007, Department of Computer Science, Aalborg university, Denmark, June.

Manthey, Mike (1997b) *The phase web paradigm and anticipatory systems, two short papers.* Technical Report R-97-5006, Department of Computer Science, Aalborg university, Denmark, June.

Mc Kevitt, Paul (Ed.) (1995/1996) *Integration of Natural Language and Vision Processing (Vols. I-IV).* Dordrecht, The Netherlands: Kluwer-Academic Publishers.

Mc Kevitt, Paul (1997) SuperinformationhighwayS. In *"Sprog og Multimedier" (Speech and Multimedia)*, Tom Broendsted and Inger Lytje (Eds.). 166-183, April 1997, Aalborg, Denmark: Aalborg Universitetsforlag (Aalborg University Press).

Rudnicky, Alexander I., Stephen D. Reed, Eric H. Thayer (1996) SpeechWear: a mobile speech system. In *Proceedings of International Symposium on Spoken Dialogue (ISSD 96), October 2-3, Wyndham Franklin Plaza Hotel, Philadelphia, USA*, Fujisaki, Hiroya (Ed.), 161-164. Tokyo, Japan: Acoustical Society of Japan (ASJ).

Smailagic, Asim and P. Siewiorek (1996) Matching interface design with user tasks: modalities of interaction with CMU wearable computers. In *IEEE Personal Communications*, 14-25, February.