A Speech Interface to Virtual Environments

Scott McGlashan and Tomas Axling

Swedish Institute of Computer Science,
Box 1263, S-16428 Kista, Sweden
e-mail: *{scott,axling}@sics.se*

Abstract

Virtual reality has sometimes been thought of as embodying a return to a ‘natural’ way of interacting by direct manipulation of objects in a world. However, in the everyday world we also act through language: speaking is a ‘natural’ way of communicating our goals to others, and effecting changes in the world. In this paper, we discuss technical and design issues which need to be addressed in order to combine a direct manipulation interface to virtual reality with a speech interface. We then describe a prototype system based on intelligent agents which provide specialised functions in the virtual world. The agents have simple dialogue capabilities allowing users to directly control them with speech.

1. Introduction

A direct manipulation interface to a virtual world can be augmented with a spoken language interface so that users can give spoken commands to manipulate objects. In order to develop such a multimodal interface, a number of technical and design issues need to be addressed [7]. Three of the most important are:

Speech Recognition There is a trade-off between coverage and accuracy in speech recognition systems: the larger the user vocabulary and grammar, the greater the potential for recognition errors. How do we restrict the user' s language yet provide a comfortable interaction?

Language Understanding The interpretation of spoken commands is dependent upon context: while some utterances are sufficiently specific to identify which object they refer to, others require knowledge of the situation for their interpretation. What (limited) situational knowledge maximises spoken language understanding?

Interaction Metaphor Adding a speech interface changes the relationship between the user and system. With direct manipulation, the system is relatively transparent: the user is directly embodied as an actor in the virtual world. Speech, however, requires a dialogue partner: who does the user talk to?

We are addressing these issues by building systems with speech and direct manipulation interfaces to virtual worlds. In this paper, we describe the benefits speech could offer virtual reality applications (Section 2) and then describe a prototype system, with speaker-independent speech recognition, which allows agents in the virtual world to act as dialogue partners (Section 3).

2. Why add a Speech Interface to Virtual Reality Applications?

Speech interfaces are increasingly being used in ‘command and control’ applications and interactive information services [2]. For example, with a voice-dialling application the user can give a command like *call Peter* and the system dials the telephone number associated with the name. Interactive applications are distinguished from ‘command and control’ applications by the complexity of the task domain and the corresponding increase in the complexity of the language used. The system typically plays the role of a *co-operative* agent in a dialogue; for example, an agent for flight and train timetable information. Like a human service agent, the system has a responsibility to ensure that the dialogue proceeds smoothly. The system is responsible since (a) it may know more about the task domain than the user, and (b) it may be the source of problems in the dialogue, such as speech recognition and language interpretation errors. Consequently, the system should follow strategies for navigation — such as taking the initiative and asking questions to obtain information not provided by the user but which is necessary to complete the task — and strategies for detecting and repairing problems in the dialogue. In fact, co-operative interactive speech systems can be seen as instances of an indirect management interface [6], [8]: i.e. an interface where the user delegates a task to a computer-based agent which initiates (and monitors) actions in order to solve the task.

By contrast, conventional interfaces to virtual reality are based on direct manipulation which has three primary characteristics [11]:

1. Manipulation is carried out by physical actions.
2. The objects manipulated are persistent.
3. The actions are rapid and reversible, and their effect on objects is immediately visible.

In the case of virtual reality, users perceive objects in the virtual world by means of a 2D, 3D or stereoscopic display. As with Macintosh and Windows95 interfaces, actions are effected on objects by selecting commands from menus, keyboard sequences or mouse operations. For example, the user can navigate in the world by using cursor keys, select objects by clicking on them, and apply actions to the selected object by choosing an operation from a menu. The range of objects and actions available to the user is explicit, and the user has the responsibility for explicitly initiating and monitoring actions to ensure that the desired effect has been achieved.

Combining a speech interface with a direct manipulation interface results in a multimodal interface where users can act upon the world by issuing physical or speech commands and, conversely, the system can respond by speaking and/or by making changes in the virtual world [8]. Speech offers two obvious benefits when compared with a direct manipulation only interface.

The first benefit is that speech offers a way of issuing commands while allowing hands and eyes to remain free. Operations normally carried out through the direct manipulation modality — such as transportation, change of view, object creation and deletion, etc — can be effected without tying up another modality. Thus multiple actions can be simultaneously carried out using different modalities. This is particularly useful in cases when hands/eyes are already busy, but other tasks need to be dealt with from time to time; for example, when direct manipulation is used to drive a car, speech can be used to control the radio, car-phone, and other on-board systems. Alternatively, the user can combine their actions to achieve synergy effects by, for example, clicking on an object and simultaneously speaking the action to be performed on the object.

The second benefit is that users can refer to objects which are not present in their current view of the virtual world; in a direct manipulation interface, actions can only be applied to objects which are visually present. Users can use speech to select and manipulate objects which were in visual focus (the last town entered), will be in visual focus (the next town on the motorway), are simply known objects (my home town), abstract objects (such as the set of towns which I have driven through), high-level actions, and so on.

Of course, the most obvious benefit of speech is naturalness, or more precisely, familiarity. Users are familiar with using language to act in the world. However, just as virtual worlds do not necessarily obey the conventions of the physical world, so too the standard conventions of language use do not necessarily apply when interacting with machines. The benefit of speech needs to be tempered with the ‘unnaturalness’ of using a restricted language which the system can recognise and understand, as well as with the user’s familiarity in using direct manipulation to carry out the same task. For example, a user may become very familiar with, or simply prefer, using direct manipulation for self-transportation. Furthermore, using speech commands for self-transportation actions is not always natural in the everyday world: normally, we simply carry out the appropriate physical actions rather than say *legs, move me to the bar!*. However, there are situations where this action is ‘naturally’ carried out using language; for example, a physically handicapped person may rely on a helper to move their wheelchair. Extrapolating from the latter type of situation, users may find it more natural to use spoken commands for certain classes of action in the virtual world if they are addressing an agent specialised for these actions.

A multimodal interface combining speech and direct manipulation can provide more efficient interaction than a single modality interface, and give us the benefits of both modalities. It can also allow one modality to compensate for limitations of the other. For example, a direct manipulation interface can compensate for limitations of speech by making immediately visible the effects of actions upon objects, and indicating through the display which objects (and by extension which actions) are currently salient for the system. In addition, the user is free to decide which modality to use for their actions; for example, users may use direct manipulation for transportation within the virtual world, but the speech modality for manipulating objects. Although the motivation for the choice of modality is frequently inscrutable, various factors, apart from personal preference, are important including: the ‘naturalness’ of an action in a modality (issuing spoken commands to move oneself seems counter-intuitive); and the difficulty or complexity of carrying out the action in the other modality (such as the sequence of menu selection and mouse clicking required to manipulate an object). Finally, recent empirical studies have suggested that users prefer to interact multimodally and that this can reduce errors and task completion time compared with a single modality interface [10].

3. The TALKING AGENTS System

We are developing a generic framework for speech interaction in virtual environments. The central innovation is that by combining intelligent agent and spoken dialogue techniques, users talk directly to agents in the virtual world which carry out specialised functions. These techniques have been implemented in a prototype system which is populated with talking agents for transporting the user, fetching objects, painting objects, increasing the size of objects, and so on.

3.1 Virtual Reality System

DIVE (Distributed Interactive Virtual Environment) is a tool kit for building distributed VR applications in a heterogeneous network environment [5]. DIVE allows a number of users and applications to share a Virtual Environment where they can interact and communicate in real-time. This virtual environment is a database of entities: graphical objects (views), and hierarchically organised abstract objects (DIVE objects). The database is actively replicated among all sites participating in a DIVE world. Each replica is controlled by an Application Process that manages the movement and interrelationship between the objects component parts and responds to interrupts generated by changes in the objects environment.

3.2 Architecture

The system is built upon the DIVE system and adds components for speech processing and language understanding as shown in Figure 1. Input from a microphone is analysed by a speech recogniser which outputs a semantic template specifying an object and the action to be applied to it. Reference resolution identifies which object in the virtual world the user is referring to, and then executes the appropriate action. In addition to feedback via the graphical interface, the system also provides spoken feedback via a speech synthesiser.

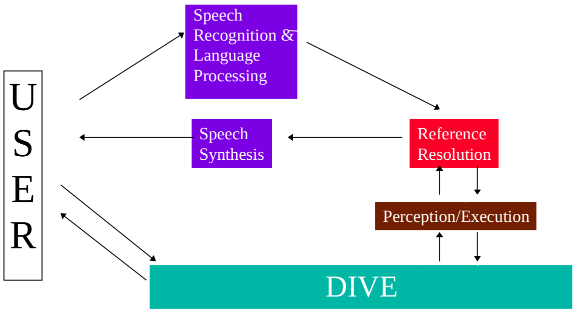

Figure 1: TALKING AGENTS Architecture

3.3 Speech Recognition

Techniques for speaker-independent, continuous speech recognition have now been developed to the point where their recognition accuracy makes them suitable for real world applications, albeit with restricted language [4], [9]. In particular, finite state grammar which only model the relevant acoustic information for pre-defined phrases, provide sufficiently high recognition performance for command and control applications which require restricted language.

A large recognition vocabulary can be defined using separate grammars appropriate to different stages of the interaction, or, in our case, for different agents in the virtual environment. While this may limit the linguistic capability of the system, it does offer the advantage that subsequent level of analysis can be restricted in scope: syntactic, semantic and pragmatic analysis need only handle phenomena which can be handled by the speech recogniser. Moreover, restricted language use is not only a natural consequence of talking to machines, but it is also a natural part of many everyday activities which are focused on a particular task. Command and control applications, as well as interactive dialogue applications where the user input at different stages of the dialogue can be accurately predicted, are very promising areas for high accuracy speech recognition using finite state grammars.

We use a commercial recognition system (Nuance) which employs finite-state grammars for recognition. Semantic representations are directly assigned in the grammar, so obviating the need for separate syntax and semantic components. Each agent has one or more grammar which specify the range of commands it can understand and execute. For example, a ‘painter’ agent’s initial grammar is:

```
.PAINTER      ([(move to OBJREF) {<command move>}
              (paint OBJREF ADJ:color
                 {<command paint> <color $color>})
              ([(what ?(can you do ?(for me))) help]
              {<command help>})])
```

which allows recognition of move and paints commands, as well as requests for help. At runtime, multiple grammars are loaded into the recogniser and, when a dialogue is initiated with an agent, the appropriate grammar is activated.

3.4 Agent Modelling Framework

For the task of reference resolution and building agents with interesting behaviour it is desirable to use a high level language suitable for complex symbolic computations. But languages such as Lisp, Prolog and Smalltalk do not support concurrency, reactivity and real-time control which are vital for concurrent reactive agents. However the new concurrent constraint programming paradigm in general, and Oz in particular support these requirements. Oz is designed to support multiple concurrent agents, which makes it well-suited for our purposes. It is based on a new computation model for higher order concurrent constraint programming (CCP) which provides a uniform foundation for functional programming, constraint and logic programming, and concurrent objects with multiple inheritance. We therefore choose Oz for the implementation of the framework and ODI, an existing interface between Oz and DIVE including an object layer for supporting agent abstractions [1].

The implementation mainly consist of a talking agent class which allows individual agents to inherit basic speech, dialogue and perception methods. Sub-classes of talking agents can refine these methods; for example, a sub-class of ‘secure’ agents might require that commands are confirmed and, if necessary, clarified.

ODI includes mechanisms for communicating between different DIVE applications. This makes creating a speech interface to any DIVE application a simple task of defining a ‘talking agent’ with an appropriate grammar and some methods to dispatch actions to the application.

3.5 Interaction Metaphor

A central issue in developing a speech interface to virtual worlds is the nature of the relationship between the system and user. In interactive dialogue systems, the role of the system is clear: it is a simulation of human agent for the information service, and the user can expect similar, albeit more limited, behaviour from the system. In direct manipulation interfaces to virtual reality the basic metaphor is **Personal Presence**: the user is embodied as an actor in the world and is thus provided with a perspective on the world. However, this metaphor is not so clearly applicable for spoken interaction since there is no obvious dialogue partner. Various metaphors for spoken interaction have been proposed:

Proxy The user can take control of various agents in the virtual world and thereby interacts with the virtual world through them; for example, *painter, paint the house red!*
Divinity The user acts like a god and controls the world directly; for example, *Let the house be red!*
Telekinesis Objects and agents in the virtual world can be dialogue partners in their own right; *house, paint yourself red!*
Interface Agent The user communicates with an agent, separate from the virtual world, which carries out their spoken commands

Selecting the appropriate interactional metaphor is very important for a speech interface since it will affect the language used in addressing the system: i.e. the complexity of user language is partially determined by what they think the communicative competence of the system is.

Using an *interface agent* to embody the competence of all agents in the virtual world can be problematic since there is no clear indication to users what commands can be understood, and speech recognition always needs to be able to process any command. Instead, we have adopted the *proxy* interaction metaphor which provides a close correlation between the functional and communicative abilities of agents. The function of the agent, partly indicated through its graphical form, subtly suggests to the user what commands are available (and the agent can explicitly specify these commands if requested); for example, it is clear that a 'painter' agent will understand commands concerned with painting objects, while a 'pump' agent (which increases the size of objects) will not. By exploiting this natural tendency for users to constrain themselves, the burden on speech recognition can be eased; only the grammar appropriate to the agent being addressed needs to be active.

3.5.1 Addressing Agents

In the current framework a dialogue is initiated by clicking on a talking agent[1]. No speech input is used until the dialogue is started. If the talking agent has not been spoken to for a while, it will greet the user. It will then determine a grammar to use and start to listen for commands. Normally each talking agent has just one grammar although in some cases they have more than one so that different command sets can be recognised depending on their state.

Using direct manipulation to initialise the dialogue is a fast and accurate way of activating a visually-present talking agent. However, another strategy is required for agents which are not visible to the user. At present we use a 'phone' talking agent which the user clicks on, and then describes the remote talking agent they want to talk to[2]. For example telling the phone to call a red pump will connect the phone to the pump. The dialogue is then conducted through the phone which adopts the grammar of the remote talking agent until the dialogue is completed. Figure 2 illustrates the situation where the user has used to the phone to call the pump to *pump up the cubes* and the pump agent is moving into view.

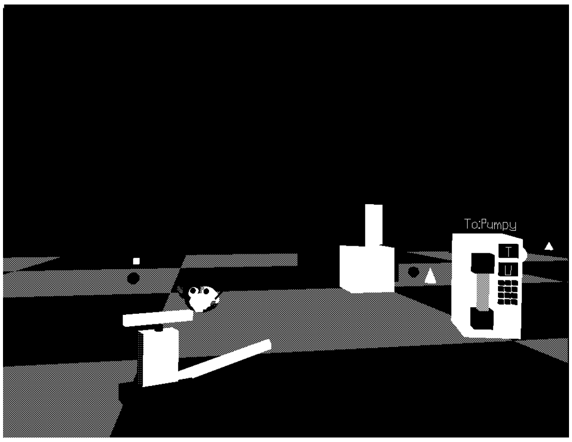

Figure 2: Controlling remote agents via a telephone agent

We are currently looking at methods for dealing with situations where more than one remote talking agent matches the user's description, and methods which allow agents to be summoned by the services they offer.

3.5.2 Feedback

An important part of a speech interface in a multimodal system is giving visual feedback to the user. It should for example be clear to the user if the talking agent is listening or not (if the user does not speak to agent for a pre-determined period of time, the agent stops listening). Either the talking agent implements its own methods for this or uses a 'talking face'. The 'talking face' appears on the side of the talking agent facing the user during the dialogue, as illustrated for the pump agent in Figure 2. It shows that it is listening by raising its ears and starts to flap them when being addressed by the user. It also nods and moves its mouth while speaking, this making clear which talking agent is speaking.

This is the default way of handling visual feedback; it is used for talking agents which are not in themselves natural dialogue partners, like a pump. But for talking agents

[1] This presupposes that user knows which agents are speech-enabled; otherwise, getting the attention of an agent can be a hit and miss affair. One approach, not yet implemented, is to give talking agents a small icon, or badge, indicating they are speech-enabled.

[2] The phone talking agent follows the user around the virtual world.

reassembling a person the ‘talking face’ should not be used, since that would give the talking agent two faces during a dialogue. The major problem with this method is that the visual feedback can be hard to discern when, for example, speaking to a talking agent far away.

3.6 Reference Resolution

The reference resolution component, in conjunction with the DIVE interface, is responsible for matching descriptions to some object in the DIVE environment which the user is referring to. It uses ontological information about objects in DIVE world, linguistic information such as definiteness, perceptual properties such as colour, and, most importantly, focus to resolve references.

3.6.1 Object Focus

Objects in the virtual world can be in focus to different degrees. What determines their focus level is a combination of different parameter from the visual and discourse situation, and how these parameters change over time. These parameters vary in their priority: an object which is being point at is more in focus than one just in the visual field. Both of these have priority over an object which has been mentioned in a user utterance. The parameters also persist/decay at different rates so that as the interaction progresses, their focal status also changes: objects which have been mentioned by user, or in successful actions, stay in focus longer than an object which has only been in visual focus.

3.6.2 Property Perception

Properties in descriptions are important for discriminating between objects. Of particular interest are non-discrete properties, such as colour, which can overlap and vary in how they are perceived and described by users. For example, the colour of an object in a graphical rendering of a virtual world may be described as red or brown due to overlap between the properties. A prototype approach is used for property matching: a property holds of an object if the semantic value is sufficiently close to the property prototype. In this way, a ‘best fit’ is established between the user description and properties of objects.

3.6.3 Discourse Modelling

Without a distinction between the discourse situation and world, reference resolution needs to be applied to all objects in the virtual environment. This is not efficient due to the potential size of the search space; for example, if the user issues the command *bring me the cube*, then the reference of *the cube* is resolved with respect to all objects in the world rather than those which are most salient. Without a discourse model, the system also lacks the ability to resolve references to previous actions.

The solution is to provide a discourse model which distinguishes between a participant’s understanding of the world, and the world itself. In our current system, all talking agents share a processed model of the world and the reference resolution methods described above. The model contains only vital perception knowledge and is structured for maximum performance; objects are ordered according to their saliency for the user at the time of speaking. While there is a potential consistency problem with this method, the real-time performance gain of not processing the raw world database for each command seems more important.

3.7 Robust Interaction

It is tempting to adopt an ' errors don' t matter' approach to interaction: since the user can see the changes the agent made to the virtual world, errors in speech understanding or reference resolution can be corrected by the users themselves. This principle, however, is problematic. Users may have difficulty in identifying the source of the problem — it may arise from a system error in recognition or reference resolution — or they may have difficulty in reversing the effects of an incorrect action. This may make the task of repairing errors inconvenient for the user. Additionally, in safety-critical applications, errors are more than simply inconvenient: the user must be able to trust an agent to execute the command authorised by the user.

The solution to this lies in augmenting the system with capabilities similar to interactive dialogue systems; namely, taking the initiative to confirm commands before they are executed, or clarifying incomplete or ambiguous commands[3]. These strategies can be configured for particular agents: not all talking agents will require that commands are confirmed by the user, or that clarification of incomplete commands is sought. If clarification is allowed, the agent can clarify the command with the user as shown below:

User: paint …
Painter: what do you want to paint?
User: this house
Painter: what colour do you want to paint the house?
User: red
Painter: painting the house red ...

In this way, the user is able to trust talking agents to only execute the appropriate actions.

3.8 Future Work

While the TALKING AGENTS system allows speech to efficiently complement direct manipulation in multimodal interaction for virtual environments, there still remain many issues which we need to resolve. Here we briefly mention three issues.

[3] The agents can also take the initiative in other circumstances. For example, after the user has asked the pump agent to increase the size of an object, the agent replies by asking the user to say *stop* when it has reached the required size.

Firstly, users are not yet able to exploit the power of a discourse model which records their actions; i.e. they cannot ask a talking agent to apply a previous action to another object, or undo the effects of an action. In order to redo or undo actions, agents must have the capability to reason about the state of the discourse model. We are working on a framework for Agent Oriented Programming where an agent is an entity whose state is viewed as consisting of mental components such as beliefs, capabilities, choices and commitments. These have the required knowledge to be able to reason about past actions and we have planned some experiments on this.

Secondly, the functionality of talking agents is not yet integrated with other capabilities of the DIVE system. One simple extension would be to activate talking agents through the aura mechanism [3]: i.e. when a user's aura intersects with a talking agent's aura, this enables the talking agent to initiates a dialogue[4]. A more complex extension is integration of talking agents with the DIVE mechanism for audio communication between distributed users. This would allow users to interact with human and computer agents in the virtual world in an analogous manner.

Finally, we are looking to test and evaluate this system with users in a realistic scenario. One possibility is a virtual travel agency; spoken dialogue systems have already been used as agents for information services such as flight information and reservation. Users can configure a trip to suit their personal needs by providing parameters to a travel talking agent and the trip is then visualized in a virtual environment. The travel talking agent can also guide the users around the locations, and answer specific questions in cases where it is inappropriate to provide the background information graphically. This type of application scenario also has the benefit of allowing us to investigate how users react to realistic levels of speech recognition error, and which tasks they find speech a more suitable modality than direct manipulation.

4. Conclusions

We have described how a direct manipulation interface to virtual worlds can be augmented with a speech interface. In order to achieve this, we have addressed the issues of constraining speech recognition so as to achieve high accuracy, understanding user language in the context of human-computer interaction, and developing an appropriate interaction metaphor. In the TALKING AGENTS system we use speaker-independent recognition and dialogue partners which are part of the virtual world itself. The dialogue partners are modelled as agents which provide specialised functions in the virtual world, and the communicative ability of the system is dynamically correlated with the agent the user is interacting with. This approach provides a generic platform for adding simple spoken dialogue capability to virtual reality applications.

[4] This will be problematic if the user's aura intersects the auras of multiple agents, thereby initiating multiple dialogues simultaneously.

References

[1] Axling, T., Fahlen, L., and Haridi, S., "Virtual Reality Programming in Oz" in *Proceedings of the 3^rd^ Eurographics Workshop on Virtual Environments*, 1996.

[2] Eckert, W. and McGlashan, S., "Managing spoken dialogues for information services," in *Proceedings of 3rd European Conference on Speech Communication and Technology,* 1993, pp. 1653-6.

[3] Fahlen, L.E., Brown, C.G., Stahl, O. and Carlsson, C., "A Space Based Model for User Interaction in Shared Synthetic Environments" in *Proceedings of SIGCHI'93*, 1993.

[4] Giachin, E. and McGlashan, S., "Spoken Language Dialogue Systems," in *Corpus-Based Methods in Language and Speech Processing*, Young, S., and Bloothooft, G., Eds. Kluwer, 1996.

[5] Hagsand, O., "DIVE - A Platform for Multi-User Virtual Environments" in *IEEE Multimedia*, Spring 1996.

[6] Kay, A., "User Interface: A personal view," in *The Art of Human-Computer Interface Design*, Laurel, B., Ed. Reading, Mass: Addison-Wesley, 1990.

[7] McGlashan, S., "Speech Interfaces to Virtual Reality" in *Proceedings of 2^nd^ International Workshop on Military Applications of Synthetic Environments and Virtual Reality*, 1995.

[8] McGlashan, S., "Towards multimodal Dialogue Management" in *Proceedings of Twente Workshop on Language Technology 11*, 1996.

[9] Meisel, W.S., "ARPA Workshop reports on speech-recognition state-of-the-art," in *Speech Recognition Update*. ISSN 1070-857X, 1995.

[10] Oviatt, S., "Multimodal Interfaces for Dynamic Interactive Maps" in *Proceedings of CHI'96*, 1996.

[11] Scheneiderman, B., "Direct Manipulation: A step beyond programming languages," *IEEE Computer*, vol. 16, no. 8, 57–69, 1988.