An Interactive Speech Interface for Virtual Characters in Dynamic Environments

Q. Mehdi, X. Zeng, and N. Gough
School of Computing and Information Technology
University of Wolverhampton
35/49 Lichfield Street
Wolverhampton WV1 1EQ, UK
Q.H.Mehdi@wlv.ac.uk

Abstract

In this paper, we propose a new improvement to our 3D Virtual Story Environment System (3DVSE) by adding a real-time animation with voice synthesis. The new system offers a flexible and easy way to generate an interactive 3D virtual Environment (3DVE) as compared to traditional 3D packages. It enables the user to control and interact with the virtual characters via speech instructions so that the characters can respond to the commands in real time. This system has the potential, if combined with artificial intelligence, to act as a dialogue interface for believable agents that have many applications such as computer games, and intelligent multimedia applications. In this system, the agent can talk and listen to fellow agents and human users.

Keywords:

3D Virtual Story Environment System (3DVSE), Virtual Character, Natural Language Processing, Speech synthesis, Dialogue manager.

1. Introduction

The development and research on computer natural language understanding (NLU), has become an interesting subject for many researchers over the last few decades. This has been further complimented by the advancement in speech recognition and natural language processing (NLP) technologies; and the significant improvement of personal computer processing power and graphic technologies (Larson 2002). The integration of natural language descriptions and visual images has become a new research area. The work has shown how physically based semantics of concrete nouns, depictive adjectives, locative prepositions and motion verbs can be seen as conveying objects, attributes or spatial, kinematics and movement (Zeng *et al* 2002). This allows for the construction of a system that has the function to create graphical simulation of scenes or events described by natural language. Speech-enabled systems have recently become a common feature in desktop, telephony, and web applications (Abbott 2002). They have also found their use in the field of Virtual Environments, where speech-enabled applications have been implemented in improving VE tool's user-interfaces (Godereaux *et al* 1999; Luin 2001). The virtual character is one of the important components in virtual environment as most of the interactions lie between the user and the computer-controlled character. In order to enable a virtual character to interact with humans via language, the character should have the capability of understanding humans through speech recognition, natural language understanding, and communication via natural language generation and speech synthesis (Jurafsky and Martin 2000).

The work described here is part of ongoing research in (3DVSE) 3D Virtual Story Environment system (Zeng *et al.* 2002; Mehdi *et al*. 2003). Story-based natural language was used as the premier input source in this system to construct a 3DVE engine. The NLP and 3D graphic presentation techniques were combined to allow construction and manipulation of VRML based scene graph in real time. This offers a flexible and easy way to generate an interactive 3DVE as compared with traditional 3D packages. In this work, we intend to improve the system by adding an additional component to the existing system, which enables the user to control and interact with the virtual character via speech instructions. The user would be able to instruct the character via speech input enabling the character to respond to the commands in real time i.e. the character can be manipulated (e.g. move around or get object, etc) to accomplish the goal. This can be further enhanced by adding a dialogue module that enables the character to request relevant information related to the previous instruction in order to complete the final goal. Therefore, when the character encounters incomplete or unclear instructions, it will have the ability to obtain further information from the user by asking questions related to the task in hand. In this work, we propose a method for creating an interactive virtual character in a dynamic virtual environment through speech instructions. The new system will be developed by using a combination of virtual reality techniques along with the integration of speech-enabled interaction, and the use of object-orientated character animation technology. Examples will be given to show how the user interfaces with the virtual character.

2. Related Work

The generating and manipulating of virtual objects in VE based on natural language input has been investigated by many researchers over the last few decades. One of the earliest systems, SHRDLU was a natural language understanding system that allowed user interaction using English terms (Winograd 1972). The program operated in a

very restricted virtual world of blocks, in which, the user instructs SHRDLU to move various objects in particular ways around the positions of the blocks, such as “Pick up a big red block” and “Grasp the pyramid”, etc. SHRDLU's world was so simple that the entire set of objects and locations could be described by including as few as 50 words. The system is capable of answering questions, executing commands, and accepting information in an interactive English dialogue. Clay and Wilhelms (1996) developed a language-based system called Put that focuses on spatial relationships to reconstruct 3D models of the world by natural language, and output the corresponding image on the graphic display. This system is limited to an artificial subset of English consisting of expressions of the form Put (object X + Preposition+ Object Y). They used just a few simple spatial relationships; such as in, on, and at, parameterised by a limited number of environmental variables that can provide easy object manipulation. In these systems, several simple ideas have been combined to make the simulation of "understanding" far more convincing even without complicated graphic components and virtual characters. There has been some research to apply NL in character animation and behaviour, such as the implementation of a command interpreter based on NLP techniques to control or guide animated characters to achieve a specified task in VE. Among those researchers are Bindiganavale *et al* (2000) and Badler *et al* (2000) who have described 3-D animated agents that understand natural language and can perform some actions in a virtual space. The agent was given language-based instructions from which it extracted parameters for its actions. The parameters contain information such as linguistic information, spatio-temporal information, and manner information that were often expressed as adverbs, or as applicability and terminating conditions. Furthermore, a user should be able to dynamically refine the avatar’s behaviour during a real time simulation without having to undertake a lengthy off-line programming session. Tanaka *et al* (2001) developed a lifelike animated agent system named Kairai to carry out preliminary research on the next generation NLU system. Kairai understands what we say in natural language, especially the words such as “left”, “right”, “in front of” and “behind” that indicate relative location in a virtual space. Typical actions performed by the (visible) software robots are “Push”, “Go”, and “Turn.” However, the robots interpretation of “left” and “right” is determined by the consideration of both human and robot software orientation issued from the command. There are several conversation systems, which use language interface to allow the user to interact with visual characters and enable easy navigation and hence help users to complete their tasks faster in a VE. For example, Godereaux *et al* (1999) developed a system called Ulysse, which was an implementation of a linguistic device in a virtual world. It features a reference resolver to associate noun phrases with entities of the virtual world and a geometric reasoner to cope with prepositions, groups, and spatial descriptions, to enable a limited understanding of the structure of the virtual world. When receiving navigation commands from the user, Ulysse analyses the word stream and navigates the user in the virtual world as if it is truly represents the user’s viewpoint. Luin *et al* (2001) have designed a natural language based system for agent navigation in a virtual reality (VR) environment. The agent is part of an agent framework that can communicate with other agents within the framework, can guide visitors in the environment and has the ability to answer questions about the environment (a theatre building). The navigation agent can help visitors in exploring the environment, and allow them to ask questions where advice can be given.

3. An Overview of Speech Interface for Virtual Environments

The research and development of computer speech recognition and synthesis have grown across many fields of study, including linguistics, computer science, mathematics, statistics, and psychology (Abbott 2002). Speech technology was once limited only to the realm of science fiction, but now it is available for use in real applications in computing (Java Speech API Programmer’s Guide, 1998). Speech interfaces are increasingly being used in ‘command and control’ applications and interactive information services (Eckert *et al* 1993). In an interactive character based virtual world, spoken enabled interface has many obvious advantages in comparison to the traditional menu-based interface. It is an easy medium that can be used efficiently to formulate run-time instructions for virtual human characters (Bindiganavale *et al* 2000). It has an advantage that makes the virtual human interface much more similar to real-life interpersonal communication. It can simplify and improve the response of real-time applications involving navigation or commands.
However, in order to develop a speech NL enabled interface for controlling a character based virtual world, the technology of understanding spoken language and some design issues need to be addressed. As a conversation in real world, one can hear and identify sound.

Speech Recognition is fundamental to the speech interface, which enables the computer to respond to the input speech by converting the spoken words into text or a similar form. Speech recognition can be divided into several steps (Larson 2002). A *recognition grammar* specifies and defines the speech input and its pattern to the speech recognizer. The incoming audio frequencies are analyzed and compared to the language phonemes in *phoneme identification*. The output of the sequences of phonemes are then compared to the words and patterns of words defined by the recognition grammar. The final step of processing is mapping sequences of phonemes to text this is called *word identification*. The results are generated to provide the application information about the words the recognizer has detected in the incoming audio. The output produced is the best guess (result) the recognizer deciphers from the user’s input, or several alternative guesses can be supplied as the result.

There are two different types of speech recognition technologies; namely speaker-dependent and speaker-independent speech recognition. Speaker-dependent recognition requires that each user goes through a process of

training the computer to recognize an individual's voice before using it. The successful commercial systems include IBM ViaVoice and Dragon NaturallySpeaking, *etc*. On the other hand, speaker-independent recognition does not require the user to train the system before use but developers must train the system with a collection of speakers. It strives to recognize what a person is saying without knowing anything about them.

Speech Synthesis is the process of transforming text into audio sounds, which simulate to human-like speech. Speech synthesis also consists of several steps. *Structure analysis* infers the structure of the input text, such as paragraphs, sentences and punctuation, etc. that are used in this stage. *Text normalization* uses data and rules from a specific database to find special constructs like abbreviations, acronyms, dates, and other forms that are usually specific for different situations. *Text-to-phoneme conversion* looks up words in a lexicon and converts each word to phonemes. *Prosody analysis* processes the sentences into a nature of speech that includes intonation, rhythm, pauses, emphasis, etc. and helps the user to easily understand the verbal message. After obtaining all the information in previous stage, the last stage of speech synthesis is *waveform production*. There are two general techniques being used, such as parameter-based synthesis (artificial and robotic sound) and concatenate synthesis (pre-recorded human speech). Because languages have so many structural differences, different phonemes and so on, the speech synthesizer must be configured to synthesize one specific language at a time.

Language Understanding extracts meaning from a text string by using predefined grammar and representing the semantics. It is composed of several steps. Once sentence output from speech recognition component, the sentence *tagging* marks individual words of the utterances from lexicon database and it gives a significant amount of information about the word and its neighbors. A syntactic parser takes tokenized text and converts it into parse trees by using predefined grammars or patterns. It involves parsing the sentence to extract whatever information the word order contains. Then a semantic parser interprets input sentences or syntactic parse tree into semantic representations such as frames or logical formulas and finally presents the meaning of the utterances.

In order to enable the user to interact with a virtual character, it is important to integrate a dialogue system to converse with users. For example, once the user's instruction have been analyzed and outputted, a semantic representation specifies the character and the action to be taken. The character then executes the appropriate action after identifying which object or direction in the virtual world the user is referring to. However, the character may not take any action when problems arise, *i.e*. incomplete or ambiguous instructions, problems with identifying the objects in a virtual environment, or improper actions. In this stage, the solution to this lies in generating an interactive dialogue system, which gives feedback to the user about any aroused problems. This will allow the user to continue providing more detailed and complete information so that any errors can be addressed. Currently, many new speech applications are using application-driven conversational dialogues (Larson 2002). This type of dialog forces the user to answer a specific question and enables the speech recognition system to recognize a relatively small number of words and phrases rather than all of the words in a large lexicon.

4. Proposed System Architecture

The methodology described here is a continuation of our ongoing research in 3DVSE (3D Virtual Story Environment) systems (Zeng *et al* 2002; Mehdi *et al* 2003). The system has employed story-based natural language as the primary input source used to generate 3DVE by integrating NLP and 3D graphic presentation techniques to construct and manipulate VRML (Virtual Reality Mark-up Language) based scene graphics in real time. In this work, the system has been improved by adding an interactive speech interface that would help the virtual character to achieve its objectives. The system now has the capability to accommodate speech enabled interaction and the use of character animation technology. This allows the user to provide the character with detailed information through the speech feedback loop, which will enable it to carry out its objectives as has been demonstrated by other speech enabled systems (Godereaux *et al* 1999; Luin *et al* 2001; Tanaka *et al* 2001). In this system, we use the Java Speech API as a standard interface; it functions as a middle layer between the underlying speech recognizer or synthesizer and the actual speech application. IBM's ViaVoice is used for speech recognition as its speech synthesis can easily interface with Java programming. This allows incorporating ViaVoice speech technology into the user interfaces. The speech for Java is built on top of the native speech recognition and synthesis of IBM ViaVoice in the same way that Java implementations are built on top of the native operating system GUI capabilities.

We have developed case grammar similar to that of Mast *et al* (1994) and Clay and Wilhems (1996) in order to represent each clause into one or more predicate. This was thought to be a suitable approach as it was unnecessary to use long and complicated grammar for a simple specified task. For example, a simple verb is directly mapped to one verb type and corresponds to the predefined grammar. Each predicate consists of a detailed expression:

<verb type, (what) object type, preposition type, object or direction>

The architecture of the system adopted in this work is illustrated in Figure 1. The system comprises of several modules that are needed to process the speech input and display the appropriate actions taken by the virtual character. When a user inputs the construction through a speech device, it is first translated into text via the speech recognition module. The sentence is then tagged and operated by grammar rules, and output is interpreted into a semantic representation to build a logical frame. This

presentation can be further improved by the use of an inference engine which implements the implicit constrains and updates the semantic representation, such as temporal and spatial reasoning.

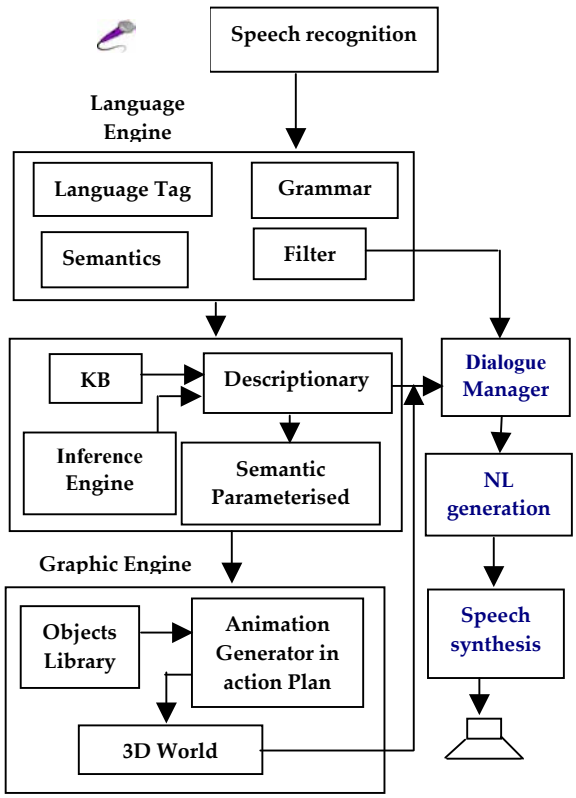

Figure 1. System Architecture

The semantic representation is converted into a set of parameterised data by a Descriptionary (a dictionary that uses XML based word frames to parameterise visual or describable words), and the verb then is parameterised by a character animation generator (a set of animation database for according diverse verbs). This utilizes behavioral rules in order to carry out the user's commands and presents them in a virtual environment. For example, *Move the red cup on the desk,* the sentence is parsed and matched to the predefined grammars and is then interpreted into the semantic representation:

1. Object1: Cup 2. Attribute: Colour: Red
3. Action: Move 4. Position: on 5. Object2: desk

The character should identify the object1 referred to as "red cup" and find object2, which matches the reference word in utterance to the objects and locations in the virtual environment. The corresponding actions should then be called and the instructions completed. This represents a complete and correct command. Whilst this is the desired output, three problems may arise when the user inputs the command. Therefore, it is important to build a dialogue module that allows the character to interact with the user and address the problems as well as trying to find out what information is missing and how they can get obtain it. The possible problems that might occur when processing the commands are as follows:

- **Incomplete sentence:** For example, *Move object X.* This sentence does not match the rule as it does not provide information as to where the user wants the character to move the object. The sentence is then marked as an incomplete in the semantic presentation stage and is sent to the Dialogue Management Module to generate a question that will request detailed information from the user, such as *Where do you want to Move object X?*
- **Wrong Expression:** If the user uses the wrong verb for manipulating objects. For example, *Pick the bed on the table*. A question must then be formulated to ask for correct information before the sentence passes through the knowledge base, which means that after checking the information with the Descriptionary, the action verb 'pick is wrong expression.

- **Complete sentence but wrong object:** For example, the user asks *Move object Y on the floor*. If there is no object Y in the virtual environment, the feedback from the environment is then sent to the Dialogue Management Module which then creates a response informing the user of the object's status, for example a message like, *Sorry, there is no object Y*, will be created.

- **Complete sentence but character requires help:** In this instance, the character understands the assigned task but requested further help from the user in order to cater for any new circumstances that may arise during carrying out the given task. This includes the introduction of a new character that may help to over come the problem.

The user would respond to these situations by either sending a new command/virtual character or provide the character with the missing information/instruction. Once the command is complete, the character will resume its task by accessing the database at its disposal. It has a choice of two possible databases, one containing information about physical objects present in the environment; the other is a general knowledge base, containing knowledge about the language understanding and meaning representation. This system therefore has the ability to control the virtual character and its behaviour by allowing the user to interface with the character and rectify any undesired behaviour that may have been generated by passing incorrect information to the character.

5. Object-Oriented Character Animation

The importance of interaction between an object and a virtual character is evident in most applications of computer animation and simulation (Kallmann *et al* 1998). The most challenging task in the creation of virtual characters and their environment is the interaction of the character with other characters and objects. Also, it is important to make the virtual characters understand the instructions given by the user and perform believable actions (Suliman 2001). This requires the character to access the necessary instructions before carrying out its task. Therefore, two databases have

been developed and made available to the character so that it can gain the required knowledge. The first is a general knowledge base, which contains knowledge about the language understanding and meaning representation. The second database contains information about physical objects that are present in the virtual environment, such as object, location, *etc*. an object-oriented character animation, that has interaction information such as intrinsic object properties, object behaviours, information on how a character should be animated while using an object would be required for this type of work. Many researchers have already proposed similar approaches, such as *SodaJack* (Geib *et al* 1994; Levison 1996) who developed objec*t specific reasoning* and Kallmann *et al* (1998) who created *smart object*. However, in this work, we have extended the idea by integrating a natural language interface and agent perceptions to identify which objects need to interact to implement the users' instructions. The information held by the object has been deposited in the Descriptionary as shown in Figure 1. These include the intrinsic object properties (*i.e*. material, volume, weight, *etc*.); and the action that is required by the character to interact with the object based on the properties the object has. Furthermore, an animation database has been developed for this purpose that contains some animation actions, which correspond to several predefined verbs that allow virtual characters to manipulate objects in a virtual environment, as illustrated in Figure 2.

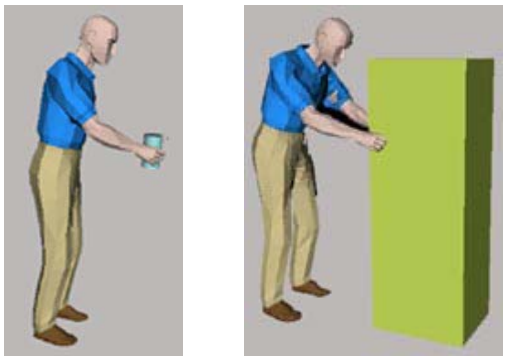

Figure 2. Two actions performed by a virtual character

6. Simulation Examples

Two examples will be used to demonstrate how the virtual characters interact with objects in the virtual world through the NL input in real time. In the first example, the user issues a command to the virtual character, for example, *move the cup.* The utterance is first converted into a word sequence through speech recognition module. The output is then analysed by the language engine and compared to the predefined grammar. Should the sentence be found as incomplete, as it is in this case, the speech synthesis will be automatically triggered and will ask the user "*Where do you want to put the cup?"* The user would then respond with the necessary instruction, for example, "*On the chair"*. The new sentence is then processed as a complete sentence in the grammar pattern <verb (move), (box) object type, (on) preposition type, (chair) object> and would be treated as correct utterance and is then processed by the semantic representation. The output of the semantic logic frame is then sent to the knowledge-based module, and then to the Descriptionary module that takes the output to the semantic parameterised and finally taken to the Graphic engine. The properties of the cup (such as size, weight, etc) indicate to the character, what actions it must carry out during interaction with the object. Now that the information is complete, the character has enough knowledge to identify where the cup and chair are located in the virtual environment. However, As the chair was not found in the environment, the character then signals to the dialogue management module to generate a message to the user, for example, "*Sorry, there is no chair in the environment"*. The sentence is then reprocessed through the system once the user has amended the information by issuing the correct command, for example, "*On the table"* Given the new information, the character is now in a position to complete the task, as shown in the screenshot in Figure 3.

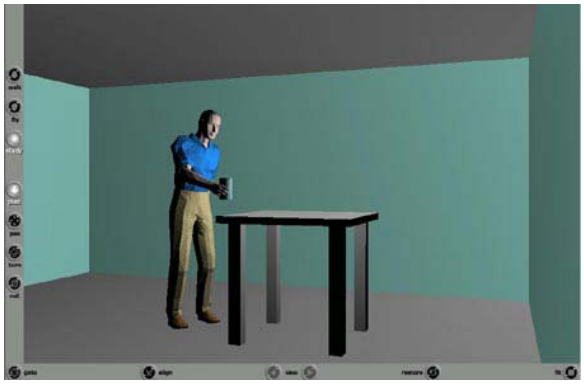

Figure 3. 3D scene generated

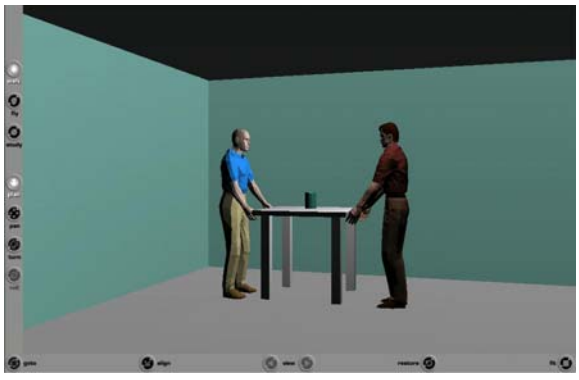

Figure 4. 3D Scene of a multi-agent corporation

In the second scenario, the user instructs the agent to "push the table into the top left hand corner of the environment". Having received the instruction, the character is found unable to complete the task on its own due to the weight of the table. Therefore, the character requests for help from the user. The user then sends another virtual character to help with the task. The new character has to navigate its way to the environment in order to reach the table. The two characters must now cooperate to achieve the given task as shown screenshot in Figure 4. This is a typical example of a

multi-agent cooperation system in which characters show believable social behaviours.

7. Conclusions and Future Work

In this work, we developed a new 3D Virtual Story Environment System by using a combination of virtual reality techniques along with the integration of speech-enabled interaction, and the use of object-orientated character animation technology. The natural language dialogue interface of this system represents an interactive interface between the user and characters in virtual storytelling environment. Many multimedia applications, computer games, believable agents and Multi-agent systems benefit from this proposed system.

The system is still in its early stage of development and more time is needed before a comprehensive conclusion can be reached. It can be further improved by incorporating a robust speech interface, artificial intelligence, real-time animation with voice synthesis and path planning to add to the features of the modelling of the object-character interaction that may be used in more complex tasks.

8. References

Abbott, K. (2002) *Voice Enabling Web Applications: VoiceXML and Beyond*. Springer-Verlag, Heidelberg, German.

Badler, N. *et al* (2000) Parameterized action representation and natural language instructions for dynamic behaviour modification of embodied agents. *AAAI Spring Sym.*, Stanford University in Palo Alto, California.

Bindiganavale, R., Schuler, W., Allbeck, J., Badler, N., Joshi, A and Palmer, M (2000) Dynamically altering agent behaviours using natural language instructions. In *Autonomous Agents*, Barcelona, Catalonia, Spain.

Cassell, J., Vilhjálmsson, H, H. and Bickmore, T. (2001) BEAT: the behaviour expression animation toolkit. *Proc 28th SIGGRAPH Annual Conf Computer Graphics & Int. Techniques*, Los Angeles, California.

Clay, R and Wilhelms, J. (1996) Put: language-based interactive manipulation of objects. *IEEE Computer Graphics and Applications,* pp. 31–39, March 1996.

Eckert, W. and McGlashan, S.(1993) Managing spoken dialogues for information services. *Proceedings of 3rd European Conference on Speech Communication and Technology*, Berlin, Germany.

Geib, C., Levison, L and Moore, M. (1994) *SodaJack: An Architecture for Agents that Search for and Manipulate Objects.* Technical Report MS-CIS-94-13*,* University of Pennsylvania.

Godereaux, C., EI-Guedj, P., Revolta, F and Nugues, P. (1999) Ulysse: An interactive, spoken dialogue interface to navigate in virtual worlds, lexical, syntactic, and semantic issues. In J. Vince and R. Earnshaw, eds. *Virtual Worlds on the Internet.* IEEE Computer Society Press.

Java Speech API Programmer's Guide, Version 1.0, October 26 1998, URL: http://java.sun.com/products/java-media/speech/forDevelopers/jsapi- guide/index.html.

Jurafsky, D and Martin, J. (2000) *Speech and Language Processing*. Prentice Hall, New Jersey.

Kallmann, M., Thalmann, D. (1998) Modeling Objects for Interaction Tasks. *Proc. Eurographics Workshop onAnimation and Simulation*, Lisbon, Portugal.

Larson, J. (2002) *VoiceXML: Introduction to Developing Speech Applications*, Pearson Education, Inc. New Jersey.

Levison, L. (1996) *Connecting Planning and Acting via Object-Specific reasoning*. PhD thesis, University of Pennsylvania.

Luin, J., Nijholt, A., and Akker, R. (2001) Natural Language Navigation Support in Virtual Reality. *Proc. International Conference on Augmented, VE and 3D Imaging.* Greece.

Mast, M., Kummert, F., Ehrlich, U., Fink, G., Kuhn, T., Niemann, H and Sagerer, G. (1994) A speech understanding and dialog system with a homogeneous linguistic knowledge base. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):pp 179 194,.

Mehdi, Q., Zeng, X., and Gough, N.E. (2003) Story Visualization for Interactive Virtual Environment. *ISCA 12th International Conference on Intelligent and Adaptive Systems and Software Engineering*. California.

Suliman, H, Mehdi, Q and Gough, N.E. (2001) Software Development for Reasoning and Cognitive NPCs. *Proc. of 3rd SCS Int. Conf. On Intelligent Games and simulation, GAME-ON,* London.

Tanaka, H., Tokunaga, T and Shinyama, Y (2001) Animated Agents that Understand Natural Language and Perform Actions. *PRICAI-02 Workshop on Lifelike Animated Agents.*

Winograd, T. (1972) *Understanding Natural Language*. PhD thesis. Massachusetts Institute of Technology.

Zeng, X., Mehdi, Q and Gough, N.E. (2002) Generation of A 3D Virtual Story Environment Based on Story Description. *Proc. of 3rd SCS Int. Conf. On Intelligent Games and simulation, GAME-ON ,* London.

Zeng, X., Mehdi, Q and Gough, N.E. (2003) Natural Language Inference Technology for Reasoning Visual Information of Virtual Environment. *Proc. of 3rd SCS Int. Conf. On Intelligent Games and simulation, GAME-ON 2003,* London.