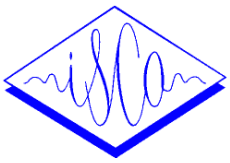
ISCA Archive
http://www.isca-speech.org/archive



6th International Conference on Spoken Language Processing (ICSLP 2000)
Beijing, China
October 16-20, 2000

A SYSTEM FOR THE RESEARCH INTO MULTI-MODAL MAN-MACHINE COMMUNICATION WITHIN A VIRTUAL ENVIRONMENT

Andrew Breen[1], *Barry Eggleton*[1], *Gavin Churcher*[2], *Paul Deans*[2], *Simon Downey*[2]

1. University of East Anglia, Norwich, England
2. BTexaCT Adastral Park, Martlesham Heath, Ipswich , England

ABSTRACT

This paper reports on work currently under development jointly at the University of East Anglia, School of Information systems and BT. The aim of the work is to design and develop advanced demonstrators which are capable of holding natural multi-modal discourse with a human interlocutor. In addition, these demonstrators will investigate such interactions within a computer generated immersive environment in which both the computer agents and human users have a virtual presence. The paper presents a design for a distributed architecture to achieve this goal and brief introductions to many of the system components.

1. INTRODUCTION

There are many approaches adopted by researchers when attempting to develop a more natural and intelligent user interface; most are directly influenced by the envisaged application domain. For instance, researchers working on automated telephony and multi-media services normally develop systems based around the concept of a dialogue manager. Such systems tend to produce highly structured and domain dependent dialogues. Speech translation researchers, while still limited to domain specific applications, through necessity, tend to approach the interpretation of the input signal using more theoretically motivated methods. Such systems often rely on both a deep and shallow semantic / syntactic interpretation of the input. AI researchers interested in producing human like responses from a machine which are not domain specific use a wide variety of techniques ranging from shallow semantic analyses to the very deep analyses. The choice, in part, depends on the whether systems are being developed to investigate specific aspects of semantic theory or to engage the user in an unstructured discourse.

This paper reports on a design that uses a combination of approaches. Ideas taken from methods of deep semantic analysis[1] will be combined with the more pragmatic approaches undertaken by researchers and games developers who are attempting to provide a dynamic interaction[2,3]. Specifically, the paper will report on investigations into the feasibility of using methods typically employed in unstructured discourse to develop a more natural style of interaction.

The human-computer interaction will take place within an immersive virtual environment. In this environment, human users and autonomous agents take control of full-body avatars capable of complex interaction with each other and other entities within the environment. Users may control their avatars using a variety of input devices for looking around, moving and manipulating objects. The agents are autonomous and may be commanded to perform actions through speech and gesture. For example “walk over there” would instruct the agent to walk to the location indicated by the gesture. With commands of this type, the user’s avatar would typically follow the agent automatically; similarly, should the user move away, the agent would follow the user. Discourse will be of mixed-initiative and will generally result in the user obtaining some information, or some interaction with the environment taking place. The constraints imposed on the user by a controlled immersive environment will provide sufficient implicit structure to enable effective directed interactions, both verbal and non-verbal, and allow the agent to become a useful expert on its environment.

This work is a result of experience gained through the development of previous systems such as Maya[4]. In particular, architectural improvements have been made that allow a larger number of researchers to independently develop and test sub-systems of components for subsequent integration. Third-party software has been replaced by in-house software to allow improved low-level control and optimisation of time-critical components. The current system uses a flatter architectural approach effectively decentralising many processing tasks, allowing better parallelism of execution and the easy duplication of sub-systems for multi-user multi-agent configurations. Also the virtual world is now integral to the system and able to provide useful context information to many of the recognition, synthesis, understanding and reasoning processes. Significant research is also being drawn from the MUeSLI project[5] to aid in the time-alignment and unification of speech and gesture.

The system is currently known as IMDUI (Intelligent Multi-model Dialogue User Interfaces). This paper outlines the current design of the architecture and the expected operation of the system.

2. ARCHITECTURE

The IMDUI system has been designed to operate distributed across a network using CORBA[6]. The CORBA approach to distributed computing allows client and server objects to inter-operate via Object Request Brokers (ORBs) across different platforms, network protocols and source languages. This

provides an enormous level of flexibility, and facilitates the integration of work from many fields of AI into the system. Furthermore, the computational requirements of the system can not be met by a single processor, and are increasing rapidly as more components are added.

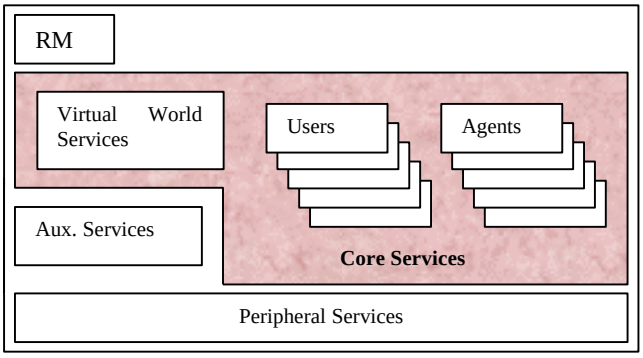


Figure 1: System classification

2.1 System Distribution

The system consists of a number of interconnected services under the administrative control of the Resource Manager (RM). The RM is responsible for system configuration, initialisation, activation and deactivation. It governs which services are started, where they execute on the network, and how they are interconnected. Once the system is running, the RM lays dormant until required for system shutdown.

The services fall into three classes: core, auxiliary and peripheral and this is shown in Figure 1.

Core. The core services are those which perform the primary information manipulation and processing within the system. Currently, the Virtual World services, one or more agents, and one or more users. Core services may be local or remote to a system and may act as multi-client servers.

Auxiliary. Auxiliary services provide communication and infrastructure for the other services. For example, Event Channels provide asynchronous communication buffers between services, typically used to transfer audio data.

Peripheral. The peripheral services provide input to and output from the system's core services. They form the immediate user interface and must be duplicated for each user of the system. They include audio I/O for speech capture, speech synthesis and spatialised audio; display interfaces and input devices.

2.2. Information Flow

The complete system consists of four major interconnected sub-systems with three main information streams between them (figure 2). The discourse sub-system performs speech and gesture unification, parsing, interpretation, inference, knowledge representation and text generation. This sub-system is duplicated for each agent present in the system. Human-computer discourse, results in commands being issued to the planning sub-system in the form of imperative case-based structures. Planning goals are formed from these structures and appropriate plans generated. For example the command:

"show me object1"

will become the planning goal:

facing (user, object1).

The planner sends lower-level commands to the behaviour sub-system until the plan has been executed or is no longer required. The behaviour sub-system controls all entities within the virtual environment in terms of form, function, movement, animation and audio.

User input will originate from a variety of sources at a level applicable to the entry point into the system. The user will use natural language combined with pointing gestures; context sensitive menus and maps; direct manipulation of environment entities through haptic and virtual reality input devices; or a combination of all three.

Event feedback drives the autonomy of the system. Events are generated by services within the virtual environment sub-system or by user-interaction. They are used to trigger further behaviour, continuation of plan execution, and modifications to the knowledge represented within the discourse sub-system.

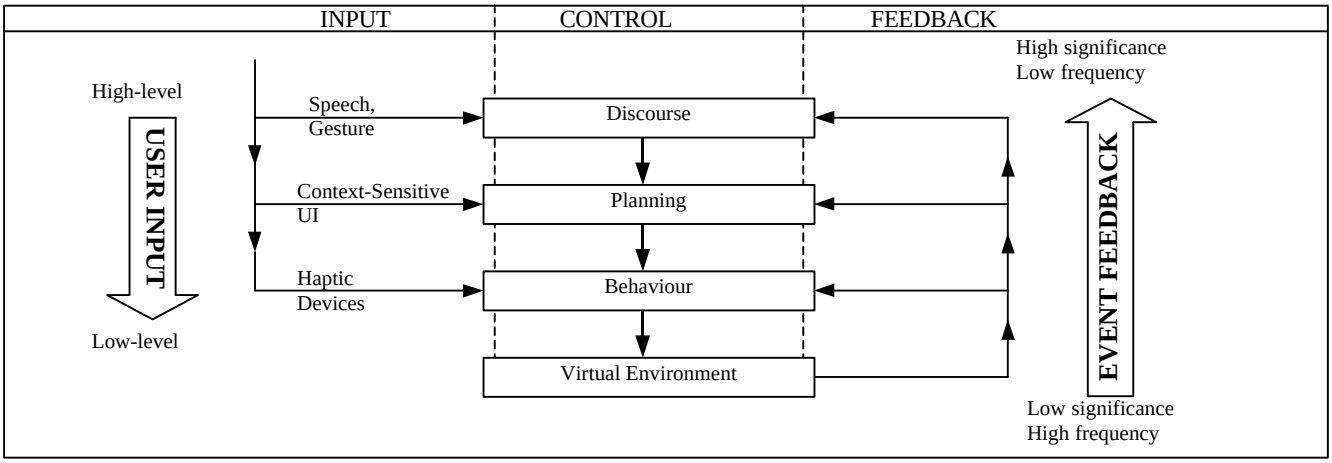


Figure 2: User Interaction with World Autonomy – a hierarchical model of control and feedback

3. SYSTEM COMPONENTS

This section describes the discourse, virtual environment, behaviour and planning sub-systems in more depth.

3.1. Discourse

All communication with an agent by other parts of the system will be through four exposed interfaces: an auditory input channel to send captured speech to an agent's recogniser; a gesture input channel; an event channel; and an interface to return detailed information about the agent. An agent's discourse sub-system consists of a number of sub-ordinate services. Each agent operating within the system has its own set.

Active Lexicon. The Active Lexicon (or ALEX) is the primary knowledge base of an agent, and represents an agent's personal view of the world and the events taking place within it. ALEX holds an agent's current and past beliefs on the properties of and relationships between objects and events, and the significance of this information. ALEX is principally a frame based semantic network, but entity relationships are assigned a scalar value indicating the degree to which the relationship is believed to be true. When the system is initialised, ALEX is loaded with all knowledge about the world, including spatial, linguistic, lexical, paralinguistic, object, event and domain specific information. This forms an agent's base-line knowledge and is identical for each agent in the system. When the system is activated, an agent's ALEX is modified and extended to reflect its own personal experiences within the world, as such an active lexicon quickly diverges from its initial base-line configuration. An ALEX is modified in response to events passed up from the world, and to the results of reasoning and inference by other system components.

Event controller. Communicative events are signalled to the agent via one of the peripheral services. For example, a button press may signal forthcoming speech interaction. Whenever an interaction is requested an event is generated and passed to the event controller. The event controller informs the appropriate input services (recognition, gesture integration and input unification) which prepare to process information. An event will contain information about the requesting interlocutor for use within multi-user, multi-agent systems - an agent entering into a conversation with another user or agent must know who they are talking to. The initiation of a new dialogue will require certain actions to be performed by the agent, for example turning, facing and greeting the new interlocutor. These types of sequences are generated by the planning sub-system.

Recognition and Parsing. Once the recognition service has received a recognise signal from the event controller, it listens for speech on a speech event channel, recognises, parses and places the result on to a processed queue where it is made available to the speech/gesture unification component. It is proposed that the output of the recognition service should be a structured network and that the structure used within the network should be based on a case-based parse of the input.

Gesture integration. Like the recognition server, the gesture integrator, once it has received a signal attempts to read data from the gesture event channel. Raw gesture information will be interpreted as specific gesture types. The result will be placed on a processed queue for the unification component. The structure returned by the gesture recogniser may also be in the form of a relationship table.

Unification. The unification component takes the processed data from the recognition and gesture queues and attempts to construct an unambiguous interpretation. To achieve this, it will utilise information from a variety of sources: the user's local environment, for example objects or locations within the user's field of view or immediate vicinity; the discourse history for information relating to previous input or topic; and the ALEX for specific properties of objects, including form, function and semantic information. If after unification no unambiguous solution can be found, an input structure is constructed and sent to the inference service.

Inference / Interpretation. The inference service takes the output of the unification process and attempts to postulate an appropriate response. It will begin by attempting to deduce the basic form of the interaction: whether it is a direct or indirect command, a statement or a query. In the case of direct commands, this service may do comparatively little and pass a case-relationship table to the planner via the generation service. Indirect command handling is subject to further research to produce a method for their recognition and translation into simpler speech acts. Information contained in statements is transferred into the ALEX. A request for information must be resolved through an analysis of the question and the recent discourse history. If all the information needed to respond can be deduced from the relevant knowledge bases, a response using this information will be generated.

Discourse History. The discourse history service retains information on all discourse performed by the agent and attempts to keep track of the general and specific topics. In many ways it will act as a short-term memory for an agent.

Text Generation. The generation service will attempt to produce syntactically and semantically appropriate dialogue. As the complexity of this service increases, it will be sensitive to discourse history and emotion, and capable of generating mixed speech and gesture responses[7].

3.2. Virtual Environment

The virtual environment sub-system encompasses all multi-modal output technologies and the servers that control and query them.

World. The world server manages all entities that a user will see displayed. It is composed of a structured hierarchy, or scene database, of entities such as avatars, objects and scenery. Each

entity has a number of properties including position, orientation, velocity, angular velocity, shape, texture and animation state. There will only be a single world service in a system.

Display. Each user of the system has a display service. Each display service accesses the scene database held in the world service at high speed to render the environment from the user's avatar's point of view in real-time.

Spatial. The spatial service provides dynamic translation between the continuous representation of the world server's scene database and the semantic representations used in the ALEX and other high-level services. The spatial service periodically queries the scene database and generates useful spatial relationships between entities. Relationships such as *TouchProximity(X, Y), Facing (X, Y), FieldOfView (X, Y)* are used to construct a semantic state of the environment, the changes to which are transmitted as events to the Behaviour service, the Planner and each ALEX.

Event Filter. Event filters are placed at certain points along the event channels. They are dynamically configurable and may be used to filter out high-frequency or irrelevant events. For example, an agent may only be interested in events occurring in its immediate vicinity.

Avatar and Object. The avatar service directly controls the movements of avatars within the world service, including motion-captured animations such as walking, and dynamic calculative motions such as manipulating objects. Similarly, the object server controls doors, lights, and moveable objects, for example. These servers both generate events signalling to the rest of the system what has or is being done.

Text-To-Speech Synthesiser. The speech synthesiser used is BT's Laureate[8], the output from which is sent via event channel to the audio service. This service also generates events indicating what has been said.

Audio. The audio service provides spatialised 3d audio from the perspective of the local user. It receives audio data from multiple speech synthesisers and can play local sound files. It periodically polls the world server for the updated positions and orientations of its sound sources.

3.3. Behaviour

The behaviour sub-system handles rule-based autonomy of the virtual environment and programmable event distribution. This behaviour is governed through a set of models which interact with a dynamically updated state-based semantic representation of the environment. The representation is updated in response to events received from the virtual environment sub-system and the peripheral services. The models consist of an antecedent, an action, and a consequent. If the current state causes an antecedent to evaluate true, then the corresponding action is sent as a command or event to a specified system service and the consequent is any state change expected. For example, a model might say that if a user faces and is close to an agent, then the agent will turn to the user and initiate a discourse. If more complex behaviour is required the model can pass raw event data to a service capable of interpreting it. The models use type-based variable unification to allow a single model to be relevant to many states. The *types* of all entities within the environment are stored in the entity hierarchy of the *Global Active Lexicon***.** This functions in a similar way to an agent's ALEX, but there is only one per system and it holds a true representation of the world as opposed to the specific beliefs of an agent.

3.4. Planning

The planning sub-system receives requests for plans in the form of imperative case-based structures. These are translated into planning goals in terms of required state changes within the environment. The planner uses the behaviour models to backtrack from the goals forming partially-ordered plans which are executed by triggering state changes within the behaviour engine.

4. CONCLUSION

This paper has presented an overview of the design of the IMDUI system. The system is currently being implemented breadth-first as a demonstrable prototype. The correct concurrent flow of information around the system is initially more important than any one service being completely implemented. As such, many system components are still in the early stages of design.

5. REFERENCES

1. Ed. Smith G., Steele N., Albrecht R., *Artificial Neural Networks and Genetic Algorithms*, Published by Springer-Wien, New York, 1998.
2. EON Technical While paper: http://www.eonreality.com/
3. Realimation Technical documents: http://www.realimation.com/overview/technical/
4. Downey S., Breen A., Fernandez M., Kaneen E., *Overview of the Maya Spoken Language System*, International Conference of Spoken Language Processing, Sydney 1998.
5. Wyard P.J., Churcher G.E (2000), *All Channels Open – Multimodal Human/Computer Interfaces*, BT Technology Journal (Millennium Edition), vol. 18, no 1, Jan 2000.
6. Object Management Group, *The Common Object Request Broker: Architecture and Specification* Revision 2.2 (February 1998). http://www.omg.org/corba/corbiiop.htm
7. McKeown, K., *Text Generation*, Cambridge University Press, 1985.
8. Page J.H., Breen A.P., *The Laureate Text-to-Speech System – Architecture and Applications*, BT Technology Journal:14:1, 1996.