Explorative Conversation for Guiding Tasks in Annotated Environments

Johannes Strassner and Sebastian Reinders

Fraunhofer IMK, Schloß Birlinghoven, D-53754 Sankt Augustin, Germany
{strassner,reinders}@imk.fraunhofer.de

Abstract

Conversation with real humans and interactions with complex virtual environments are important features of multi modal guides. Exploring incomplete conversational situations in virtual environments with different interfaces helps to understand the required context and to annotate required information to the environment on the fly. The system is based on an adapted chatterbot infrastructure. It consists of a database with interaction patterns defined in a XML based language, different conversational interfaces, a context server, virtual human modules and a virtual reality system.

Key words: Virtual Actors, interaction patterns, natural language conversation

1. Introduction

Following a guide through a museum, we observed that she gave only short introductions on her own. Most of her presentation was driven by the visitors asking questions about the exhibition, artifacts or artists. Interested visitors learnt more by their conversation with the guide, but there was not enough time to see everything in the museum. Others followed the guide in a passive fashion, but they saw almost everything.

Information needs to be told in a different way depending if the visitors are adults, elderly people or children. Adults are usually motivated on their own to visit a museum. They are more likely interested in facts. Children come their with their class or their parents and prefer an emotional presentation style.

Our system provides an infrastructure to drive a virtual guide by a sequence of interaction patterns. These patterns enable content authors to recommend responses to the system for different situations belonging to guiding tasks.

The application and natural language interaction are not completely defined at the beginning, but will be improved with the number of users interacting with the system. The system serves as testbed for multimodal conversation. End users can be integrated early at the beginning. A multimodal system is implemented by coupling a number of systems in a framework. We want to decouple the description of knowledge and interaction patterns from the implementation. Interaction patterns can be defined and evaluated, before final system modules are implemented. So system modules do not constrain the development of interaction patterns, e.g. by restricting the vocabulary.

2. Background and Related Work

Steels [10] hypothesis is that language emerges through self organization out of interactions of language users. Meaning is constructed as part of language games and supported by environmental stimuli.

Chatterbots [12] are programs, which make natural language conversation. They decompose text sequences on the basis of rules which are triggered by key words appearing in the input text. Responses are then *generated* by reassembly rules. The Chatterbot Alice (www.alicebot.org) uses AIML [1], an XML based markup language to assign replacement rules to text patterns. Information in the database is hidden from the user, so user unfamiliar with the system take her for more intelligent than she is. Alice can react to the user's input, only. So passive users can not be motivated to interact with the system. She does not actively push a conversation in a direction to match the intention of a conversation, even if AIML includes a topic tag and rules can depend on the last answer.

Virtual humans [6] have multi modal capabilities including language and nonverbal behaviour. Synthesized speech and lip-synchronization can be generated automatically from text with good quality. BEAT [2] uses the timeline from the text-to-speech system to add non-verbal behaviour to the speech using a pipeline of modules. To be believable the behaviour of an character can be supported by annotations added to the environment by agents or authors [4].

Synthetic actors can teach humans in a direct or indirect way as explored by Rickel et.al. [7]. Their synthetic actor Steve is used as a collaborative instructor or teammate can help students by providing feedback on their actions and asking them what to do next. He can also guide his human students around the virtual world, demonstrate task, guide their attention or play the role of a teammate whose activities the human students can monitor. As J.Rickel stated interactive virtual world provide a powerful medium for entertainment and experimental learning.

The applications developed at Disney [8] take use of skills people acquired in the physical world. Visitors are encouraged to share their experiences in the virtual world. Attention of user's is attracted by spectacular events, such as erupting volcanoes or burning towns. They restrict users' navigation by using embedded limitations such as storms. A fallback solution to navigation is provided by an autopilot mode. Immersion theaters are also used early in the development pipeline to support designers of virtual world applications.

Göbel [5] also presents a virtual reality framework used for interactive installations and storytelling. Applications are modeled by nodes organized in a distributed scene graph. Each node has a number of typed fields, which can be connected by a data flow graph. Nodes are instances of C++ classes. Sensors provide an interface to external devices. Storyboard mechanisms are used to implement large scale narrative installations. Events are triggered by the plot.

In Cavazzas' system [3] each character maintains its contribution to the plot in a hierarchy of tasks, already containing possible alternatives to solve a goal. Heuristic values can be attached to sub tasks to describe when a character will select a subtask. Integration happens in the virtual environment. Situations are not explicitly described but arise from the characters' interactions. Users can modify these situations, by suggesting actions to the characters.

Torres [11] characters are able to sense a number of stimuli provided by other characters. Their known state of the world is the set of stimuli at some time frame. A temporal memory system keeps a number of such world states. Then a pattern analysis will be used to find event dependencies.

Arafa [13] compares a number of scripting languages for virtual humans. Most of them take use of XML and concentrate on animation, behaviour and character definition. All languages try to be domain independent and to use high level abstraction.

3. Conversational System

We extended the AIML replacement rules by additional markups for character and world behaviour. An adapted version of the Alice chatterbot parses the text input using the current interaction context and composes a response including markups to drive behaviour of a multi modal character. Context is defined by a number of variables, such as the current topic and location, the visitor's name and age. It includes the guide's last answer and the user's current question. Users explore the capabilities of the system while they are guided through the virtual world. Missing or incompletely defined interaction units are detected during the conversation with the guide. We equipped the Alice chatterbot with functions to make context accessible to other system components. Context can not only be monitored but also be set by these components. Doing so, it's possible to revisit any situation the bot has been in before and to explore it in detail. Note, the response to the same situation can be different, if variability is defined in the interaction unit. To meet guidance requirements, we adapted the chatterbot such a way that it still supports off-topic talk, but will switch the topic back to the current location when the conversation stuck. Furthermore it's now able to trigger itself, when the user remains silent. Progress will be rated by a number of counters queried and changed by a new chatterbot function.

3.1 Interaction patterns

While navigating through the virtual world, people write interaction patterns for situations they are exploring. Such situations contain natural speech interaction and need appropriate reaction of a multimodal character. The interaction pattern integrates all relevant parameters to produce the multimodal output. By assigning different rules to a pattern a variety of reactions can be produced. Building multi modal systems requires establishing concepts for the synchronization of speech with lip movements, emotions and gestures. Gestures serve different functions during natural language conversation. Depending on the guides' personality and internal state gestures will be chosen. Emotions also depend on the internal state. Gestures, which look similar, but contain small variations, belong often to the same pattern.

Authors need to experiment with the patterns and the multimodal vocabulary needed in the different situations. The vocabulary is not fixed and will be developed while defining the interaction patterns. Authoring starts with closely integrated patterns. Descriptions are often on an abstract level and will be incrementally refined or replaced. A number of templates results from the exploration and authoring process, which can be reused for other topics and applications. Every interaction pattern can be equipped with a number of selection tags with different priority. Such tags include the category, the topic or the last answer given by the bot. The pattern matching procedure uses these parameters to find the pattern best fitting the input.

Figure 1 shows a pattern to start the museum's tour. The character needs to be moved to the start position. The variables "location" and "wright" will be created and initialized. The guide has 3 options to welcome the visitor. The "topic" will then set to "wright".

```
<pattern>START THE TOUR</pattern>

 <template>
   <body>Go to Start</body>
   <set name="location">the Entrance Room</set>
   <set name="wright">0</set>
  <random>
      <li>Welcome to the Frank Llyod Wright
      Tour.</li>
     <li>Hi there. My Name is Guide. I'm your
     Guide.</li>
    <li>This is the Frank Llyod Wright Museums
     Tour. My Name is Guide.</li>
  </random>
  <set name="topic">wright</set>
 </template>
```

Fig. 1 Example: Interaction pattern

Interaction patterns can be used for natural language interaction, agent communication, or will be implicitly triggered by applications. Multi modal systems consists of a number of modules. Interfaces between the different components depend on interaction patterns which need to be implemented to support natural language communication.

Interaction patterns implement a number of design decisions and requirements to realize the application. In our museum application the user is free to look around or to move away from the guide while the guide is still talking about an artifact. If the user moved by a certain distance the guide will try to shorten it's current presentation. Text needs to contain interruption marks, to break a presentation at useful points. If the user asks a question while the guide is still talking, the guide will stop after the next word. If the user asks to change the location, the guide will not immediately move to another artifact, but implicitly reduce the amount of knowledge, which needs to be talked about at a certain topic. If the guide asks a question and the user reacts to the questions, an topic score will increased by some progress points and the conversation continues. If the user is interested in a topic he can ask for more information. The conversation flow is supported by using state variables embedded in interaction patterns or by creating a dependency to the guides' last answer. To prevent repetition of questions a state variable will be defined, when the question has been asked or the visitor has reacted to the question. Furthermore a timestamp can be used to prevent questions from being asked to often. The guide should not interrupt the user while talking to the guide. The user needs some time to react to the guides' questions.

3.2 System Architecture

The system (Fig.2) consists of a number of components which are loosely coupled by the context server. The different modules produce the data at different time lines. The input to the system and the most of its' modules consists of text. In our application the user is

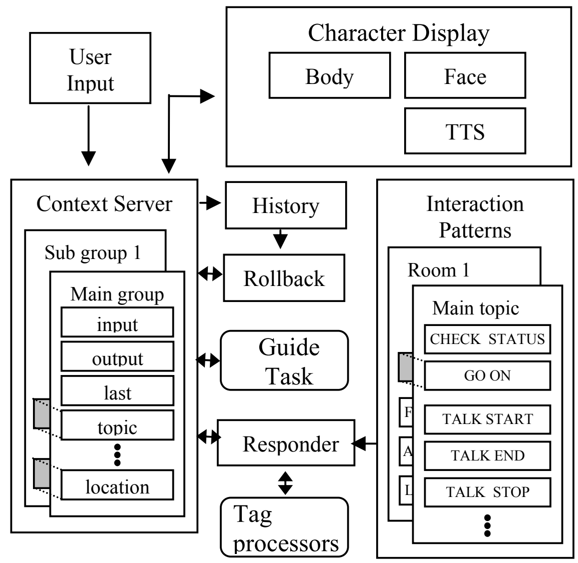

Fig. 2 System Architecture

using a keyboard to talk to the guide at any time. When a robust speech recognition is available it can be easily connected to the system without changing the other modules. Most of the devices and sensors connected to the VR framework implicitly produce text with tags. Natural language input is transformed into a sequence of interaction patterns using the responder. It's also possible to directly call an interaction pattern. Each system component can use its own markup language to describe its results.

The *context server* maintains a number of typed fields. Fields are organized into communication groups to serve the need of the clients. A field can be member of different groups at the same time. The server provides clients with functions to create fields, add a field to a group, to remove it from a group or to change the values of fields. Clients use these functions by sending XML messages to the server. Clients can register for groups to be informed if fields are changed. Individual update rules can be attached to groups. The server is implemented in Java, but since all communication is done by XML objects clients only need to implement a socket connection. Clients are currently available for Linux and Windows platforms implemented in C++ and Java.

The *responder* transforms any text input into a sequence of interaction patterns provided by the database. It is based on the Alice chatterbot. Furthermore it's also possible to directly call an interaction pattern. The responder's output is then composed using the rules associated with the interaction patterns. The responders' state is defined by a variable number of variables, which will be automatically mirrored to the context server. Other clients connected to the context server can control the state by changing the variables. The variables are changed while the responder processes the input. The output can then be displayed by different clients connected to the server. If the output is interrupted the responders state needs to be repaired by an external component.

The responder uses a number of tag processors to interpret input enclosed by tags. E.g. a *set processor* creates a variable or changes its' value. To adapt the chatterbot to our purpose we replaced the set processor by our own version which also updates the context server. Additionally it keeps the tags in the response, so a later *rollback* can reset the values using the *history*, if the sentence did not get spoken completely. Our *score processor* provides a number of functions to use counters and to check their limits. Variables are automatically registered to different groups at the context server by the processors handling the different tags. E.g. the score processor registers it's variables to a group "progress". Tags which can not be handled by the responder, will be included in the generated response.

The response will be scheduled to be played by the different display modules depending on the current system configuration. The *character display* includes a number of modules, such as lip-synchronized speech, mimics and body movements. Lip-synchronized speech is produced using the AT&T text and speech server. The characters can look at static and dynamic targets using inverse kinematics. Pointing at artifacts is realized using a motion library. Body movements are triggered by interaction patterns. World *effects* are implemented in Avango. They consist of a combination of a instance of a C++ class and a scheme script interpreted at runtime. The script is bound to the fields of the C++ object by a callback function. The fields of the Avango runtime objects are automatically parsed by a connection service client and a group is created at the context server. Fields are added to the group and the group is connected to the Avango object. Field changes of the Avango object are sent to the server and messages from the server affects the Avango object.

Different system components can react to events at different timelines. The text-to-speech system requires a part of a sentence up to a punctuation mark. Emotions last for several words. Resolving coarticulation effects requires several phonemes to assign the correct visemes to speech. The responders answer might consist of several sentences. Interrupt of content currently displayed depends on multimodal constraints by the different modules. The text contained in the interaction patterns has to be equipped with interruption marks.

The responders' output need to be synchronized with the display modules. This is done by the patterns "TALK START", "TALK END", "TALK STOP". These patterns tell the responder that the output has been started, it is finished or the output should be interrupted.

An independent client, the *guide task* controls the conversation by watching communication activities and sending interaction patterns to the responder. It is connected to two subgroups at the context server modeling user activity and progress. If the user is inactive for a time defined by the variable "timeout", the guide task sends a "GO ON" to the "input" field. The pattern "GO ON" needs to be provided for every topic. It's used to motivate the user e.g. by telling stories or asking questions. The guide also reacts to field changes in the group "progress" by sending "CHECK STATUS" to the responder. In another mode "CHECK STATUS" will be called regularly. The pattern "CHECK STATUS" is implemented in the main topic. It checks scores and current location of the tour and changes the topic after it contributed enough knowledge.

4. Exploration

Different conversational interfaces are provided, which include free text conversation, a multiple-choice interface and a guidance interface. All interfaces result in text input which is parsed by the chatterbot to generate the system response.

4.1 Free Conversation

The user talks to the guide by typing questions and comments on a keyboard. The input will be interpreted by the chatterbot, which drives the guide's output.

Figure 3 shows a part of a communication. The guide welcomes the user and fills context variables with the user's name and gender. She correctly answers a couple of questions. But then there was nothing about the number of Wright's wives in the data base and so she failed here. Later we extended her knowledge about Wright's family. At the end of the conversation, she continued to explain Wright's idea, when the user further enquired.

4.2 Multiple-Choice Interaction

For a number of situations, it is useful to constrain the answers of the user. For this purpose a multiple-choice board (see figure 6) can be moved in the view of the user. A number of statements or options defined in an interaction pattern will be displayed on the board. The board can be configured to allow single or multiple selections. Furthermore there are two buttons for emotional feedback included, so the user can tell the guide that he likes or dislikes the tour.

It can be used in a variety of ways. The guide can ask the user a question and the user selects the answer from the multiple choice board. The selected answer will be interpreted by the chatterbot which drives the system. To test the user in a learning situation, correct and wrong answers can be shown on the board. The user can select a number of interests, configure the presentation style of the guide or to constrain the context in a menu-like fashion.

Guide: Hi there. This is the Frank Llyod Wright Museum's Tour. My Name is Guide. How can I call you?
Visitor: Call me Johannes
Guide: Hi Johannes . Are you a man or a woman?
Visitor: Im a man.
Guide: Thanks for telling me your gender, Johannes.
Guide: Most of my clients are men.
Visitor: Could you tell me where I am?
Guide: You are in the virtual Museum of Frank Lloyd Wright. This is the Entrance Room.
Visitor: What was his profession again?
Guide: Once more? He was a famous american architect.
Visitor: Is that him on the left picture?
Guide: Yes, thats him.
Visitor: Why do you think he was famous?
Guide: He worked hard on his buildings, but he worked hard on his fame and public image, too.
Visitor: How many wives did he have?
Guide: Around 100.
Visitor: Do you like him?
Guide: Hmm. I've never met him personally but I like his creations and the Idea behind them.
Visitor: Which Idea?
Guide: To redefine the home of a family as practical and comfortable living space.

Fig. 3 Example conversation

4.3 Remotely Guided Conversations

A remote person can directly control the guide and react to the user by using this interface. It allows to explore the interaction potential of different situations. Furthermore it's possible to run Wizard of Oz experiments by directly triggering interaction patterns or sending text to the multimodal guide.

The interface (see figure 4) consists of a situation interface and a text window. Text or XML typed in the text window will be sent to the server. The situation interface enables the user to browse through different situations starting at an entry point. It is separated into an hierarchical organization of available actions and a background map showing a setup of a room augmented by a number of icons. It represents the interaction potential of a situation for a guide with a certain personality and presentation style for a visitor type. Switching the presentation style will result in the same map with a different set of buttons. Depending on the personality of the guide, a situation provides a number of actions. A guide can move to a different location, show different mimics, make different gestures or the situation can be changed. Changing the situation or

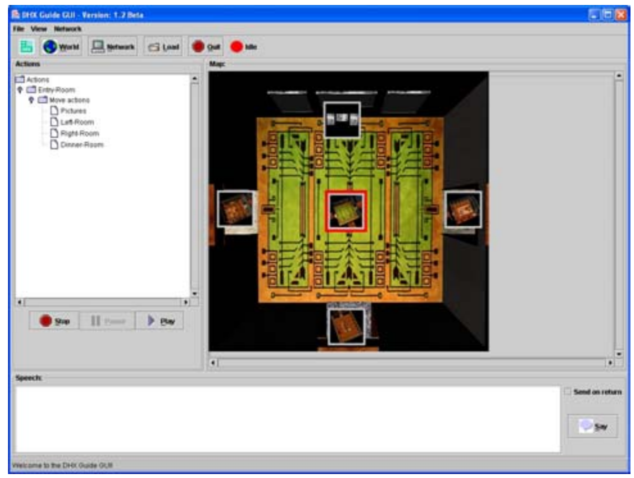

Fig. 4 The Remote Guide GUI

moving to a new position will result in displaying a new map with buttons accordingly.

5. Experimentation

The system has been tested with a virtual museum for the architect Frank Lloyd Wright. The tour starts in a room containing 3 pictures of Wright. This room is used to introduce the user in the system and to learn about the architect. The museum consists of 3 more rooms, containing furniture, wallpapers, windows, newspapers and photos. The interaction pattern database has been built by an author annotating knowledge to some artifacts in the environment and collecting knowledge about Wright. The knowledge database has been tested in several sessions by 4 primary users. After analyzing the log files the author adapted and extended the interaction patterns.

After setting up the basic system, 10 users, 2 of them were native English speakers, explored the museum. The experimentation took place in two steps. First with a limited system and then with a full system in the virtual environment.

5.1 Desktop System

In a first test step the users were talking with the guide system using a standard terminal. The tour started in the entrance room. They were able to move around in the museum using a joystick. Navigation was constrained to prevent collision with the furniture and to keep the eye height. Users communicated with the guide bot typing in text in a chat window. The bot's answer was displayed in the same window. In this setup no 3D character was included and the view of the user was not connected to the state of the tour.

The users received a short introduction from a person familiar with the system. The test candidates should learn how the guide bot works and to understand its limits.

They have been asked to type full sentences as they would talk with a real person. The chatterbot answers every sentence independently. To prevent inconsistencies people need to type only one sentence at a time. The instructor also mentioned to use the joystick to turn around and to take a closer look at the pictures. A few users did not get started, so the instructor gave them an example what they could ask. Other people tried later to get background knowledge about Wright from the instructor. They got advised to ask the bot the same question using their keyboard. The instructor left they room after watching the first steps. He came back after some time and asked the user, if he already found out something about Wright.

The guide was reacting to the user's questions. When they felt silent he tried to get their attention by asking questions about them and their motivation to visit the virtual museum. He was also suggesting topics, quoting Wright and telling short stories. Most people figured out on their own that the conversation got more interesting, if they also reacted to questions and not only asked questions. None of the users did not react to Wrights cites. Stories combined with questions were much better to get their attention. The users liked it to be involved in stories. Most of them tried e.g. to figure out who murdered Wright's second wife or why he has been arrested once. It was appreciated that the system was telling the knowledge not at once but in parts.

Sometimes people complained they were not able to figure out how to drive the guide to tell them more about a topic. Most of them changed in such situations their way to communicate with the guide in order to reach a common ground. When they had no success, they tried first to use keywords instead of full sentences, then to express their question with different words and finally to use easier words. Using keywords have been prohibited by the instructor, to allow application authors to learn about the variety of this interaction form. Patterns consist of several words to force people to use full sentences. After being asked the users also agreed to this strategy and acknowledged that they prefer to use full sentences. One user typed in a situation, in which the system failed finally frustrated: "Thanks", which got misunderstood by the system. It assumed that the user was polite. But it correctly figured out that the conversation reached a dead end and changed the topic. This was one of the strategies we added to the original chatterbot. The conversation could continue. People were not disturbed by the bot's strategy to switch back to the topic at dead ending off-topic talk.

It did not matter to most of them, if they received sometimes a wrong answer. We had the impression that this sometimes even motivated the people. On the other hand users donot want to get stupid answers, especially if they answer a question in a serious matter (see figure 5).

```
Marion > Which important buildings did Wright design?
Guide > The large  one, I think.
Marion> Which houses did Wright build?
Guide > The first  one, I think.
Marion> When did Wright live?
Guide > Some time ago.
```

Fig. 5 Example: Wrong answers

An user got offended by the sexual content in one of the answers. Another used rude words and the system annotated him with a rude personality.

6 persons succeeded to get the required score of 10 points to get in the next room. 2 persons gave up with 8 and 9 points. Another person reached only 4 points because of his English. One person expected a more intelligent guide and left.

After successfully completing the task of the first room, the guide bot started to talk about some artifacts in one of the exhibition rooms. The user had to search the objects the guide bot was talking about on his own, since the system was not controlling his view. Most of the users got lost and gave up.

5.2 Virtual Environment

At the next step of the experimentation, people continued at a 2-sided Workbench, a Responsive Workbench with an additional back wall. The guide bot was now driving a life-size virtual human appearing in front of the back screen. Lip-synchronized speech is produced using the AT&T text and speech server. The characters could look at static and dynamic targets and point at artifacts using a motion library. Body movements were triggered by the interaction patterns. The users experienced it as more motivating for their text conversation with the guide than in front of the screen. The guide was able to make the virtual world alive and to direct their attention. People could navigate around in the current room using a joystick, but were not allowed to leave the room or the location connected to their current topic on their own. After reaching the score for the current topic, the guide moved to the next room taking the user with him. When the user was moving around using the joystick he was implicitly driving the guide to come with him. The person used an illuminated keyboard for text input. The text has been displayed on a small virtual monitor so the user could see the text while they are typing. Additionally they could use a stylus to select objects, if this was supported by an interaction

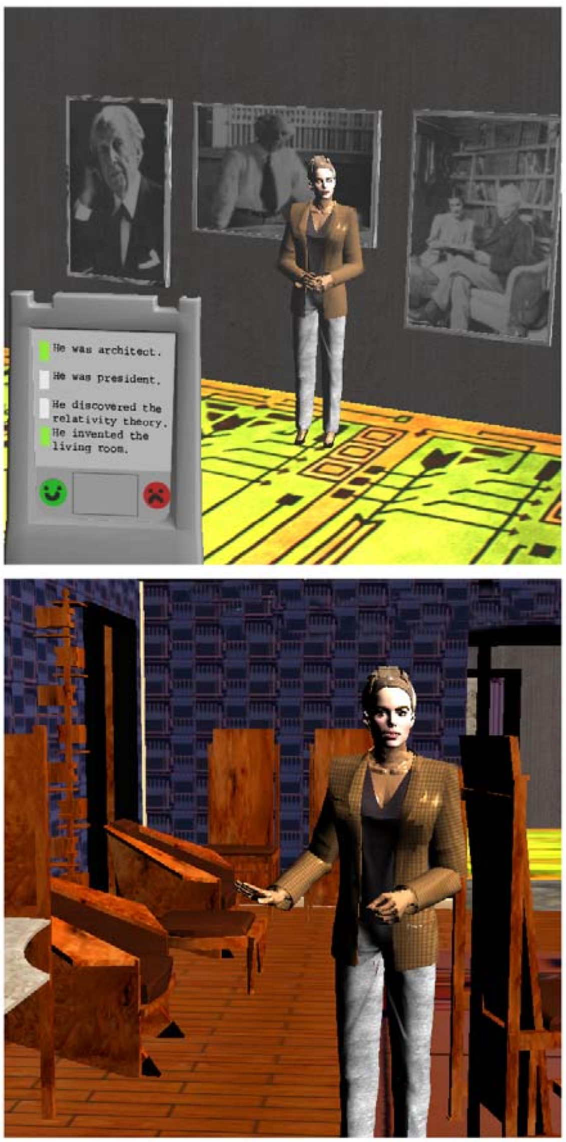

Fig. 6 Up: Multiple-choice board. Down: The guide talks about the chairs

pattern belonging to their current topic.

People already used different interaction and navigation devices and virtual tools in several virtual world applications. They experienced text as an additional interaction form and liked not to be limited by a menu like interaction style.

Users found it motivating to give emotional feedback.

Guide> Here we have an interesting desklamp. Do you like it?
Klaus> No.
Guide > What's wrong with it?
Klaus > It is ugly.
Guide > What makes it so ugly?
Klaus > The trees

Fig. 7 Example: Emotional Feedback

A test has been implemented by providing a number of patterns belonging to the topic "test". The guide started the test by asking the user to repeat what he has learned about Wright. Depending on a condition variable reflecting the user's preferred interaction style the user could use natural speech or a multiple-choice board. In the natural language mode the system used pattern matching to rate the user's answers. Instead the slots of the multiple-choice got filled with a number of correct and wrong answers (see fig. 3). The multiple-choice board appeared in the view of the user and the user selected a number of answers using the stylus. As expected the natural language approach failed a couple of times and not all the answers of the users got correctly rated. The multiple-choice was more robust, but the answers were limited to the displayed choices.

In test situations all users preferred accurate feedback and a robust rating mechanism. Even if they would like to have free conversation, they decided for the multiple-choice board. In situations, in which success is less critical, the free conversation was more fun, because it supports the exploration of hidden knowledge embedded in the virtual world.

Users spend more time in the virtual environment than in the desktop version. Log files of their conversation in the virtual world consists of 3.5 up to 12.5 pages.

6. Conclusion

We presented a natural language system based on an adapted version of a chatterbot to explore incomplete conversational situations in virtual environments. The multimodal behaviour of a virtual guide is driven by a sequence of interaction patterns. These patterns integrate all relevant parameters to establish concepts for the synchronization of speech with lip movements emotions and gestures. Authors experiment with closely integrated interaction patterns to define the behaviour of the guide. The knowledge of the guide is improved with the number of users interacting with the system. The system has been tested in 2 experiments with a number of users. . Most of the people enjoyed the natural conversation in the virtual environment, even if they preferred a more reliable interface in test situations. In the future we will extend the interaction pattern concept and integrate the system with more applications and autonomous system components handling e.g. emotions of the guide to make his behavior more realistic. We will also experiment with different interfaces which can be used together with the natural language communication to allow more robust interaction with the guide.

References

[1] Aiml reference manual. URL:http://www.alicebot.org/aiml.html.

[2] Justine Cassell, Hannes Högni Vilhj´almsson, and Timothy Bickmore. BEAT: The behavior expression animation toolkit. In Eugene Fiume, editor, *SIGGRAPH 2001, Computer Graphics Proceedings*, pages 477–486. ACM Press / ACM SIGGRAPH, 2001.

[3] Marc Cavazza, Fred Charles, and Steven J. Mead. Interacting with virtual characters in interactive storytelling. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 318–325. ACM Press, 2002.

[4] Patrick Doyle. Believability through context using "knowledge in the world" to create intelligent characters. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 342–349. ACM Press, 2002.

[5] Göbel, M., Barrass, S., Eccles, J., Eckel, G., Hasenbrink, F., Lechner, U., Mostafawy,S., Toth, S., Tramberend, H., and Unnützer, P. Digital storytelling - creating interactive illusions with avocado. *9th International Conference on Artificial Reality and Teleexistance*, 1999

[6] J. Gratch, J. Rickel, E. Andre, J. Cassell, E. Petajan, and N. Badler. Creating interactive virtual humans: Some assembly required. *IEEE Intelligent Systems*, pages 54–63, 2002.

[7] J. Gratch R. Hill D. Traum J. Rickel, S. Marsella andW. Swartout. Toward a new generation of virtual humans for interactive experiences. *IEEE Intelligent Systems*, 17(4):32–38, July/August 2002.

[8] Mark Mine. Towards virtual reality for the masses: 10 years of research at disney's vr studio. In *Proceedings of the workshop on Virtual environments 2003*, pages 11–17. ACM Press, 2003.

[9] Mark Riedl, C. J. Saretto, and R. Michael Young. Managing interaction between users and agents in a multi-agent storytelling environment. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 741 748. ACM Press, 2003.

[10] Luc Steels. The talking heads experiment. words and meanings. Photography Museum, 1999.

[11] Daniel Torres and Pierre Boulanger. The animus project: a framework for the creation of interactive creatures in immersed environments. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 91–99. ACM Press, 2003.

[12] JosephWeizenbaum. Elizaa computer program for the study of natural language communication between man and machine. *Commun. ACM*, 9(1):36–45, 1966.

[13] Ebrahim Mamdani Yasmine Arafa, Kaveh Kamyab. *Toward a Unified Scripting Language: Lessons Learned from Developing CML and AML*, pages 39–63. Springer, 200.