Some Notes on Agents in the Twente Virtual Theatre Environment

Anton Nijholt and Hendri Hondorp*

Centre of Telematics and Information Technology
University of Twente
PO Box 217, 7500 AE Enschede, the Netherlands
(anijholt|hendri)@cs.utwente.nl

1 Introduction

We present research on visualisation and interaction in a realistic model of an existing theatre. Our virtual theatre has been built according to design drawings of the architects of an existing building. Visitors can enter the theatre and walk around. Information about today's performances is available on a board that is automatically updated. Visitors may also go to the information desk, see previews and start a (natural language) dialogue with an agent called 'Karen' who has access to the theatre database with information about performances.

A first version of our environment has been available on WWW since early 1998. Another version has been installed in a regional technology activity centre. The information that is provided by the system is up to date, many people use it to get informed about performances and we think that at this moment it is the only virtual environment in the world that is really used by the general public. This success has made it possible to start new research. In this paper the emphasis is on research results that are now being incorporated in a next version of the environment, again accessible for the general audience, and new research which has to provide results that will be integrated in the environment in its next update.

2 Agents and Interactions

2.1 A Navigation Agent

Navigation in virtual worlds is a well known problem. Usually, navigation input is done with keyboard and mouse. This input allows the user to move and to rotate, to jump from one location to an other, to interact with objects and to trigger them. We developed a navigation agent that helps the user to explore the environment and to interact with objects in this environment by means of speech commands. Obviously, we do not want completely separated modalities. It

* Acknowledgements: Many people are involved in the research reported here and have contributed to this paper. I would like to mention Joris Hulstijn, Danny Lie and Boris van Schooten.

should be left to the user to choose between the interacting means or to use both, sequentially or simultaneously. A smooth integration of the pointing devices and speech in a virtual environment requires has to resolve deictic references that occur in the interaction. Moreover, the navigation agent should be able to reason (in a modest way) about the geometry of the world in which it moves.

Straightforward typed commands or similar speech commands make it possible for the user to explore the virtual environment. Navigation also requires that names have to be associated with the different parts of the building, the objects and the agents. Users may use different words to designate them, including references that have to be resolved during reasoning. Our present agent uses keyword spotting. A next version of the system will allow a finite state specification of the user's input. We slowly extend and improve the interaction and navigation intelligence of our present navigation agent. At this moment we are exploring the possibility of speech recognition for several clients on a central server and the advantages of making the navigation agent visible for the user.

Our agent is able to understand command-like speech or keyboard input. It tries to recognise the name of a location in the visitor's utterance. When the recognition is successful, the agent guides the visitor to this location. When the visitor's utterance is about performances the navigation agent makes an attempt to contact Karen, the information and transaction agent. Our next implementation of the navigation agent assumes that the navigation agent knows about (or should be able to compute) the current position and focus of gaze of the user, the objects that are in the eyesight of the visitor, properties objects have, geometric relations between objects and locations, possible walks towards objects and locations, a history of previously visited locations or routes, actions the agent is performing (or has performed) and the previous communication with the visitor. This knowledge should make it possible that the navigation agent is able to interpret a majority of user utterances correctly and act accordingly.

2.2 The Information and Transaction Agent

As mentioned before, a second agent called Karen allows a natural language dialogue with the system about performances, artists, dates, prices, etc. Karen wants to sell tickets. She is fed from a database that contains information about performances in some of our local theatres. Developing intelligence for Karen, in this particular environment, is a main aim of our project. Presently the input to Karen is keyboard-driven natural language and the output in our for the general audience WWW accessible virtual world is screen and menu based. In a prototype system we allow Karen to use a combination of speech synthesis and information presentation on the screen. Based on the user utterance, the context and the database, the system has to decide on a response action, consisting of database manipulation and dialogue acts.

As mentioned earlier, in our experimental system Karen's spoken dialogue contribution is presented by visual speech, that is, a 'talking face' on the screen, embedded in the virtual world, mouths the questions and part of the responses. If necessary, information is given in a window on the screen and the user can click items to get more information. The virtual face that has been designed allows animation of lip and face movements and animation of some simple face expressions. For speech-image synchronisation 3D images of visemes are called when corresponding phonemes are spoken. Since our application is web-based, it requires the solution of technical problems dealing with sending and compressing sound files, commands and synchronising sounds and animations.

2.3 Language Skills of the Agents

At this moment our agents have different language skills. On the one hand we have Karen and a grammar specification of the input for Karen based on a corpus of WoZ obtained keyboard-based dialogue utterances. On the other hand we have a navigation agent with language skills that are based on the current limitations of speech input. We see two, hopefully, converging developments. On the one hand, speech recognition moves from keyword spotting, to finite state utterance specification to (word-graph) based context-free language specification. On the other hand, we hope to see the development of corpus-based grammar inference methods that allow the design of an interface between domain-dependent speech recognition, syntactic analysis and interpretation of user dialogue utterances.

We have induced for our navigation agent a probabilistic grammar from a corpus of user utterances that have been obtained from scenarios presented to (potential) visitors. This allows the design of a primitive system and it allows bootstrapping this system from the original corpus and from corpora obtained from logging the interactions between visitors and the navigation agent.

2.4 Future Agents

Our approach to designing a virtual environment for an interest community is bottom-up. We started with two agents and in order to introduce others a Java implemented agent platform has been designed. Using this platform other agents have been introduced, e.g., an information board showing upcoming events. Users can ask questions to the board and get answers. The board passes the questions to other agents, in this case questions about performances are passed to Karen. One issue that is receiving much attention currently is the non-verbal behaviour of the agents.

3 Software Engineering Issues

3.1 Use Cases

In our system we have to deal with different agents and with different dialogues or interactions between agents and users (visitors of our theatre). We are investigating the use of object-oriented and agent-oriented design techniques to rebuild our theatre.

One design approach which we found useful is the description of use cases. A use case (Cockburn [1]) is a detailed account of the context and goals of the different actors involved in a transaction, together with a possible sequence of actions. They are recommended in object-oriented development to find common data elements to be modelled as objects. Use cases become part of the requirement specification document that serves as a contract between the system developer and the customer, in this case the service provider. Because they can be intuitively understood, use cases facilitate communication with user focus groups and with the customer. Based on several scenarios for the navigation and the information & transaction agent use cases have been developed (Hulstijn [2]).

3.2 Formal Methods

Both from an ergonomical and a software-engineering viewpoint, the design of interaction in virtual environments is complex. Virtual environments feature a variety of interactive objects, agents which may use natural language to communicate, and multiple simultaneous users. All may operate in parallel, and may interact with each other concurrently. Next to this, the possibility of using Virtual Reality techniques to enhance the experience of virtual worlds offers new ways of interaction, such as 3D navigation and visualisation, sound effects, and speech input and output, possibly used so as to complement each other.

We attempt to address both of these issues by means of a formal modelling technique based on the process algebra CSP. For that reason, in our virtual theatre a simplified flow of interaction has been specified, showing all relevant interaction options for any given point in time. The architecture has been modelled in an agent-oriented way, representing all system- and user-controlled objects, and even the users themselves, as parallel processes. The interaction between processes is modelled by signals passing through specific channels. Interaction modalities (video versus audio, text versus graphics, etc.) may also be modelled as separate channels.

This modelling technique has some strong points. Firstly, a simplified and formal model enables a clear and unambiguous specification of system architecture and dynamics. Secondly, it may be useful as a conceptual model, modelling the fact that a user experiences interaction with other users and agents in a similar way than in a completed system, and explicitly showing which options are available when and through which modalities. Thirdly, it enables automatic prototyping, such as architecture visualisation and verification of some system properties (Schooten [3]).

References

1. Cockburn, A.: Using goal-based use cases - transitioning from theory to practice. *J. of object-oriented programming*, 1997.
2. Hulstijn, J.: Modelling usability: Development methods for dialogue systems. In: Alexandersson, J. et al. (eds.) IJCAI-99 Workshop on *Knowledge and Reasoning in Practical Dialogue Systems*, Sweden.
3. Schooten, B. van et al: Modelling Interaction in Virtual Environments using Process-algebra. In: *Interactions in Virtual Worlds*, May 1999, University Twente, 195–212.