

# Interactive Agents Learning their Environment\*

Michiel Hildebrand, Anton Eliëns, Zhisheng Huang and Cees Visser

Intelligent Multimedia Group  
Vrije Universiteit, Amsterdam, Netherlands  
{mhildeb,eliens,huang,ctv}@cs.vu.nl

## Abstract

In this paper we describe the implementation of interactive agents capable of gathering and extending their knowledge. Interactive agents are designed to perform tasks requested by a user in natural language. Using simple sentences the agent can answer questions and in case a task can not be fulfilled the agent must communicate with the user. In particular, an interactive agent can tell when necessary information for a task is missing, giving the user a chance to supply this information, which may in effect result in teaching the agent. The interactive agent platform is implemented in DLP, a tool for the implementation of 3D web agents. In this paper we discuss the motivation for interactive agents, the learning mechanisms and it's realization in the DLP platform.

**Keywords:** *Interactive Agents, Virtual Environments, Natural Language, Learning, DLP, VRML, Grail.*

## 1 Introduction

Research done in the combined fields of computational linguistics, computer graphics and autonomous agents has led to the development of autonomous virtual characters. These characters will make interaction with our computer systems more natural, as an interface for existing systems and as a base for new applications. The use of autonomous characters in virtual or physical musea, shops, information kiosks or public buildings are just a few of the possibilities.

Autonomous agents with a humanoid appearance and autonomous behavior provide a user friendly alternative to traditional interfaces. Agents may perform actions, display information or, in addition, gather information as well. They use interaction to learn about their environment by adding or modifying their knowledge. The benefit of this approach is that agents need not be given all information in advance but can instead built up their knowledge during the proces.

Natural language provides an easy and flexible way to communicate. We can obtain information with questions and add information with declarative sentences. In response the agent uses simple sentences to answer questions and

---

\*<http://www.cs.vu.nl/~eliens/research/media/title-interactive.html>

to communicate about the information he is missing. The embodiment of agents allows movements and gestures. How interaction takes place will be shown in more detail in section 2, where we describe a sample scenario.

Our system is implemented in DLP [2], a distributed logic programming language suited for the implementation of 3D intelligent agents [3]. Avatars for the agents are built in the Virtual Reality Modelling Language (VRML) or X3D, the next generation of VRML. These avatars have a humanoid appearance based on the H-anim specification<sup>1</sup>. The gestures for the agent are made in the STEP scripting language [4]. Natural language processing is done by *grail*<sup>2</sup>, a parser based on categorial grammar [11].

The structure of this paper is as follows: In section 2 we demonstrate how users and agents can interact with each other. Section 3 provides a formal description of interaction and learning. The realization of the platform is described in section 4. In section 5 we describe other work related to interactive agents. Section 6 ends the paper with a conclusion and suggestions for further research.

## 2 A Sample Scenario of Interaction

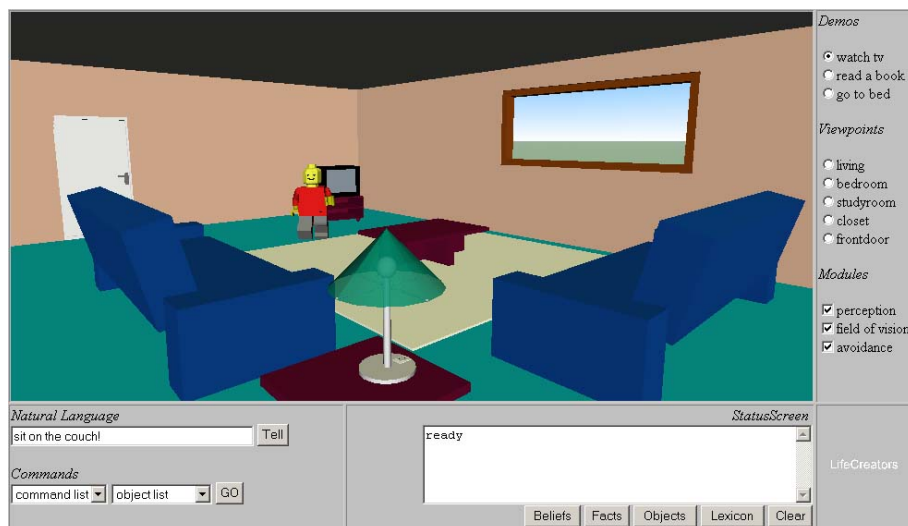


Figure 1: Interactive agent situated in a virtual house

Figure 1 shows the application as it is run in a web browser. The main screen contains the virtual world including the agent. The interface is created with standard html forms. At the bottom it contains a field for language input, two selection menus for command shortcuts and a status screen with four buttons to display the agents knowledge. On the right side there are various shortcuts to demo actions and predefined viewpoints, and a selection list of active modules.

A user can give natural language input to an interactive agent by typing English sentences. There are three types of sentences possible: commands,

<sup>1</sup><http://h-ahim.org>

<sup>2</sup><http://www.let.uu.nl/~Richard.Moot/personal/grail.html>

questions and declarative sentences. To indicate the type of a sentence the marks, respectively (!,?,.), have to be used. The following sentences present a sample of possible inputs:

*Switch on the TV!*  
*Sit on the blue couch!*  
*Sit on the table!*  
*Where is your bed?*  
*Can you give me a book?*  
*Yes, you can sit on the table.*  
*There is a book inside the studyroom.*

Given a command the agent will try to perform the corresponding action. In response to the command *Switch on the TV!* the agent will walk to the TV and raise his hand to switch the power button. The command *sit on the couch!* is less straightforward, because there is more than one couch in the livingroom. If a specific couch was part of the interaction before, the agent will continue using the familiar couch. Otherwise the user has to be more specific and the agent will prompt him to tell which couch is meant. A user can describe the object by adding extra properties about it, and for instance say: *sit on the blue couch!*

Instead of multiple possibilities it could also be that the whereabouts of an object are unknown. To locate an object the agent first looks around. If no object is found he will ask the user for help. The user can in response give directions to explain where the object is located. If for example the agent asks where to find a book the user can respond that *there is a book inside the studyroom*. Information exchange could also be reversed. A user can for example ask *where is your bed?* and the agent will respond that it is inside the bedroom.

Interactive agents contain knowledge about their environment. In situations where common knowledge is not sufficient to make a judgement the agent will communicate with the user. For example an agent will not immediately sit on a table if requested. Instead he will tell the user he is not sure sitting on tables is allowed. The user can confirm, for example if all seats on the couch are taken, and say: *Yes, you can sit on the table*. The agent adds this new belief to his knowledge and takes place on the table. In this case the user takes the role of an instructor to teach the agent.

### 3 Formal Description of Interaction

Input can result in three types of actions according to the types of sentences we distinguished. A command will result in a basic action, a physical action in the environment. Questions can be used for two different reasons: to ask information, *Is there a book on the table?* or to request an action, *Can you give me the book?* The former results in an answer action while the latter results in a basic action. Declarative sentences are used to give information that agents can use to modify their knowledge, a learning action.

To control the performance of actions they are embedded in conditions and effects, in a similar way as in *capabilities* used in [9], [8]. The conditions check if the actions are possible in the current state. Only if all conditions are satisfied the actions are performed. The effects update the agent's knowledge according

to the changes made in the environment. The actions itself consist of calculations, physical actions, text outputs or references to other actions.

A state, which we will denote with the letter  $S$ , is determined by the agents knowledge. The knowledge of interactive agents consists of objects and their properties. We characterize a state as a mapping from object to (property, value) pairs. Given a state  $S$  an agent can perform the actions if the corresponding conditions  $C$  can be satisfied in that state. Formally we write:

$$\langle S, C \rangle \xrightarrow{a} S',$$

where  $a$  is the observable behavior of the agent in his environment. The new state  $S'$  is obtained by modifying the agents knowledge in  $S$  by the effects.

**Learning.** A learning action has no observable behavior. Learning affects the actions an agent can perform. In other words, we can describe a learning action as a transition from one state to another changing the set of possible actions. For example, the permission to sit on a table extends the actions the agent can perform, by adding the possibility to sit on tables.

## 4 Realization

The agent is built from components as depicted in figure 2. The components are implemented in the distributes logic programming language DLP [2]. The avatar and the virtual world are constructed in VRML. Interaction between the agent and his environment is possible because DLP has programmatic control over VRML through the External Authoring Interface (EAI) [2]. In particular DLP can get/set values from objects through handles.

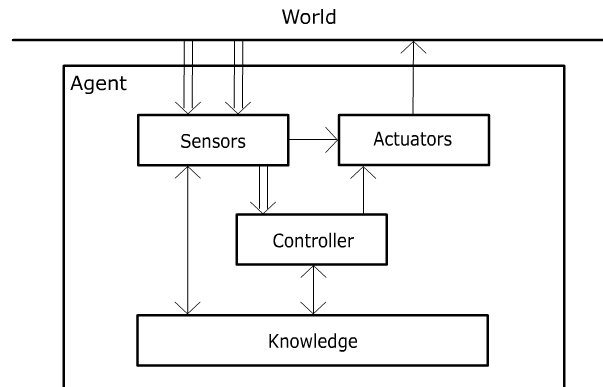


Figure 2: Agent components

All objects in the VRML world are created according to a prototypical structure. This structure guarantees that each object has fields for *position*, *rotation*, *center*, *scale*, *size* and *name*. The sensor component checks the values of these fields and stores them in the knowledge base. Perception is restricted in the sense that the agent can only get information about objects located in the same room and positioned in his field of vision.

The agent has a humanoid appearance based on the H-anim standard. The H-anim standard describes a humanoid as consisting of a large number of joints. In our system we use a simplified version with only 7 joints. The agent's body movements are defined with STEP, a scripting language for embodied agents based on dynamic logic [4]. The actuator component contains the STEP kernel to execute these scripts.

The knowledge component contains all data that may change from one agent to another. Therefore to create an interactive agent only this knowledge has to be defined, leaving the other components untouched. The actions defined by STEP scripts, beliefs and the lexicon are all treated as knowledge. Currently only beliefs can be learned. However, the system is constructed in such a way that it can be extended to learn the other types of knowledge as well.

Beliefs gathered through perception give an objective view of the world, since their values are determined by the properties of objects. Subjective beliefs are gathered or modified through interaction with the user, in the form of learning actions. Beliefs can also be given in advance to minimize the need for irrelevant interaction.

Given an input the controller creates the appropriate behavior based on the agent's knowledge. Input from the command shortcuts can be directly performed by a basic action. Input given in natural language is first parsed, as described in the final section. The term that results from parsing may contain quantified or free variables. The quantified variables are substituted according to the agent's beliefs. Further processing is done according to the three types of actions.

An answer action creates an appropriate answer by matching the term with the agent's beliefs, substituting free variables. A learning action updates the agents beliefs either by modifying or adding beliefs. A basic action is a collection of conditions, actions and effects. As an example a basic action to switch an object on or off (TV or lamp) can be described as<sup>3</sup>:

```
action(switch(Object, Status), Conditions, Actions, Effects) :-
  Conditions = [
    property(Object, [position,X0,Y0,Z0]),
    property(Object, [rotation,R0]),
    property(belief, [switch,Object,Button]),
    property(belief, not([status, Status])),
    property(Button, position(Xswitch,Yswitch,Zswitch))
  ],
  Actions = [
    get_in_room(Object),
    get_in_reach(X0,Y0,Z0, Xswitch,Yswitch,Zswitch, R0, rightHand),
    arm_rotation(Y0, Yswitch, Rarm),
    action([switch, Switch, Rarm])
  ],
  Effects = [
    change(Object, [status,Status], [status,not(Status)])
  ].
```

---

<sup>3</sup>For clarity the fail actions from the conditions are removed. The position property has a try action to find the object. The status property outputs that the object already has the desired status.

The position and rotation properties are looked up in the object's perception data. In the beliefs a lookup is done to determine which button can be used and if the object does not already have the desired status. The actions make the agent move to the same room as the object and place him in reach of the switch button. After the necessary rotation of the agent's arm is calculated the physical switch action can be performed. Finally the effects modify the agent's beliefs by changing the status of the switched object.

The natural language parser, *grail*, is based on categorial grammar [11], that can be characterized by the slogan 'parsing as deduction'. A categorial grammar is given by an assignment of types to the words in the lexicon. The types are constructed from atoms and logical connectives. An expression is well-formed if a derivation in the logic for these connectives is possible. The meaning of an expression is then assembled according to the derivation [1].

## 5 Related Work

The Gesture and Narrative Language group have created several embodied conversational agents [5, 6]. They focus on controlled dialog with multi-modal input and output. In addition their system uses camera's and speech recognition.

In [10] Virtual Teletubbies are developed to create believable agents. These agents combine long-term goals with low-level reactive behavior. Virtual physics are used to create realistic environments.

At the Synthetic Creatures group virtual agents are used to simulate animal behavior [7]. Their implementation of a sheepdog uses reinforcement learning to associate actions with utterances (single words or clicks). Both their reasoning and learning behavior is very well suited for modelling animal intelligence.

Jacob, the animated instruction agent is a project from the University of Twente [12]. Jacob instructs and assists a user to learn to perform in a virtual environment. The user can communicate with Jacob through natural language.

Just-Talk is an application to train law enforcement personal in a virtual role-playing environment [13]. The virtual personality uses a combination of facial gesture, body movements, and spoken language. The students can converse with that personality using spoken natural language.

In comparison with the work just mentioned we may characterize our project as focusing on natural language interaction between an agent and the user. The agent uses logic-based learning to adapt to his environment.

## 6 Conclusions and Future Research

In this paper we have described a system where users interact with agents using natural language. In response to user input the agent can perform physical actions, give answers to questions or learn new information. If non of the three actions are possible the agent generates an output message to indicate what information is missing. This approach makes it possible that the agent can learn to act in unfamiliar environments.

Currently the agent can learn by modifying his beliefs. In further research the system can be extended to learn other kinds of knowledge as well. For

example actions can be learned by combining existing actions, or by defining simple STEP scripts.

An altogether different challenge is to apply this approach to another application domain. The INCCA (International Network for the Conservation of Contemporary Art) has developed a multimedia repository about contemporary art. In such an application an agent can aid the user in navigating the information space and create presentations about the INCCA repository, on demand of the user.

## References

- [1] Moortgat, M. (2002). *Categorial Grammar and Formal Semantics*. Encyclopedia of Cognitive Science, Nature Publishing Group, Macmillan Publishers Ltd.
- [2] Eliëns, A. (1992). *DLP, A Language for Distributed Logic Programming*, Wiley
- [3] Eliëns, A., Huang, C., Visser C. (2002). *A platform for Embodied Conversational Agents based on Distributed Logic Programming*. AAMAS Workshop – Embodied conversational agents - let's specify and evaluate them!, Bologna 16 July 2002.
- [4] Huang, Z., Eliëns, A., Visser, C. (2002). *STEP: A Scripting Language for Embodied Agents*. PRICAI-02 Workshop – Lifelike Animated Agents: Tools, Affective Functions, and Applications, Tokyo 19 August 2002.
- [5] Cassell, J., Bickmore, B., Campbell, L., Vilhjalmsón, H., Yan, H. (2001). *Conversation of a System Framework: Designing Embodied Conversational Agents*. Embodied Conversational Agents, pp. 29-63. MIT Press.
- [6] Cassell, J., Stocky, T., Bickmore, T., Gao, Y., Nakano, Y., Ryokai, K., Tversky, D., Vaucelle, C., Vilhjalmsón, H. (2002). *MACK: Media lab Autonomous Conversational Kiosk*. In Proc. of Imagina '02, pp. 12-15, Monte Carlo.
- [7] Isla, D., Burke, R., Downie, M., Blumberg, B. (2001). *a Layered Brain Architecture for Synthetic Creatures*. In Proc. of the Int. Joint Conf. on Artificial Intelligence (IJCAI), pp. 1051-1058, Seattle.
- [8] Hendriks, K., de Boer, F., van der Hoek, W., Meyer, J-J. (1999). *Agent programming in 3APL*. Autonomous Agents and Multi-Agent Systems, 2(4), pp. 357-401.
- [9] Panayiotopoulos, T., Anastassakis, G. (1999). *Towards a virtual reality intelligent agent language*. 7th Hellenic Conf. on Informatics, September 1999, Ioannina.
- [10] Ballin, D., Aylett, R., Delgado, C. (2002). *Towards Autonomous Characters for Interactive Media*. Intelligent Agents for Mobile and Virtual Media, Ch. 5, pp. 55-76. Springer.

- [11] Moot, R. (2002) *Proof Nets for Linguistic Analysis*. Ph. D. thesis, Utrecht Institute of Linguistics OTS, Utrecht University.
- [12] Evers, M., Nijholt, A. (2000). *Jacob - An animated instruction agent in virtual reality*. In Proc. of the 3rd Int. Conf. on Multimodal Interaction (ICMI 2000).
- [13] Frank, G., Hubal, R. (2002) *An Application of Responsive Virtual Human Technology*. In Proc. of the 24th Interservice/Industry Training, Simulation and Education Conf. 2002.