OVERVIEW OF THE MAYA SPOKEN LANGUAGE SYSTEM

Simon Downey, Andrew Breen, Maria Fernandez, Edward Kaneen

Email: simon.downey@bt.com

BT Labs, Martlesham Heath, Ipswich, UK

ABSTRACT

Recent developments in distributed system processing have opened the doors to the running of highly complex systems over a number of networked computers. This enables the complexity of a system to be hidden behind a small, lightweight user interface - for example a downloadable web page. The Maya system makes use of such interfaces to combine the functionality of speech recognition, synthesis robust parsing, text generation and dialogue management into a highly flexible multimodal architecture, working in real time.

This paper describes the development of the architecture and interfaces to each system component. The configuration of the system to particular tasks is discussed, making use of an email secretary task as an example. Once configured, the system is able to adopt all the functionality of a conventional email system and extend these capabilities by allowing complex queries to be made about mail messages.

1. INTRODUCTION

A significant component of any intelligent environment is the human - machine interface. It is highly likely that in the future such an interface will, for the majority of applications, closely model human to human communication. In fact we may expect that the human - machine interface will increasingly mimic the behavior and appearance of humans.

Two years ago BT set up the Maya project. The aim of this project was to research into the spoken language and multimodal aspects of such an interface and to provide an effective computational research framework. Due to scale of the problem, Maya collaborates closely with a number of other groups within BT, these include speech synthesis [1], recognition [2], understanding [3] and virtual humans [4]. The project is developing an infrastructure that enables advanced research to coexist with demonstrable solutions and has designed an environment for an email application, described further in [5].

The eventual goal of the work is to construct a system which will communicate with people in a way that is both natural and pleasant for its human users. Such a goal is a long way off, and is unlikely ever to be achieved simply by bolting together "off-the-shelf" components. Clearly no single machine or program can currently hope to accommodate this degree of complexity - the solution is to provide component services, which exist as part of a distributed computer system. These component services (e.g. speech synthesis, recognition and understanding) exist independently of any particular application, on a variety of computers and are written in a number of different computer languages. The services communicate through a unifying standard interface language (IDL). Currently the system uses the Object Management Group's (OMG) Common Object Request Broker Architecture (CORBA).

One practical target for the system was that it should be able to work in real time – responses to user requests must not generate unacceptable pauses or system delays. In order to achieve this, an event-based system architecture was developed. This architecture allows individual components of the system to run in parallel and allows the system to continue with other tasks if a request is made which could involve a long or indeterminate transaction time (for instance a query to a remote database). The event driven nature of the system raises issues of sequencing and concurrency, which are dealt with in the paper.

The communication between layers and services in the Maya system is then discussed through means of an example interaction with the system.

2. THE MAYA SYSTEM

This section outlines the system's architecture and gives a brief overview of the different components. Maya consists of four Functional Layers: the User Layer; the Presentation Layer; the Strategy Layer and the Information Layer. Communication between layers take place via event channels, which are effectively queues storing the information passing between them. It is the interaction between the layers that makes possible an effective dialogue with the user. A schematic view of the system is shown in Figure 1.

Each layer comprises a set of co-operating services. These services come in two forms, core services (labeled *agents* in the diagram) and component services. The agents act as facilitating services helping control the flow of information through the system and co-ordinate the behaviour of component services. The number and type of component services available within a particular configuration of Maya will vary depending on the modality of the application being built. As a minimum, the Maya system will contain speech recognition, parsing, dialogue and speech generation services. Each of these component services may in turn be composed of a number of co-operating sub-processes. For example, the speech generation component within Maya has at its core, BT's Laureate text-to-speech system [1].

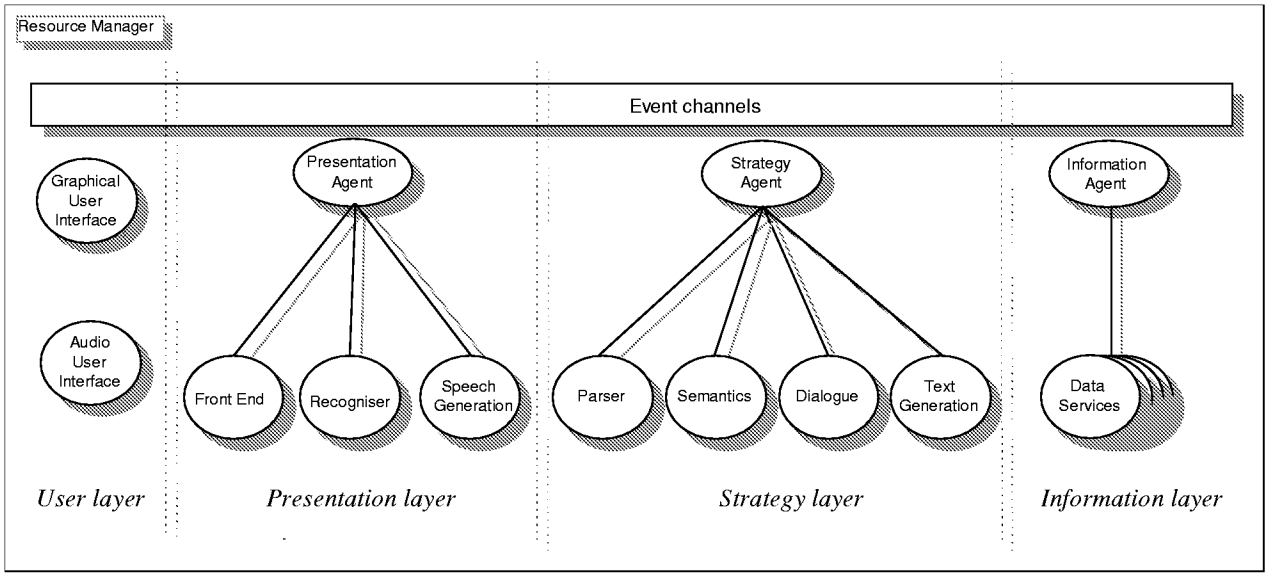

Figure 1 Overview of the Maya System Architecture

The User layer facilitates the interaction of the user with the system. This interaction may use multiple input and output modalities. For input, text and mouse are supplied via the graphical user interface (GUI), and speech via the audio user interface. Accompanying the input modalities, a face recognition component is available for logging the user on to the system.

The output modalities are text, graphics and again, speech. In addition to these output possibilities, combining both graphical and auditory data is a talking head component (not shown).

The Presentation layer is responsible for presenting data to, and retrieving information from the user layer via input and output queues. The component services in the Presentation layer include:

- An audio Front End Component. Incoming speech data from the audio interface is routed here for parameterisation before being fed on to the recogniser. Utterances are also recorded to allow user options such as playback/ re-recognition.
- The Recognition service. Passes the speech data to the core recognition engine(s). Various forms of recognition output are used by other parts of the system: the top result is passed back to the user layer to be displayed by the GUI, whereas the parsing components of the system require a full acyclic directed graph output (see section 4.2).
- The Speech Generation Component. Passed an input of SGML marked-up text from the strategy layer, this component uses text-to-speech synthesis to generate the system response, as well as rendering lip/face movements for replies sent to the Talking Head service.

The Strategy Layer is the "intelligent component" of the system, and consists of:

- The Syntactic Parser. This takes as its input, a graph of possible utterances from the speech recogniser. As recognition is an inexact process, the syntactic parser processes all the different possible word combinations that could have been recognised. The parser processes these combinations to produce an ordered list of possible utterances, together with their syntactic parses.
- The Semantic Component. This represents meaning by interpreting the information passed across from the Syntactic parser. This semantic representation forms the input for the Dialogue Manager.
- The Dialogue Manager. This is the core language processing part system, the main function of which is to produce a direct discourse with the user.
- The Text Generator. This produces marked up text from a semantic representation coming out from the dialogue manager.

The fourth and final layer, the Information layer, contains the actual data that the user wants to access. Requests to information services are made using standard database queries, for instance when the Maya system is configured for the email task, the information service is effectively a client to an IMAP server.

Sitting above all the system layers is a Resource Management service. This service has the responsibility of starting up and configuring the system to the user environment. The Maya system is designed for a multi-user environment, and as such the Resource Manager may choose to share certain core services amongst several users.

3. SYSTEM INTERFACES

One of the architectural aims for the Maya system was to make the design fully flexible, allowing a plug-and-play approach to the core components wherever possible. For this to be achieved, 'vendor independent' APIs were needed for each of the services, thus allowing the core 'engines' of each service to be swapped in a manner seamless to the rest of the system. This is made possible by distributing components of the system 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. The advantages of such a design are that individual services may be worked and tested in isolation, using the most appropriate machine or language, and new system components can be easily integrated. For instance, many of the components in the strategy layer are written in Prolog, being the standard programming language for natural language programming, whereas presentation components are written in C or C++ as these languages are better suited to processor intensive tasks such as recognition or text-to-speech synthesis.

The resulting interfaces, written in IDL [6], are very simple and share many common commands – such as *configure*, *start* and *stop*. To replace an existing core service, all that is required is to map each IDL command to one providing the same function in the new service, thus providing a CORBA 'wrapper' around the service. Most IDL compilers will automatically generate skeleton interfaces for these mappings.

3.1. Information Flow

Information can be represented in several ways in the system. The basic type of information is called an 'event', and may be generated by any of the services. An event will typically represent a packet of speech data, a marked up text phrase or a database query. Events are passed from one core service to another via the event channels.

The system also passes around 'messages', these are typically generated by component services to indicate when a task has been achieved or when they have changed state (for example when the recognition service has returned a result). System messages are usually dealt with by the Maya agents, which act upon the message by generating further messages/events to update the system with the new information appropriately. For the recognition result example, the message will be sent to the Presentation agent which may then choose to display the recognition result on the User GUI, tell the Front End component to stop sending data to the recogniser and send the strategy layer any other information which may be relevant to the current utterance.

3.2. Concurrency

For distributed systems with multi-modal input and output capabilities, the issue of data sequencing arises. For example, if a response to a user request requires some speech data to be played via a talking head component, and other speech/sound data to be produced using a TTS service, the system needs to ensure that the events are played in the correct order, and do not overlap. Sequencing is also dealt with by the system agents. Each message and event carries a time stamp, and it is the responsibility of the agents to interpret these when playing out and routing information around the system.

This still leaves the problem of how long events that occur closely in time may be considered 'concurrent'. For example, a user looking at several objects on a screen may utter a request of the form 'put that there'. In addition to the speech data being sent to the system, additional events will be generated by the GUI to indicate what *that* and *there* refer to in the present context (the objects may have been selected via mouse clicks or a touch screen for instance). As the recognition result is likely to arrive several milliseconds after the location events, the system needs to determine whether the events can be considered 'concurrent'.

Although the Maya agents can deal with this problem heuristically to some extent, the semantics layer must then be able to interpret a set of concurrent events. In the above case, the timing information should be sufficient to determine what *that* and *there* refer to, given a correct transcription of the user utterance. If there is a recognition error, or in cases of ambiguity, the strategy component can make use of its knowledge of the conversation so far to generate a dialogue with the user and attempt to resolve the ambiguity.

4. USER INTERFACE

In Maya, the user is able to make requests to the system and obtain information concerning their electronic mail. The user can either make a query such as: "Do I have any new messages?" or give commands such as: "Please read me messages in April". The aim is to produce a system that can establish a full meaningful conversation with the user. This requires that the human-machine interface is able to take into account every feature that contributes to human dialogue, such as 'fluent' speech effects (i.e. utterances subject to phonological and phonetic effects) and prosodic modifications present in human-to-human communication such as hesitation and restarts.

Apart from dealing with human speech phenomena, a spoken dialogue system needs to attain coherence, consistency and conciseness in its conversation with the user. This is very important in dialogue because both participants need to produce their utterances in a coherent and co-operative way to achieve full communication between each other.

Maya is also responsible for keeping track of the user's requests in order to do ellipsis and reference resolution. The fact that Maya is aware of the point at which the users are in the dialogue is also important because it forces the user to finish any uncompleted tasks, and ensures that the dialogue is always coherent.

4.2. System Interaction

An example dialogue with the Maya system is shown in Figure 2. A user request *"List new messages?"* produces a directed graph output from the recogniser representing all the possible paths through the language model which were active at the end of the recognition stage.

This is passed to the robust parser, which produces a syntactic representation of the network in the form of an ordered list of

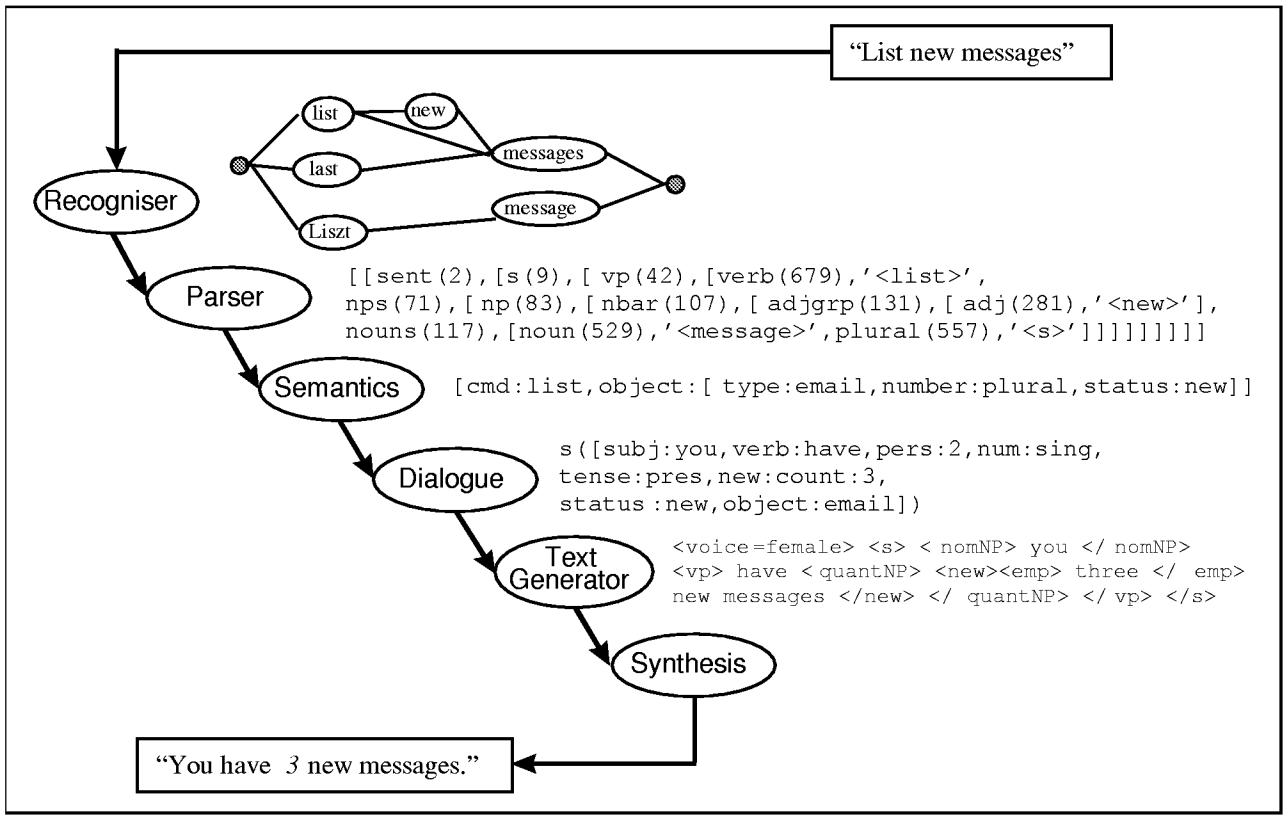

Figure 2 Example system interaction: Email configuration

possible parses. The list is ordered using 'higher level' knowledge sources than the recogniser, such as the frequency of occurrence of words and syntactic knowledge. A combination of standard network minimization and modified chart-parsing [5] techniques are used. The semantics service uses these utterances to build a semantic representation of the best possible input, which is represented as a set of feature-value pairs. A caseframe parsing technique is used [5].

The dialogue service then evaluates the meaning of the 'best' utterance in the context of the overall dialogue between the user and computer. The current dialogue state may require the dialogue service to query one of the information services to get information about the user's email, or perform some action.

The result of the user's request will be returned by the last strategy service, the text-generation service. Accepting feature-value pairs as its input, this service acts in reverse to the semantics service, by using a definite clause grammar to generate text from semantics. The output contains additional syntactic and pragmatic information, which is used by the synthesis service to improve the prosody of the response. In the above example, the semantics services are able to tag the parts of the response carrying 'new' information, which can then be emphasized by the synthesis component.

5. CONCLUSIONS

This paper has presented an overview of the Maya Spoken Language System. The architecture of the system has been specifically designed to allow research into the individual core services *and* to how these components integrate in to the system as a whole.

The system can presently be configured for task specific 'email' style applications. In future the direction of Maya will be towards a more domain independent implementation, and to improving the human-machine interface. Research in the former area will consider spoken utterances as a part of a relevant "discourse". The aim is to construct an advanced semantic and pragmatic representation, taking a 'world view' to understand meaning in context [5]. Improvements in the human-machine interface will investigate additional services with the intention of 'personifying' the system (i.e.. through use of the 'talking head').

6. REFERENCES

1. Page J.H., Breen A.P. *The Laureate Text-to-Speech System - Architecture and Applications.* BT Technol. J.:14:1. 1996.
2. Pawlewski. M, Downey S N et al. 1996 *Advances in Telephony-Based Speech Recognition*, BT Technol. J.:14:1. 1996.
3. Kaneen E & Wyard. P. A *Spoken Language Interface to Interactive Multimedia Services* IVTTA. 1997.
4. Breen. A. P., Bowers. E., Welsh. W. *An Investigation into the Generation of mouth Shapes for a Talking Head,* ICSLP'96
5. Breen AP, Downey S N, Fernandez M & Kaneen E *The Maya Multi-Modal Spoken Language Understanding System* Proc. 2nd International Workshop on Human Computer Conversation, Bellagio 1998
6. Object Management Group *The Common Object Request Broker: Architecture and Specification* Revision 2.2 (February 1998) [http://www.omg.org/corba/corbiiop.htm].