A new approach to decision-making within an Intelligent MultiMedia distributed platform hub

A New Approach to Decision-Making within an Intelligent MultiMedia Distributed Platform Hub

Glenn G. Campbell, Tom Lunney, Aiden Mc Caughey, and Paul Mc Kevitt

School of Computing and Intelligent Systems
Faculty of Engineering
University of Ulster, Magee
Derry/Londonderry, BT48 7JL, N. Ireland
{Campbell-g8, TF.Lunney, A.McCaughey, P.McKevitt}@ulster.ac.uk

Abstract. Research relating to the development of an intelligent multimedia distributed platform hub (MediaHub) is presented. Related research is reviewed and a new approach to decision-making based on Bayesian networks is proposed. A system architecture, including a Whiteboard, Dialogue Manager, Semantic Representation Database and Decision-Making Module is outlined. Psyclone, a platform for distributed processing, will facilitate communication within MediaHub, Bayesian networks will enact decision-making within the Decision-Making Module and the Hugin Bayesian decision-making tool will implement Bayesian reasoning within MediaHub.

1 Introduction

Gone are the days when the only form of input to a computer was through the traditional keyboard and mouse. We can now talk to the computer in natural language which can convert our speech into text. There are currently many commercially available software products, such as ViaVoice [1], that provide speech-to-text technology. Much research has been carried out into investigating other forms of human-computer interaction and new ways of communicating with computers have subsequently evolved. Humans can provide input to systems using facial expressions, gestures, touch and gaze. The term ‘multimodal systems’ is used to refer to systems capable of accepting such input using multiple modes of communication.

Although much has been achieved in the development of intelligent multimodal systems in recent years, many challenges still remain. While recent research has resulted in systems capable of multimodal communication, this communication is very much on the computer’s terms. The user must learn to use the system and the communication is constrained to suit the application. If we are to achieve truly human-like communication with computers, then the user must be able to dictate the terms of communication. That is, the system must learn to meet the needs of the user instead of the user learning to use the system. In order to realise such systems we must investigate new methods of representing multimodal input/output, communication and decision-making in multimodal systems. In this paper we describe our ongoing work on the development of an intelligent multimedia distributed platform hub that will utilise Bayesian networks for decision-making over multimodal data. In section 2 we give a brief review of related research. Section 3 presents the current system architecture of MediaHub. Section 4 discusses distributed processing in MediaHub. In section 5 we look briefly at the Hugin software tool [2]. Section 6 considers decision-making in MediaHub, whilst section 7 concludes with a discussion on MediaHub's future development.

2 Related Research

In this section we provide a brief review of related research. Section 2.1 reviews existing distributed processing tools. In section 2.2 we discuss some existing multimodal platforms, whilst in section 2.3 we consider the Psyclone platform [3] in greater detail.

2.1 Distributed Processing Tools

Numerous distributed processing tools and platforms have been developed that assist the creation of intelligent multimodal distributed systems. DACS (Distributed Applications Communication System) [4], as used in the CHAMELEON platform [5], is a tool for process synchronisation and intercommunication. DACS uses simple asynchronous message passing to enable large distributed systems to be developed. Each module within the system must register using a unique name. This name is passed to a name server and is used by other modules to address it. DACS is used in applications that require the integration of existing heterogeneous software systems.

CORBA (Common Object Request Broker Architecture) [6] constitutes a standard architecture for distributed object systems, allowing the services of an object to be requested using the Object Request Broker (ORB). The operation of the ORB is entirely transparent to the client object. The client simply requests the services of a distributed object, the ORB delivers the request to the object and returns the result back to the client. The Open Agent Architecture (OAA) [7] provides a powerful framework for the development of agent based systems.

2.2 Multimodal Platforms

Several intelligent multimodal platforms have been developed with the aim of advancing towards truly intelligent multimodal systems. SmartKom [8] employs an intelligent conversational agent called Smartakus to communicate with the user. Smartakus uses speech, gestures and facial expressions to engage in human-like conversation with users. SmartKom's system architecture is shown in Fig. 1.

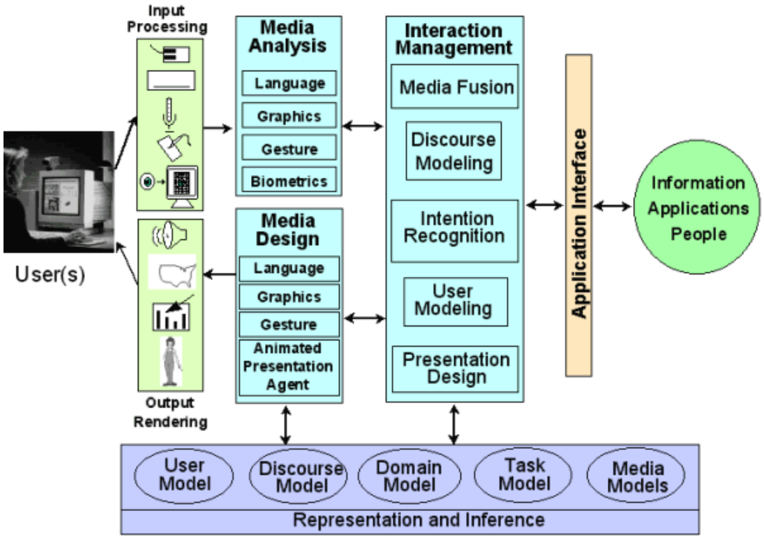

Fig. 1. Architecture of SmartKom [8]

SmartKom's distributed architecture constitutes a multi-blackboard system, including more than 40 asynchronous modules coded in C, C++, Java and Prolog. SmartKom attempts to address the problems of uncertainty and ambiguity that typically plague the analysis of multimodal input. In order to do this, several sets of scored hypotheses are produced by the various components of SmartKom. The SmartKom platform supports both multimodal fusion and multimodal fission processes. SmartKom's modality fusion component attempts to reduce the uncertainty of the analysis results by performing unification of all scored hypothesis graphs and applying mutual constraints. The presentation planner is used to control modality fission by specifying presentation goals for the text, graphics and animation generator. The developers of SmartKom hope to provide a kernel system that can be utilised within several application scenarios.

The Interact project [9] aims to explore natural human-computer interaction. An information storage knowledge base acts similar to a blackboard, with other system components accessing the knowledge base via the Information Manager. An unlimited amount of modules can be added to the system and the Interaction Manager deals with all connections between the modules. The architecture also allows the system to be distributed over multiple computers. Agents within Interact have different capabilities and the most suitable agent to perform a given task can be selected dynamically at runtime. Evaluators are used to determine which agent is best suited to deal with a particular situation. The decision-making process involves the evaluator giving a score between zero and one to each of the agents. A score of zero indicates the agent is not suited to the situation, a score of one means the agent is deemed perfectly suited, whilst values between zero and one indicate the degree of suitability. Several evaluators are used to give scores to an agent, indicating its suitability for use in the current situation. Scaling functions can be used to give greater importance to certain evaluators, before the scores are multiplied to give final scores (or suitability factors) for each of the agents. Several evaluations are performed before an agent is chosen for a particular task. Interact uses an XML-based method of semantic representation and, through the use of a shared knowledge base, implements a blackboard-style method of semantic storage.

Following on from previous research in Ymir [10], the Psyclone platform [3] enables software to be easily distributed across multiple machines and allows communication to be managed using rich messages. Psyclone is discussed in more detail in section 2.3. Other work in this area includes CHAMELEON [5], a blackboard-based architecture for processing multimodal data, COMIC [11], a multimodal dialogue system and XWAND [12], a user interface for intelligent spaces which also uses Bayesian decision-making.

2.3 Psyclone

Psyclone [3] is a message-based middleware that enables large distributed systems to be developed. The Psyclone platform enables modules to be implemented and tested without the developer needing to worry about issues of communication and dataflow. An overview of the Psyclone platform is illustrated in Fig. 2.

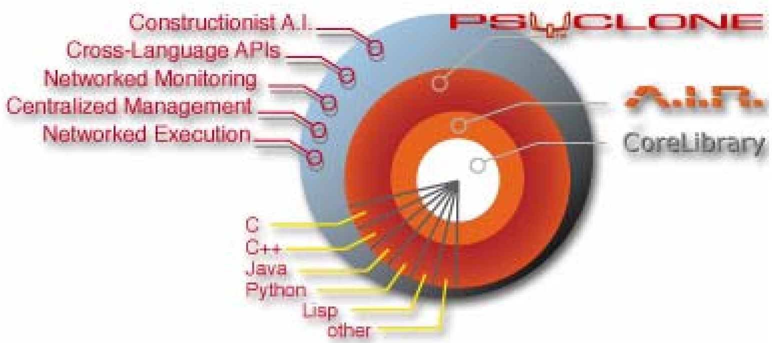

Fig. 2. Overview of Psyclone [3]

Psyclone introduces the concept of a whiteboard. Whiteboards are an extension of the blackboard model and allow heterogeneous systems, running on different computers, to be connected together. The whiteboards in Psyclone effectively act as publish/subscribe servers. Information is both posted on the whiteboard and dispatched from the whiteboard to all modules subscribed to that type of information. Communication is achieved using the OpenAIR Specification [3].

An XML specification file called *psySpec* is used to initialise values at start-up, specifying the setup of the system's modules. It is also possible to create multiple whiteboards and multiple modules within a system. Multiple programs can communicate with each other via the whiteboards using plugs written in various programming languages. Psyclone allows the developer to 'see inside the system' at runtime using a *psyProbe*. A *psyProbe* is a built-in monitoring system that allows developers to monitor all activities of the system. Developers can view time-stamped information on whiteboards and the content of individual messages using a standard web browser. The system architecture of Psyclone is illustrated in Fig. 3.

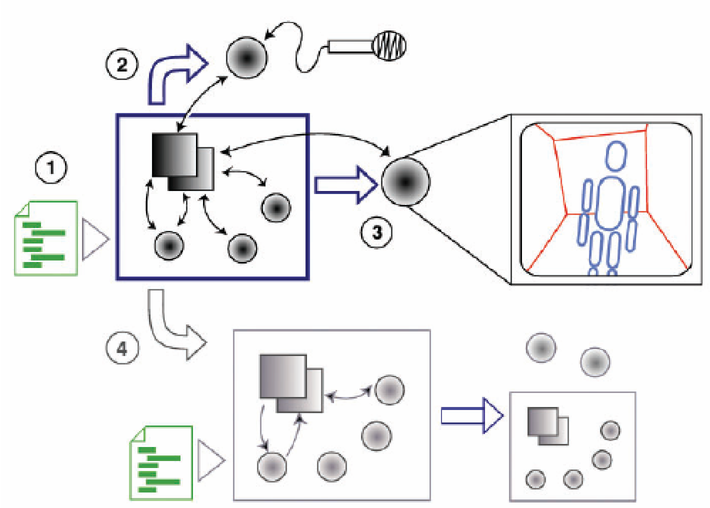

Fig. 3. Architecture of Psyclone [3]

When Psyclone starts it first reads the *psySpec* as illustrated by step (1) in Fig. 3. Then, any internal or external modules are started, such as speech recognition (2) and computer graphics (3). Psyclone then sets up appropriate subscription mechanisms for the modules and can be configured to automatically start other Psyclone servers as indicated by step (4).

Psyclone also introduces the concept of contexts. Contexts in Psyclone are globally announced system states that are used to help manage the runtime behaviour of the modules. Each of the modules in Psyclone is assigned at least one context and the module will not run until one of its contexts becomes true. Contexts allow individual modules to change their behaviour to meet the overall requirements of the system. Modules can be configured to do completely different things in different contexts, thus reducing the number of separate modules that a system will need to implement.

3 System Architecture

MediaHub's system architecture is presented in Fig. 4. The key components of MediaHub are:

- Dialogue Manager
- MediaHub Whiteboard
- Decision-Making Module
- Semantic Representation Database

MediaHub will use Hugin [2], a software tool for creating Bayesian networks, for decision-making. Psyclone is used for distributed processing in MediaHub. A blackboard-based method of semantic storage is implemented using the MediaHub Whiteboard within the Psyclone platform. Input to the system is in the form of marked up multimodal data (e.g. XML format). The Dialogue Manager facilitates the interactions between the other components of MediaHub. The MediaHub XML Specification file is read by Psyclone at start-up and defines the setup of the MediaHub modules.

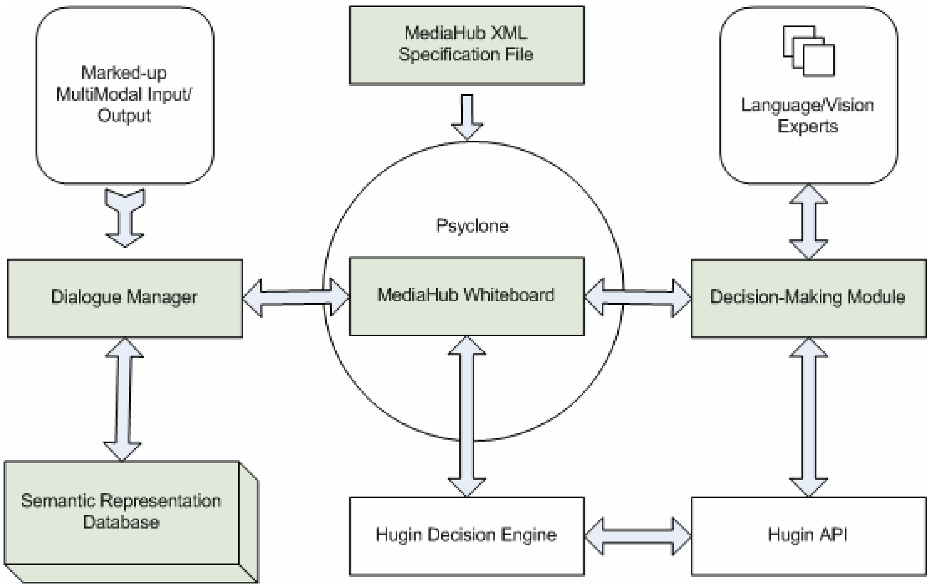

Fig. 4. Architecture of MediaHub

The Decision-Making Module accesses the Hugin Decision Engine through the Hugin API (Java). As illustrated in Fig. 4, the Decision-Making Module can also liaise with external language/vision experts such as language and vision processors. The Semantic Representation Database will contain information relating to the semantics of multimodal input and output data in XML or EMMA [13] format. Communication between the modules of MediaHub will be achieved using the OpenAIR specification implemented in Psyclone. MediaHub will utilise the decision-making capabilities of the Hugin API [2], coupled with Psyclone's distributed processing technology to provide an intelligent multimodal distributed platform hub.

4 Distributed Processing in MediaHub

MediaHub utilises the Psyclone platform for distributed processing. Psyclone implements a whiteboard, through which all modules in MediaHub communicate. Psyclone's built-in OpenAIR specification [3] provides a protocol for communication. MediaHub uses a *psySpec* specification file to declare the existence of modules and the type of information that the modules wish to subscribe to. Modules within MediaHub subscribe to a certain type of message using the syntax defined in the OpenAIR specification. For example, the Decision-Making Module registers with the MediaHub Whiteboard for messages of type *dm.input.language*, *dm.input.vision*, etc. To do this the following syntax is used in the specification file:

```
<triggers from="any" allowselftriggering="no">
  <trigger after="100" type="*input*"/>
</triggers>
```

Asterisks are used as a 'wild card'. The above three lines of XML requests that the associated module be 'triggered' by all messages that appear on the whiteboard containing the word 'input'. The above code effectively subscribes the module to all messages relating to input. Any messages that appear on the whiteboard relating to input will now be automatically sent to the Decision-Making Module. In addition to subscribing to receive messages of a certain type, a module may also define the type of messages it will send to the whiteboard. For example, the following XML code causes a module to post a registration message to the MediaHub Whiteboard:

```
<posts>
    <post to="MediaHub_Whiteboard" type="dmm.register" />
</posts>
```

This code is again entered into the *psySpec* which is run automatically when Psyclone is started. Starting Psyclone subsequently starts MediaHub, since all the modules of MediaHub are defined in the *psySpec* specification file.

5 Hugin

Hugin [2] is a software tool for creating and implementing Bayesian networks [14] and influence diagrams for decision-making. Hugin allows developers to create Bayesian networks using its Graphical User Interface (GUI). The GUI provides an easy to use interface to the Hugin Decision Engine. The Hugin GUI is illustrated in Fig. 5. Hugin also allows developers to access Bayesian networks from their applications using the Hugin API. APIs are available in C, C++, Java and as an ActiveX Server for use with Visual Basic. The Hugin tools have been designed to run on the Windows 2000/XP, Solaris, Linux and MAC operating systems.

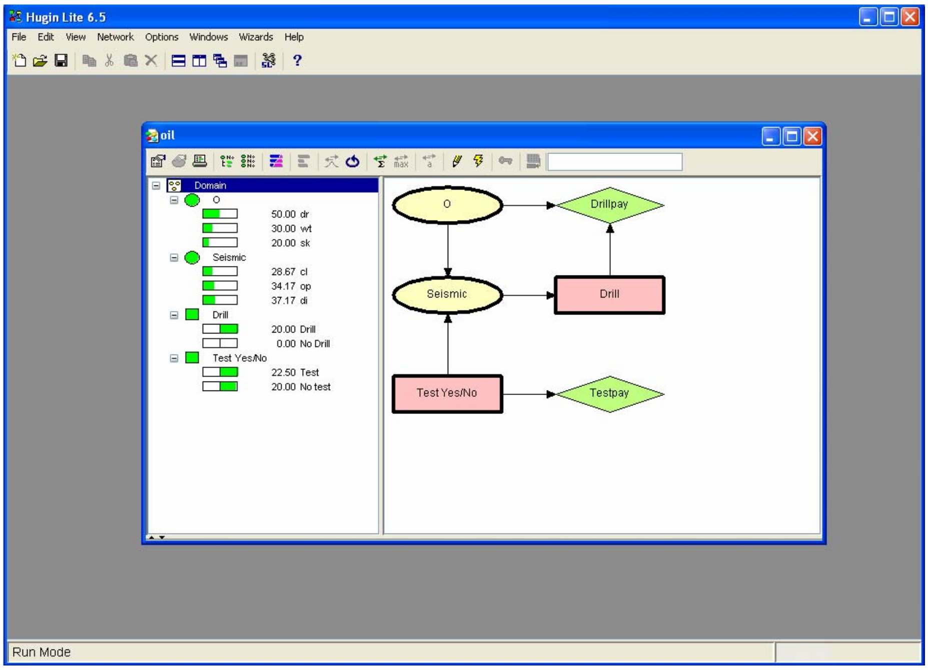

Fig. 5. Hugin GUI

The Hugin software tool enables the creation of Bayesian networks to model real world scenarios. When the Bayesian networks have been constructed, evidence can be entered to the model and the results are automatically re-calculated. Hugin computes Bayesian calculations and determines new probabilities when new information has been added. The Hugin software has been used in various applications to provide reasoning and decision-making. Applications include risk prediction, information management, safety assessment and medical diagnostics.

6 Decision Making in MediaHub

The Hugin software tool [2] will be used for Bayesian decision-making in MediaHub. Decisions taken within MediaHub will relate to both multimodal input and output. Decisions on multimodal input will primarily be related to the resolving of ambiguity. For example, in situations where ambiguity occurs in one input modality, another input modality may be used to resolve the uncertainty. Other decisions on the input will be needed to determine the semantic content and to fuse semantics of respective input modalities. Decisions on multimodal output will mainly relate to the choice of output for a particular situation. For example, some information may be better presented using graphical output (i.e. a map showing directions to the airport), whilst other information may be better presented using speech output (e.g. feedback to the user such as "ok" or "I don't understand"). Other decisions on the output could relate to synchronisation of multimodal output (e.g. matching a sequence of images to corresponding speech output).

7 Conclusion and Future Work

The design rationale for MediaHub, an intelligent multimodal platform hub, has been presented. We have given an overview of the current system architecture and explained how the Psyclone platform will be used for distributed processing. The role of the Hugin software tool, in providing MediaHub with a new approach to decision-making based on Bayesian networks, has been outlined. A brief review of the functionality of both Psyclone and Hugin has also been given.

Key considerations for future work include the semantic representation of multimodal data and the implementation of Bayesian decision-making using the Hugin API. Another topic currently being addressed is the structure and content of suitable multimodal data that can be used in the testing of MediaHub. In summary, this paper provides a report on the design of MediaHub and outlines future development.

Acknowledgements

The authors would like to thank Frank Jensen for his help with the Hugin software, Kristinn Thórisson and Thor List for their support with Psyclone and the OpenAIR specification. We would also like to thank Jean-Claude Martin and Michael Kipp for their advice on multimodal corpora.

References

1. ViaVoice http://www.nuance.com/viavoice/
2. Jensen, F.: 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 (1996)
3. Thórisson, K.R., List, T., Pennock, C., DiPirro, J. Whiteboards: Scheduling Blackboards for Semantic Routing of Messages & Streams, AAAI-05 Workshop on Modular Construction of Human-Like Intelligences, Twentieth Annual Conference on Artificial Intelligence, Pittsburgh, PA, July 10, 16-23 (2005)
4. Fink, G.A., Jungclaus, N., Kummert, F., Ritter, H., Sagerer, G.: A Distributed System for Integrated Speech and Image Understanding. International Symposium on Artificial Intelligence, Cancun, Mexico, 117-126 (1996)
5. Brøndsted, T., Dalsgaard, P., Larsen, L.B., Manthey, M., Mc Kevitt, P., Moeslund, T.B., Olesen, K.G.: The IntelliMedia WorkBench - An Environment

for Building Multimodal Systems. In Advances in Cooperative Multimodal Communication: Second International Conference, CMC '98, Tilburg, The Netherlands, January 1998, Selected Papers, Harry Bunt and Robbert-Jan Beun (Eds.), 217-233. Lecture Notes in Artificial Intelligence (LNAI) series, LNAI 2155, Berlin, Germany: Springer Verlag (2001)

6. Vinoski, S.: Distributed object computing with CORBA. C++ Report, Vol. 5, No. 6, July/August, 32-38 (1993)
7. Cheyer, A., Julia, L., Martin, J.C.: A Unified Framework for Constructing Multimodal Experiments and Applications. In Proceedings of CMC '98: Tilburg, The Netherlands, 63-69 (1998)
8. Wahlster, W.: SmartKom: Symmetric Multimodality in an Adaptive and Reusable Dialogue Shell. In: Krahl, R., Günther, D. (Eds.), 47-62, Proceedings of the Human Computer Interaction Status Conference, June. Berlin, Germany: DLR (2003)
9. Jokinen, K., Kerminen, A., Kaipainen, M., Jauhiainen, T., Wilcock, G., Turunen, M., Hakulinen, J., Kuusisto, J., Lagus, K.: Adaptive Dialogue Systems – Interactions with Interact. In Proceedings of the 3rd SIGdial Workshop on Discourse and Dialogue of ACL-02, Philadelphia, PA, July 11-12, 64-73 (2002)
10. Thórisson, K.R.: A Mind Model for Multimodal Communicative Creatures & Humanoids. In International Journal of Applied Artificial Intelligence, Vol. 13 (4-5), 449-486 (1999)
11. Foster, M.E.: Corpus-based Planning of Deictic Gestures in COMIC. Student session, Third International Conference on Natural Language Generation (INLG 2004), Brockenhurst, England, July, 198-204 (2004)
12. Wilson, A., Shafer, S.: XWand: UI for Intelligent Spaces, In Proceedings of the SIGCHI conference on human factors in computing systems, 545-552 (2003)

13 EMMA http://www.w3.org/TR/2004/WD-emma-20041214/

14. Jensen, F.V.: Bayesian Graphical Models, Encyclopaedia of Environmetrics, Wiley, Sussex, UK (2000)