The IntelliMedia WorkBench – An Environment for Building Multimodal Systems

Michael Manthey, Paul Mc Kevitt[⋆],
Thomas B. Moeslund, and Kristian G. Olesen

Institute for Electronic Systems (IES)
Aalborg University, Aalborg, Denmark
`mmui@cpk.auc.dk`

Abstract. Intelligent MultiMedia (IntelliMedia) focuses on the computer processing and understanding of signal and symbol input from at least speech, text and visual images in terms of semantic representations. We have developed a general suite of tools in the form of a software and hardware platform called "Chameleon" that can be tailored to conducting IntelliMedia in various application domains. Chameleon has an open distributed processing architecture and currently includes ten agent modules: blackboard, dialogue manager, domain model, gesture recogniser, laser system, microphone array, speech recogniser, speech synthesiser, natural language processor, and a distributed Topsy learner. Most of the modules are programmed in C and C++ and are glued together using the Dacs communications system. In effect, the blackboard, dialogue manager and Dacs form the kernel of Chameleon. Modules can communicate with each other and the blackboard which keeps a record of interactions over time via semantic representations in frames. Inputs to Chameleon can include synchronised spoken dialogue and images and outputs include synchronised laser pointing and spoken dialogue.

An initial prototype application of Chameleon is an *IntelliMedia WorkBench* where a user will be able to ask for information about things (e.g. 2D/3D models, pictures, objects, gadgets, people, or whatever) on a physical table. The current domain is a *Campus Information System* for 2D building plans which provides information about tenants, rooms and routes and can answer questions like *Whose office is this?* and *Show me the route from Paul Mc Kevitt's office to Paul Dalsgaard's office.* in real time. Chameleon and the IntelliMedia WorkBench are ideal for testing integrated signal and symbol processing of language and vision for the future of SuperinformationhighwayS.

[⋆] Paul Mc Kevitt was 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 and recently took up appointment as Chair in Digital MultiMedia at The University of Ulster (Magee), Northern Ireland (`p.mckevitt@ulst.ac.uk`).

H. Bunt and R.-J. Beun (Eds.): CMC'98, LNAI 2155, pp. 217–233, 2001.

1 Introduction

IntelliMedia, which involves the computer processing and understanding of perceptual input from at least speech, text and visual images, and then reacting to it, is complex and involves signal and symbol processing techniques from not just engineering and computer science but also artificial intelligence and cognitive science (Mc Kevitt 1994, 1995/96, 1997). With IntelliMedia systems, people can interact in spoken dialogues with machines, querying about what is being presented and even their gestures and body language can be interpreted.

People are able to combine the processing of language and vision with apparent ease. In particular, people can use words to describe a picture, and can reproduce a picture from a language description. Moreover, people can exhibit this kind of behaviour over a very wide range of input pictures and language descriptions. Although there are theories of how we process vision and language, there are few theories about how such processing is integrated. There have been extensive debates in psychology and philosophy with respect to the degree to which people store knowledge as propositions or pictures (Kosslyn and Pomerantz 1977, Pylyshyn 1973). Other recent moves towards integration are reported in Denis and Carfantan (1993), Mc Kevitt (1994, 1995/96) and Pentland (1993).

The Institute for Electronic Systems at Aalborg University, Denmark has expertise in the area of IntelliMedia and has already established an initiative called IntelliMedia 2000+ funded by the Faculty of Science and Technology. IntelliMedia 2000+ coordinates research on the production of a number of real-time demonstrators exhibiting examples of IntelliMedia applications, a new Master's degree in IntelliMedia, and a nation-wide MultiMedia Network (MMN) concerned with technology transfer to industry. A number of student projects related to IntelliMedia 2000+ have already been completed and currently five student groups are enrolled in the Master's conducting projects on multimodal interfaces, billiard game trainer, virtual steering wheel, audio-visual speech recognition, and face recognition. IntelliMedia 2000+ is coordinated from the Center for PersonKommunikation (CPK) which has a wealth of experience and expertise in spoken language processing, one of the central components of IntelliMedia, but also in radio communications which would be useful for mobile applications (CPK Annual Report, 1998). IntelliMedia 2000+ involves four research groups from three departments within the Institute for Electronic Systems: Computer Science (CS), Medical Informatics (MI), Laboratory of Image Analysis (LIA) and Center for PersonKommunikation (CPK), focusing on platforms for integration and learning, expert systems and decision taking, image/vision processing, and spoken language processing/sound localisation respectively. The first two groups provide a strong basis for methods of integrating semantics and conducting learning and decision taking while the latter groups focus on the two main input/output components of IntelliMedia, vision and speech/sound. More details on IntelliMedia 2000+ can be found at `http://www.cpk.auc.dk/imm`.

2 CHAMELEON and the IntelliMedia WorkBench

IntelliMedia 2000+ has developed the first prototype of an IntelliMedia software and hardware platform called Chameleon which is general enough to be used for a number of different applications. Chameleon demonstrates that existing software modules for (1) distributed processing and learning, (2) decision taking, (3) image processing, and (4) spoken dialogue processing can be interfaced to a single platform and act as communicating agent modules within it. Chameleon is independent of any particular application domain and the various modules can be distributed over different machines. Most of the modules are programmed in C++ and C. More details on Chameleon and the IntelliMedia WorkBench can be found in Brøndsted et al. (1998).

2.1 IntelliMedia WorkBench

An initial application of Chameleon is the *IntelliMedia WorkBench* which is a hardware and software platform as shown in Figure 1. One or more cameras and lasers can be mounted in the ceiling, and a microphone array placed on the wall, and there is a table where things (objects, gadgets, people, pictures, 2D/3D models, building plans, or whatever) can be placed. The current domain is a *Campus Information System* which at present gives information on the architectural and functional layout of a building. 2-dimensional (2D) architectural plans of the building drawn on white paper are laid on the table and the user can ask questions about them. At present the plans represent two floors of the 'A' (A2) building at Fredrik Bajers Vej 7, Aalborg University.

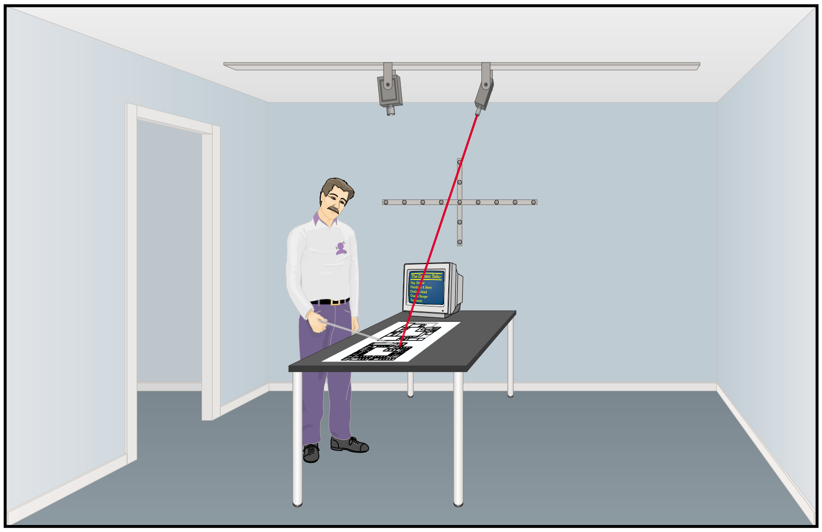

Fig. 1. Physical Layout of the IntelliMedia WorkBench.

Presently, there is one static camera which calibrates the plans on the table and the laser, and interprets the user's pointing while the system points to locations and draws routes with a laser. Inputs are simultaneous speech and/or pointing gestures and outputs are synchronised synthesised speech and pointing. We currently run all of Chameleon on a 200 MHz Intel pentium computer (r2d2) which handles input for the Campus Information System in real-time.

The 2D plan, which is placed on the table, is printed out on A0 paper having the dimensions: 84x118cm. Due to the size of the pointer's tip (2x1cm), the size of the table, the resolution of the camera and uncertainty in the tracking algorithm, a size limitation is introduced. The smallest room in the 2D plan, which is a standard office, can not be less than 3cm wide. The size of a standard office on the printout is 3x4cm which is a feasible size for the system. The 2D plan is shown in Figure 2.

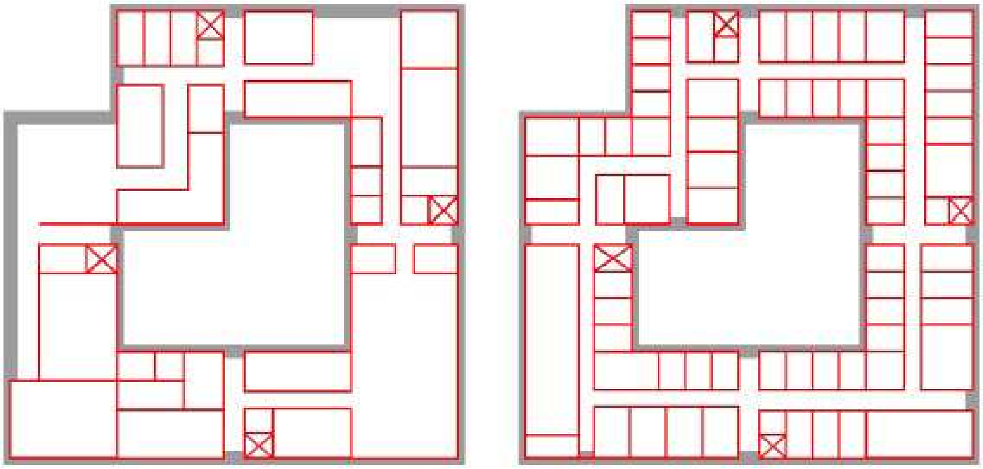

Fig. 2. 2D Plan of the 'A' Building at Fredrik Bajers Vej 7, Aalborg University. Left: Ground Floor; Right: 1st Floor.

2.2 Sample Interaction Dialogue

We present here a sample dialogue which the current first prototype can process. The example includes user intentions which are instructions and queries, and exophoric/deictic reference.

```
USER:      Show me Tom's office.
CHAMELEON: [points]
           This is Tom's office.
USER:      Point to Thomas' office.
CHAMELEON: [points]
           This is Thomas' office.
```

```
USER:      Where is the computer room?
CHAMELEON: [points]
           The computer room is here.
USER:      [points to instrument repair]
           Whose office is this?
CHAMELEON: [points]
           This is not an office, this is instrument repair.
USER:      [points]
           Whose office is this?
CHAMELEON: [points]
           This is Paul's office.
USER:      Show me the route from Lars Bo Larsen's office to
           Hanne Gade's office.
CHAMELEON: [draws route]
           This is the route from Lars Bo's office to Hanne's office.
USER:      Show me the route from Paul Mc Kevitt's office
           to instrument repair.
CHAMELEON: [draws route]
           This is the route from Paul's office to instrument repair.
USER:      Show me Paul's office.
CHAMELEON: [points]
           This is Paul's office.
```

2.3 Architecture of CHAMELEON

Chameleon has a distributed 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. The process synchronisation and intercommunication for Chameleon modules is performed using the Dacs (Distributed Applications Communication System) Inter Process Communication (IPC) software (see Fink et al. 1996) which enables Chameleon modules to be glued together and distributed across a number of servers. Presently, there are ten software modules in Chameleon: blackboard, dialogue manager, domain model, gesture recogniser, laser system, microphone array, speech recogniser, speech synthesiser, natural language processor (NLP), and Topsy as shown in Figure 3. Information flow and module communication within Chameleon are shown in Figures 4 and 5. Note that Figure 4 does not show the blackboard as a part of the communication but rather the abstract flow of information between modules. Figure 5 shows the actual passing of information between the speech recogniser, NLP module, and dialogue manager. As is shown all information exchange between individual modules is carried out using the blackboard as mediator.

As the intention is that no direct interaction between modules need take place the architecture is modularised and open but there are possible performance costs. However, nothing prohibits direct communication between two or more modules if this is found to be more convenient. For example, the speech recogniser and NLP modules can interact directly as the parser needs every recognition

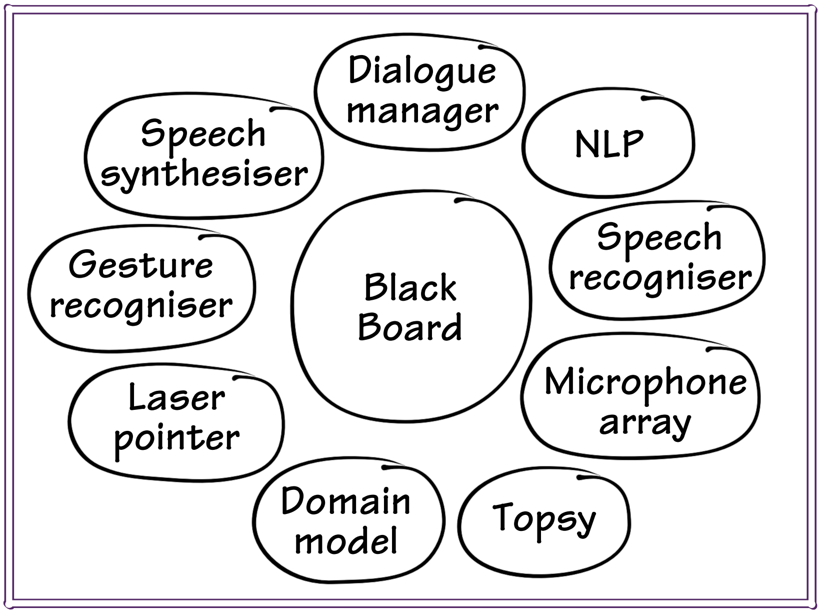

Fig. 3. Architecture of CHAMELEON.

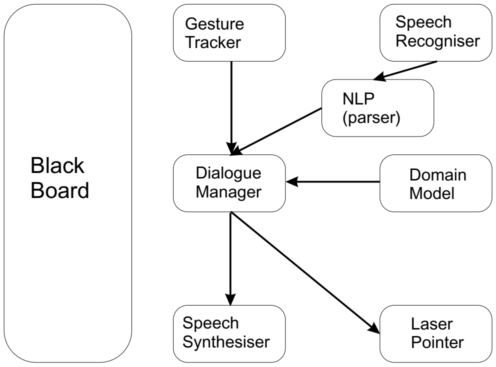

Fig. 4. Information Flow and Module Communication.

result anyway and at present no other module has use for output from the speech recogniser. The blackboard and dialogue manager form the kernel of CHAMELEON. We shall now give a brief description of each module.

The **blackboard** stores semantic representations produced by each of the other modules and keeps a history of these over the course of an interaction. All modules communicate through the exchange of semantic representations with each other or the blackboard. Semantic representations are frames in the spirit of Minsky (1975) and our frame semantics consists of (1) input, (2) output, and (3) integration frames for representing the meaning of intended user input and

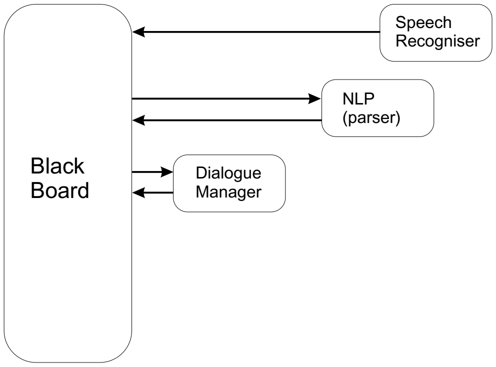

Fig. 5. Information Flow with the Blackboard.

system output. The intention is that all modules in the system will produce and read frames. Frames are coded in Chameleon as messages built of predicate-argument structures following the BNF definition given in Appendix A. The frame semantics was presented in Mc Kevitt and Dalsgaard (1997) and for the sample dialogue given in Section 2.2. Chameleon's actual blackboard history in terms of frames (messages) is shown in Appendix B.

The **dialogue manager** makes decisions about which actions to take and accordingly sends commands to the output modules (laser and speech synthesiser) via the blackboard. At present the functionality of the dialogue manager is to integrate and react to information coming in from the speech/NLP and gesture modules and to sending synchronised commands to the laser system and the speech synthesiser modules. Phenomena such as managing clarification subdialogues where Chameleon has to ask questions are not included at present. It is hoped that in future prototypes the dialogue manager will enact more complex decision taking over semantic representations from the blackboard using, for example, the HUGIN software tool (Jensen (F.) 1996) based on Bayesian Networks (Jensen (F.V.) 1996).

The **domain model** contains a database of all locations and their functionality, tenants and coordinates. The model is organised in a hierarchical structure: areas, buildings and rooms. Rooms are described by an identifier for the room (room number) and the type of the room (office, corridor, toilet, etc.). The model includes functions that return information about a room or a person. Possible inputs are coordinates or room number for rooms and name for persons, but in principle any attribute can be used as key and any other attribute can be returned. Furthermore, a path planner is provided, calculating the shortest route between two locations.

A design principle of imposing as few physical constraints as possible on the user (e.g. data gloves or touch screens) leads to the inclusion of a vision based **gesture recogniser**. Currently, it tracks a pointer via a camera mounted in the ceiling. Using one camera, the gesture recogniser is able to track 2D pointing gestures in real time. Only two gestures are recognised at present: pointing and not-pointing. The recognition of other more complex kinds of gestures like marking an area and indicating a direction (with hands and fingers) will be incorporated in the next prototype.

The camera continuously captures images which are digitised by a framegrabber. From each digitised image the background is subtracted leaving only the motion (and some noise) within this image. This motion is analysed in order to find the direction of the pointing device and its tip. By temporal segmenting of these two parameters, a clear indication of the position the user is pointing to at a given time is found. The error of the tracker is less than one pixel (through an interpolation process) for the pointer.

A **laser system** acts as a 'system pointer'. It can be used for pointing to positions, drawing lines and displaying text. The laser beam is controlled in real-time (30 kHz). It can scan frames containing up to 600 points with a refresh rate of 50 Hz thus drawing very steady images on surfaces. It is controlled by a standard Pentium PC host computer. The pointer tracker and the laser pointer have been carefully calibrated so that they can work together. An automatic calibration procedure has been set up involving both the camera and laser where they are tested by asking the laser to follow the pointer.

A **microphone array** (Leth-Espensen and Lindberg 1996) is used to locate sound sources, e.g. a person speaking. Depending upon the placement of a maximum of 12 microphones it calculates sound source positions in 2D or 3D. It is based on measurement of the delays with which a sound wave arrives at the different microphones. From this information the location of the sound source can be identified. Another application of the array is to use it to focus at a specific location thus enhancing any acoustic activity at that location. This module is in the process of being incorporated into Chameleon.

Speech recognition is handled by the grapHvite real-time continuous speech recogniser (Power et al. 1997). It is based on HMMs (Hidden Markov Models) of triphones for acoustic decoding of English or Danish. The recognition process focuses on recognition of speech concepts and ignores non content words or phrases. A finite state network describing phrases is created by hand in accordance with the domain model and the grammar for the natural language parser. The latter can also be performed automatically by a grammar converter in the NLP module. The speech recogniser takes speech signals as input and produces text strings as output. Integration of the latest CPK speech recogniser (Christensen et al. 1998) which is under development is being considered.

We use the Infovox Text-To-Speech (TTS) **speech synthesiser** which at present is capable of synthesising Danish and English (Infovox 1994). It is a rule based formant synthesiser and can simultaneously cope with multiple languages, e.g. pronounce a Danish name within an English utterance. Infovox takes text as input and produces speech as output. Integration of the CPK speech synthesiser (Nielsen et al. 1997) which is under development for English is being considered.

Natural language processing is based on a compound feature based (so-called unification) grammar formalism for extracting semantics from the one-best utterance text output from the speech recogniser (Brøndsted 1998). The parser carries out a syntactic constituent analysis of input and subsequently maps values into semantic frames. The rules used for syntactic parsing are based on a subset of the EUROTRA formalism, i.e. in terms of lexical rules and structure building rules (Bech 1991). Semantic rules define certain syntactic subtrees and which frames to create if the subtrees are found in the syntactic parse trees. The natural language generator is currently under construction and at present generation is conducted by using canned text.

The basis of the Phase Web paradigm (Manthey 1998), and its incarnation in the form of a program called "Topsy", is to represent knowledge and behaviour in the form of hierarchical relationships between the mutual exclusion and co-occurrence of events. In AI parlance, Topsy is a distributed, associative, continuous-action, dynamic partial-order planner that learns from experience. Relative to MultiMedia, integrating independent data from multiple media begins with noticing that what ties otherwise independent inputs together is the fact that they occur simultaneously. This is also Topsy's basic operating principle, but this is further combined with the notion of mutual exclusion, and thence to hierarchies of such relationships (Manthey 1998).

2.4 DACS

Dacs is currently the communications system for Chameleon and the IntelliMedia WorkBench and is used to glue all the modules together enabling communication between them. Applications of Chameleon typically consist of several interdependent modules, often running on separate machines or even dedicated hardware. This is indeed the case for the IntelliMedia WorkBench application. Such distributed applications have a need to communicate in various ways. Some modules feed others in the sense that all generated output from one is treated further by another. In the Campus Information System all modules report their output to the blackboard where it is stored. Although our intention is currently to direct all communication through the blackboard, we could just as well have chosen to simultaneously transfer output to several modules. For example, utterances collected by the speech recogniser can be sent to the blackboard but also sent simultaneously to the NLP module which may become relevant when efficiency is an important issue.

Another kind of interaction between processes is through remote procedure calls (RPCs), which can be either *synchronous* or *asynchronous*. By synchronous RPCs we understand procedure calls where we want immediate feedback, that is, the caller stops execution and waits for an answer to the call. In the Campus Information System this could be the dialogue manager requesting the last location to which a pointing event occurred. In the asynchronous RPC, we merely submit a request and carry on with any other task. This could be a request to the speech synthesiser to produce an utterance for the user or to the laser to point to some specific location. These kinds of interaction should be available in a uniform way in a heterogeneous environment, without specific concern about what platform the sender and receiver run on.

All these facilities are provided by the Distributed Applications Communication System (Dacs) developed at the University of Bielefeld, Germany (Fink et al. 1995, 1996), where it was designed as part of a larger research project developing an IntelliMedia platform (Rickheit and Wachsmuth 1996) discussed further in the next section. Dacs uses a communication demon on each participating machine that runs in user mode, allows multiple users to access the system simultaneously and does not provide a virtual machine dedicated to a single user. The demon acts as a router for all internal traffic and establishes connections to demons on remote machines. Communication is based on simple asynchronous message passing with some extensions to handle dynamic reconfigurations of the system during runtime. Dacs also provides on top more advanced communication semantics like RPCs (synchronous and asynchronous) and *demand streams* for handling data parts in continuous data streams. All messages transmitted are recorded in a Network Data Representation which includes type and structure information. Hence, it is possible to inspect messages at any point in the system and to develop generic tools that can handle any kind of data. Dacs uses Posix threads to handle connections independently in parallel. A database in a central name service stores the system configuration to keep the network traffic low during dynamic reconfigurations. A Dacs Debugging Tool (DDT) allows inspection of messages before they are delivered, monitoring configurations of the system, and status on connections.

3 Relation to Other Work

Situated Artificial Communicators (SFB-360) (Rickheit and Wachsmuth 1996) is a collaborative research project at the University of Bielefeld, Germany which focuses on modelling that which a person performs when with a partner he cooperatively solves a simple assembly task in a given situation. The object chosen is a model airplane (Baufix) to be constructed by a robot from the components of a wooden building kit with instructions from a human. SFB-360 includes equivalents of the modules in Chameleon although there is no learning module competitor to Topsy. What SFB-360 gains in size it may loose in integration, i.e. it is not clear yet that all the technology from the subprojects have been fitted together and in particular what exactly the semantic representations passed between the modules are. The Dacs process communication system currently used in Chameleon is a useful product from SFB-360.

Gandalf is a communicative humanoid which interacts with users in MultiModal dialogue through using and interpreting gestures, facial expressions, body language and spoken dialogue (Thórisson 1997). Gandalf is an application of an architecture called *Ymir* which includes perceptual integration of multimodal events, distributed planning and decision making, layered input analysis and motor-control with human-like characteristics and an inherent knowledge of time. Ymir has a blackboard architecture and includes modules equivalent to those in Chameleon. However, there is no vision/image processing module in the sense of using cameras since gesture tracking is done with the use of a data glove and body tracking suit and an eye tracker is used for detecting the user's eye gaze. However, it is anticipated that Ymir could easily handle the addition of such a vision module if one were needed. Ymir has no learning module equivalent to Topsy. Ymir's architecture is even more distributed than CHAMELEON's with many more modules interacting with each other. Ymir's semantic representation is much more distributed with smaller chunks of information than our frames being passed between modules.

AESOPWORLD is an integrated comprehension and generation system for integration of vision, language and motion (Okada 1997). It includes a model of mind consisting of nine domains according to the contents of mental activities and five levels along the process of concept formation. The system simulates the protagonist or fox of an Aesop fable, "The Fox and the Grapes", and his mental and physical behaviour are shown by graphic displays, a voice generator, and a music generator which expresses his emotional states. AESOPWORLD has an agent-based distributed architecture and also uses frames as semantic representations. It has many modules in common with CHAMELEON although again there is no vision input to AESOPWORLD which uses computer graphics to depict scenes. AESOPWORLD has an extensive planning module but conducts more traditional planning than CHAMELEON's Topsy.

The INTERACT project (Waibel et al. 1996) involves developing MultiModal Human Computer Interfaces including the modalities of speech, gesture and pointing, eye-gaze, lip motion and facial expression, handwriting, face recognition and tracking, and sound localisation. The main concern is with improving recognition accuracies of modality-specific component processors as well as developing optimal combinations of multiple input signals to deduce user intent more reliably in cross-modal speech acts. INTERACT also uses a frame representation for integrated semantics from gesture and speech and partial hypotheses are developed in terms of partially filled frames. The output of the interpreter is obtained by unifying the information contained in the partial frames. Although Waibel et al. present sophisticated work on multimodal interfaces it is not clear that they have developed an integrated platform which can be used for developing multimodal applications.

4 Conclusion and Future Work

We have described the architecture and functionality of CHAMELEON: an open, distributed architecture with ten modules glued into a single platform using the DACS communication system. We described the IntelliMedia WorkBench application, a software and physical platform where a user can ask for information about things on a physical table. The current domain is a *Campus Information System* where 2D building plans are placed on the table and the system provides information about tenants, rooms and routes and can answer questions like *Whose office is this?* in real time. CHAMELEON fulfils the goal of developing a general platform for integration of at least language/vision processing which can be used for research but also for student projects as part of the Master's degree education. More details on Chameleon and the IntelliMedia WorkBench can be found in Brøndsted et al. (1998).

There are a number of avenues for future work with Chameleon. We would like to process dialogue that includes examples of (1) spatial relations and (2) anaphoric reference. It is hoped that more complex decision taking can be introduced to operate over semantic representations in the dialogue manager or blackboard using, for example, the HUGIN software tool (Jensen (F.) 1996) based on Bayesian Networks (Jensen (F.V.) 1996). The gesture module will be augmented so that it can handle gestures other than pointing. Topsy will be asked to do more complex learning and processing of input/output from frames. The microphone array has to be integrated into Chameleon and set to work. Also, at present Chameleon is static and it might be interesting to see how it performs whilst being integrated with a web-based virtual or real robot or as part of an intellimedia videoconferencing system where multiple users can direct cameras through spoken dialogue and gesture. A miniature version of this idea has already been completed as a student project (Bakman et al. 1997).

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. Our Chameleon platform and IntelliMedia WorkBench application are ideal for testing integrated processing of language and vision for the future of SuperinformationhighwayS.

Acknowledgements

This opportunity is taken to acknowledge support from the Faculty of Science and Technology, Aalborg University, Denmark and Paul Mc Kevitt would also like to acknowledge the British Engineering and Physical Sciences Research Council (EPSRC) for their generous funded support under grant B/94/AF/1833 for the Integration of Natural Language, Speech and Vision Processing (Advanced Fellow).

References

Bakman, L., M. Blidegn, T.D. Nielsen, and S. Carrasco Gonzalez (1997) *NIVICO - Natural Interface for VIdeo COnferencing*. Project Report (8th Semester), Department of Communication Technology, Institute for Electronic Systems, Aalborg University, Denmark.

Bech, A. (1991) Description of the EUROTRA framework. In *The Eurotra Formal Specifications, Studies in Machine Translation and Natural Language Processing*, C. Copeland, J. Durand, S. Krauwer, and B. Maegaard (Eds), Vol. 2, 7-40. Luxembourg: Office for Official Publications of the Commission of the European Community.

Brøndsted, T. (1998) *nlparser*. http://www.kom.auc.dk/˜tb/nlparser

Brøndsted, T., P. Dalsgaard, L.B. Larsen, M. Manthey, P. Mc Kevitt, T.B. Moeslund, and K.G. Olesen (1998) *A platform for developing Intelligent MultiMedia applications.* Technical Report R-98-1004, Center for PersonKommunikation (CPK), Institute for Electronic Systems (IES), Aalborg University, Denmark, May.

Christensen, H., B. Lindberg, and P. Steingrimsson (1998) *Functional specification of the CPK Spoken LANGuage recognition research system (SLANG).* Center for PersonKommunikation, Aalborg University, Denmark, March.

CPK Annual Report (1998) *CPK Annual Report.* Center for PersonKommunikation (CPK), Fredrik Bajers Vej 7-A2, Institute for Electronic Systems (IES), Aalborg University, DK-9220, Aalborg, Denmark.

Denis, M. and M. Carfantan (Eds.) (1993) *Images et langages: multimodalité et modelisation cognitive.* Actes du Colloque Interdisciplinaire du Comité National de la Recherche Scientifique, Salle des Conférences, Siège du CNRS, Paris, April.

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 Applications for Parallel Processing*, V. L. Narasimhan (Ed.), 881-890. IEEE, Brisbane, Australia.

Fink, G.A., N. Jungclaus, and F. Kummert, H. Ritter, and G. Sagerer (1996) A distributed system for integrated speech and image understanding. In *Proceedings of the International Symposium on Artificial Intelligence*, Rogelio Soto (Ed.), 117-126. Cancun, Mexico.

Infovox (1994) *INFOVOX: Text-to-speech converter user's manual (version 3.4).* .Solna, Sweden: Telia Promotor Infovox AB

Jensen, F.V. (1996) *An introduction to Bayesian Networks.* London, England: UCL Press.

Jensen, F. (1996) Bayesian belief network technology and the HUGIN system. In *Proceedings of UNICOM seminar on Intelligent Data Management*, Alex Gammerman (Ed.), 240-248. Chelsea Village, London, England, April.

Kosslyn, S.M. and J.R. Pomerantz (1977) Imagery, propositions and the form of internal representations. In *Cognitive Psychology*, 9, 52-76.

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, 235-238.

Manthey, M.J. (1998) The Phase Web Paradigm. In *International Journal of General Systems, special issue on General Physical Systems Theories*, K. Bowden (Ed.). in press.

Mc Kevitt, P. (1994) Visions for language. In *Proceedings of the Workshop on Integration of Natural Language and Vision processing*, Twelfth American National Conference on Artificial Intelligence (AAAI-94), Seattle, Washington, USA, August, 47-57.

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

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

Mc Kevitt, P. and P. Dalsgaard (1997) A frame semantics for an IntelliMedia TourGuide. In *Proceedings of the Eighth Ireland Conference on Artificial Intelligence*

(AI-97), Volume 1, 104-111. University of Uster, Magee College, Derry, Northern Ireland, September.

Minsky, M. (1975) A framework for representing knowledge. In *The Psychology of Computer Vision*, P.H. Winston (Ed.), 211-217. New York: McGraw-Hill.

Nielsen, C., J. Jensen, O. Andersen, and E. Hansen (1997) *Speech synthesis based on diphone concatenation.* Technical Report, No. CPK971120-JJe (in confidence), Center for PersonKommunikation, Aalborg University, Denmark.

Okada, N. (1997) Integrating vision, motion and language through mind. In *Proceedings of the Eighth Ireland Conference on Artificial Intelligence (AI-97), Volume 1*, 7-16. University of Uster, Magee, Derry, Northern Ireland, September.

Pentland, A. (Ed.) (1993) *Looking at people: recognition and interpretation of human action.* IJCAI-93 Workshop (W28) at The 13th International Conference on Artificial Intelligence (IJCAI-93), Chambéry, France, August.

Power, K., C. Matheson, D. Ollason, and R. Morton (1997) *The grapHvite book (version 1.0).* Cambridge, England: Entropic Cambridge Research Laboratory Ltd..

Pylyshyn, Z. (1973) What the mind's eye tells the mind's brain: a critique of mental imagery. In *Psychological Bulletin*, 80, 1-24.

Rickheit, G. and I. Wachsmuth (1996) Collaborative Research Centre "Situated Artificial Communicators" at the University of Bielefeld, Germany. In *Integration of Natural Language and Vision Processing, Volume IV, Recent Advances*, Mc Kevitt, Paul (ed.), 11-16. Dordrecht, The Netherlands: Kluwer Academic Publishers.

Thórisson, K.R. (1997) Layered action control in communicative humanoids. In *Proceedings of Computer Graphics Europe '97*, June 5-7, Geneva, Switzerland.

Waibel, A., M.T. Vo, P. Duchnowski, and S. Manke (1996) Multimodal interfaces. In *Integration of Natural Language and Vision Processing, Volume IV, Recent Advances*, Mc Kevitt, Paul (Ed.), 145-165. Dordrecht, The Netherlands: Kluwer Academic Publishers.

Appendix A

Syntax of Frames

The following BNF grammar defines a predicate-argument syntax for the form of messages (frames) appearing on CHAMELEON's implemented blackboard.

```
FRAME          ::= PREDICATE

PREDICATE      ::= identifier(ARGUMENTS)

ARGUMENTS      ::= ARGUMENT
                |  ARGUMENTS, ARGUMENT

ARGUMENT       ::= CONSTANT
                |  VARIABLE
                |  PREDICATE
```

```
CONSTANT        ::= identifier
                 |  integer
                 |  string

VARIABLE        ::= $identifier
```

FRAME acts as start symbol, CAPITAL symbols are non-terminals, and terminals are lower-case or one of the four symbols ( ) , and $. An *identifier* starts with a letter that can be followed by any number of letters, digits or _, an *integer* consists of a sequence of digits and a *string* is anything delimited by two "'s. Thus the *alphabet* consists of the letters, the digits and the symbols ( ) , _ and $. A parser has been written in C which can parse the frames using this BNF definition.

Appendix B

Blackboard in Practice

Here we show the complete blackboard (with all frames) as produced exactly by Chameleon for the example dialogue given in Section 2.

```
Received: nlp(intention(instruction(pointing)),location(person(tb),
type(office)),time(889524794))
which is passed on to dialog_manager
Received: dialog_manager(output(laser(point(coordinates(249,623))),
speech_synthesizer(utterance("This is Toms office"))))
Calling laser: laser(point(coordinates(249,623)))
Calling speech_synthesizer:
speech_synthesizer(utterance("This is Toms office"))
Received: nlp(intention(instruction(pointing)),location(person(tbm),
type(office)),time(889524818))
which is passed on to dialog_manager
Received: dialog_manager(output(laser(point(coordinates(278,623))),
speech_synthesizer(utterance("This is Thomass office"))))
Calling laser: laser(point(coordinates(278,623)))
Calling speech_synthesizer:
speech_synthesizer(utterance("This is Thomass office"))


Received: nlp(intention(query(where)),location(place(a2_221)),
time(889524831))
which is passed on to dialog_manager
Received: dialog_manager(output(laser(point(coordinates(132,500))),
speech_synthesizer(utterance("computer room is here"))))
Calling laser: laser(point(coordinates(132,500)))
Calling speech_synthesizer:
speech_synthesizer(utterance("computer room is here"))
```

```
Received: nlp(intention(query(who)),location(this($Deixis),
type(office)),time(889524864))
which is passed on to dialog_manager
Received: dialog_manager(output(laser(point(coordinates(658,546))),
speech_synthesizer(
utterance("This is not an office, this is instrument repair"))))
Calling laser: laser(point(coordinates(658,546)))
Calling speech_synthesizer:
speech_synthesizer(
utterance("This is not an office, this is instrument repair"))


Received: nlp(intention(query(who)),location(this($Deixis),
type(office)),time(889524885))
which is passed on to dialog_manager
Received: dialog_manager(output(laser(point(coordinates(223,568))),
speech_synthesizer(utterance("This is Pauls office"))))
Calling laser: laser(point(coordinates(223,568)))
Calling speech_synthesizer:
speech_synthesizer(utterance("This is Pauls office"))


Received: nlp(intention(instruction(show_route)),
source(location(person(lbl),type(office))),
destination(location(person(hg),type(office))),time(889524919))
which is passed on to dialog_manager
Received: dialog_manager(output(laser(route(
coordinates(278,585,278,603,249,603,220,603,197,603,197,623))),
speech_synthesizer(
utterance("This is the route from Lars Bos office to Hannes office"))))
Calling laser:
laser(route(
coordinates(278,585,278,603,249,603,220,603,197,603,197,623)))
Calling speech_synthesizer:
speech_synthesizer(
utterance("This is the route from Lars Bos office to Hannes office"))


Received: nlp(intention(instruction(show_route)),
source(location(person(pmck),
type(office))),destination(location(place(a2_105))),time(889524942))
which is passed on to dialog_manager
Received:
dialog_manager(output(laser(route(
coordinates(174,453,153,453,153,481,153,500,153,510,153,
540,153,569,153,599,153,603,184,603,197,603,220,603,249,
603,278,603,307,603,330,603,330,655,354,655,911,655,884,
655,884,603,810,603,759,603,717,603,717,570,696,570))),
speech_synthesizer(
utterance("This is the route from Pauls office to instrument repair"))))
Calling laser:
laser(route(coordinates(174,453,153,453,153,481,153,500,153,
510,153,540,153,569,153,599,153,603,184,603,197,603,220,603,
249,603,278,603,307,603,330,603,330,655,354,655,911,655,884,
655,884,603,810,603,759,603,717,603,717,570,696,570)))
Calling speech_synthesizer:
speech_synthesizer(
utterance(
"This is the route from Pauls office to instrument repair"))


Received: nlp(intention(instruction(pointing)),location(person(pd),
type(office)),time(889524958))
which is passed on to dialog_manager
Received: dialog_manager(output(laser(point(coordinates(220,585))),
speech_synthesizer(utterance("This is Pauls office"))))
```