ECA Perspectives – Requirements, Applications, Technology

Anton Eliëns, Zhisheng Huang, Johan F. Hoorn and Cees Visser
Intelligent Multimedia Group
Vrije Universiteit, Amsterdam, Netherlands
{eliens,huang,jfhoorn,ctv}@cs.vu.nl

ABSTRACT

In the last years we have developed a platform for the realization of embodied (conversational) agents, in a distributed logic programming framework. In this paper we will present an overview of our work, by discussing the requirements that acted as our guidelines for design decisions during development, some of the applications that have served as target demonstrators for developing and testing new functionality, and the (distributed logic programming) technology which we used for the realization of the platform and the implementation of our STEP scripting language.

Although the focus of our paper will primarily be our own DLP+X3D platform, we believe that our discussion along the perspectives of requirements, applications and technology might be more generally worthwhile in establishing the relative merits of the operational use of ECA-technology. At the end of this paper, we will moreover provide some hints of how to approach the experimental validation of the (possible) benefits of embodied conversational agents in user applications.

1. INTRODUCTION

Embodied conversational agents may be regarded as a merge of technology resulting from research in artificial intelligence (agent technology) and computer graphics (humanoid animation). A wellknown example is the Ananova[1] virtual newscaster, that presents the latest news in so-called *video reports*, that are created using "*a unique combination of computer animation, text-to-speech and real-time information systems*".

Other applications of embodied conversational agents include the *Signing Avatar*[2], (a system that allows for translating arbitrary text in sign language for the deaf), talking heads in a variety of e-commerce applications, and the agent-bots in virtual environments such as Active Worlds[3], *blaxxun* Community[4] and Adobe Atmosphere[5].

Clearly, there is such a great diversity of systems in which embodied conversational agents play a role, for which it is not always easy to establish how essential the agent's contribution is, that it is hard to find the common denominator of ECA applications. Our view of what embodied conversational agents should provide is summarized in the five propositions below:

- proposition 1: agents need content
- proposition 2: agents must provide added value
- proposition 3: gestures and (facial) animation must be meaningful
- proposition 4: agents should be naturally embedded in their environment
- proposition 5: the behavior of agents must be understandable

Although, admittedly, a direct translation of these assumptions to system requirements is not feasible, the central idea that the meaningful application of embodied agents requires a relevant context has been a motivating force in developing a declarative platform with powerful computational features.

In addition to the assumptions expressed in the propositions above, we consider two issues as of relevance for establishing the merits of a platform supporting embodied conversational agents, as expressed in the following remarks.

- remark 1: the aesthetics of agents should be centered around *function*, that is form and content
- remark 2: developing agents requires intelligent multimedia technology

Both remarks purport to the need for a strong notion of programmability of both the agents appearance and the (virtual) information world in which the agent lives. In particular, the second remark is a shameless plug for our own technology, as presented in this paper.

[1] http://www.ananova.com
[2] http://www.signingavatar.com
[3] http://www.activeworlds.com
[4] http://www.blaxxun.com
[5] http://www.adobe.com/products/atmosphere

Dagstuhl Seminar Proceedings 04121
Evaluating Embodied Conversational Agents
http://drops.dagstuhl.de/opus/volltexte/2006/461

evaluation criteria There is a wide range of evaluation criteria against which embodied agent application and systems can be validated.

Evidently, the primary criterium against which to evaluate applications that involve embodied conversational agents is whether the application becomes more effective by using such agents. Effective, in terms of communication with the user.

In a more general fashion, however, we may try to establish how well particular systems support features that contribute to the effectiveness of the applications developed with them.

As concerns, for example, the embedding of conversationl agents in VR, we might make a distinction between *presentational VR*, *instructional VR* and *educational VR*. An example of educational VR is described in [1]. No mention of agents was made in the latter reference though. In instructional VR, explaining for example the use of a machine, the appearance of a conversational agent seems to be quite natural. In presentational VR, however, the appearance of such agents might be considered as no more than a gimmick.

Considering the use of agents in applications in general, we must make a distinction between *information agents*, *presentation agents* and *conversational agents*. Although the boundaries between these categories are not clearcut, there seems to be an increasing degree of interactivity with the user.

From a system perspective, we might be interested in what range of agent categories the system covers. Does it provide support for managing information and possibly information retrieval? Another issue in this regard could be whether the system is built around open standards, such as XML and X3D, to allow for the incorporation of a variety of content.

Returning to a user perspective, what seems to matter most is the naturalness of the (conversational) agents. This is determined by the graphical quality, as well as contextual parameters, that is how well the agent is embedded in its environment. Equally important are emotive parameters, that is the mood and style (in gestures and possibly speech) with which the agents manifest themselves. In other words, the properties that determine whether an agent is (really) convincing. Again, translating such features to system requirements is hardly feasible. Yet, it indicates the range of support a platform for embodied conversational agents should offer.

structure The structure of this paper is as follows. In section 2 we will discuss the requirements that acted as guidelines in developing the platform. In section 3, we will briefly describe a selected number of applications that acted as target demonstrators for our technology. Section 4 contains a somewhat more detailed description of the technology on which our platform is built. And finally, in section 5 we will discuss how to approach the validation of agent-based applications and address directions for future research. As a remark, this paper is based on [2], and available online[6].

2. REQUIREMENTS – GUIDELINES FOR SYSTEM DESIGN

Before we started developing our platform, we had a rather strong conception of how to approach the software engineering issues involved, as expressed in [3]. We had a clear interest in distributed web applications, and in particular developing support for 2D and 3D web agents, as reported in [4] and [5]. As our focus turned to VRML-based virtual environments we adopted VRML as our presentation platform of choice, due to the availability of (free) browsers and the convenience of the Java External Authoring Interface.

There is a wide range of presentation platforms for embodied conversational agents. On one end of the spectrum we have digitized video, as for example in the early versions of the Ananova newsreader, On the other end of the spectrum we have rich media real-time 3D platforms, as in the aforementioned virtual environments.

For both type of systems, advanced authoring tools may be used to create content. In addition, however, for the latter type of systems also declarative means of modeling may be used as well as a programmatic interface to create dynamic content.

Evidently, from our background, we have a clear preference for web-based real-time 3D environments with a strong programmatic interface. In other words, this excludes systems that rely on offline rendering, but also systems that rely on the native graphics of a particular machine. Also, we rather program the dynamic behavior than create such behavior using advanced authoring tools. However, our approach allows for easily incorporating content produced by these tools. Our requirements with respect to a platform supporting such environments may be summarized as follows.

- *declarative language* – for agent support
- *multiple threads of control* – for multiple shared objects
- *distributed communication* – networking capabilities

The first release of our DLP+X3D platform was used to create agent-based multi-user virtual environments, deploying a logic-based declarative language providing support for intelligent agents. We have shown in [6] that both autonomous agents and shared objects may be realized using agent-technology with networking capabilities,

scripting behavior Now, although a platform as described above offers powerful computational capabilities, this is clearly not enough to create embodied conversational agents with a rich repertoire of gestures. On top of the DLP+X3D platform, we developed the STEP scripting language for defining gestures and driving the behavior of our humanoid agent avatars [7].

[6]http://www.cs.vu.nl/∼eliens/papers/title-eca.html

The design of the scripting language was motivated by the requirements listed below.

- *convenience* – for non-professional authors
- *compositional semantics* – combining operations
- *re-definability* – for high-level specification of actions
- *parametrization* – for the adaptation of actions
- *interaction* – with a (virtual) environment

STEP is based on dynamic logic [8] and allows for arbitrary abstractions using the primitives and composition operators provided by our logic.

The rationale underlying STEP is rather different than found in approaches relying on animation tools or low level scripting languages. STEP offers the means to model complex gestures on a very high level in a purely declarative fashion, using the primitives and compound operators in more detail described in section 4.2.

3. APPLICATIONS – TARGET DEMONSTRATORS

In the course of developing our platform, we repeatedly shifted focus depending on our interests at the time. We started out with developing a soccer game, that allowed (multiple) users and autonomous agents to interact in a 3D environment. Over time, we had a steady, and very practically motivated, interest in presentational VR, including the use of agent avatars commenting on the material presented. However, the true focus of our work has become our STEP scripting language, which has been extended over time to include both gesture and facial animation, as well as a limited form of text-to-speech synchronization. A particularly nice example of the application of STEP is a student-project featuring a learning domestic servant, that listens to natural language input in a more or less intelligent fashion. Very complex gestural patterns, however, such as for example conducting music, pushed our technology to its limits, as we will discuss more extensively later on.

3.1 Agents in multi-user virtual environments

We chose the soccer game as a demonstrator for multi-user virtual environments because it provided us with a number of challenges, as indicated below [6].

- *multiple (human) users* – may join during the game
- *multiple agents* – to participate in the game (e.g. goalkeeper)
- *reactivity* – players (users and agents) have to react quickly
- *cooperation/competition* – requires intelligent communication
- *dynamic behavior* – sufficiently complex (dynamic) 3D scenes

The soccer player agents each have a simple cognitive loop based on the extended BDI model described in [9], which may be summarized as *sense*, *think*, *act*. [7]

[7] The users' avatars do not have any cognitive model, but act directly on the users' keyboard and mouse input.

Furthermore, to allow multiple users to join the game, taking the place of autonomous agent players if necessary, we designed a special purpose *Agent Communication Language* (ACL) to deal with the communication necessary to keep the world updated. The message format is, schematically: *Action, Type, Parameters*. It allows for a wide range of (distributed) actions, such as the notification of the positions of the player, the movement of the ball, and the arrival and departure of players. It is interesting to note that we may significantly optimize on the communication load if we introduce additional compound messages such as *run & trace ball*, to replace a sequence of primitive messages.

Although both the graphical quality and the efficiency (that is reactive speed) of the soccer game left to be desired, this application demonstrated the viability of our approach in terms of resources need for its development. Despite the complexity of the soccer game, it took only about two months developers time, whereas our previous experience with applications using the blaxxun Community Server[8] would indicate a far more demanding development path using 'low-level' technology.

3.2 Mixed-media presentations with commentators

Desktop VR is an excellent medium for presenting information, for example in class, in particular when rich media or 3D content is involved. As reported in [10], at VU, we have been using *presentational VR* for quite some time, and recently we included dialogs using balloons (and possibly avatars) to display the text commenting on a particular presentation, [11].

Each presentation is organized as a sequence of slides, and dependent on the slides (or level within the slide) a dialog may be selected and displayed. The dialog is encoded in an XML syntax, as an alternation between phrases uttered by the agents. A dialog is activated automatically dependent on the content of a particular slide.

Our presentational VR system supports a number of style parameters to decide for example whether the avatars or persona are visible, where to place the dialogs balloons on the display, as well as the color and transparancy of the balloons. Apart from phrases, we also allow for gestures, taken from the built-in repertoire of the avatars as well as gestures defined using the STEP scripting language. Both phrases and gestures are compiled into DLP code and loaded when the annotated version of the presentation VR is started.

We should note, as we will discuss in section 5.1, that our approach supports a unified presentation of both the material and the agents commenting on the material, unlike the *Agneta & Frida* characters developed in the Persona project[9].

3.3 Grasping the essence of Tai-Chi, domestic servants and conducting music

[8] http://www.blaxxun.com

[9] http://www.sics.se/humle/projects/persona/web

The limited repertoire of gestures of standard VRML avatars, as deployed for example in VRML-based blaxxun Community Server worlds, quite soon became annoying. Apart from suitable cognitive rules, a Web agent should have a convincing repertoire of gestures, and be able to manipulate objects in its environment. So, in consonance with the logic programming environment from which we controlled the cognitive processes governing our agents, we developed the scripting language STEP as a dynamic logic for defining actions and gestures on top of the DLP platform [7].

The first demonstrator developed with STEP technology was a humanoid performing Tai-Chi movements. Tai-Chi exercises are sufficiently complex, requiring parallel motion of the limbs and subtle positioning of the body. In particular, the possibility to define parallel motions independently proved to be effective for concisely modeling the Tai-Chi movements. A subject subsequently addressed was the use of inverse kinematics to allow for pointing to body parts and grapsing objects in the world [12]. Since objects may not always be within reach of an agent, a system of rules was developed to determine whether the agent should move to the object, bow towards the object, or simply grasp it [13]. In some cases, a combination of these alternatives is required to grasp an object from the optimal stance, taking into account the possible presence of obstructing objects.

Later on STEP was extended to support facial animation which, in combination with text-to-speech, culminated in a virtual presenter[10] reading out the text of a powerpoint presentation. Facial animation for speech is triggered by events generated by the text-to-speech engine[11], indicating the beginning and end of phonemes.

an interactive learning agent for domestic services

STEP has been used, in a student project, for creating an adaptive learning agent, that acts as a servant in a domestic environment [14]. The agent reacts to natural language input, which can take the form of requests for information or commands, such as "*where is the book*", "*get me the book*", "*turn on the television*". After processing the natural language input, the agent checks its capabilities to see whether the request can be answered in an appropriate way. Capabilities, in this context, may be regarded as rules relating *Conditions*, *Actions*, and *Effects*. A response can be either an answer in text, a gesture or a sequence of gestures, or an action involving a modification of the actual world in which the agent lives, for example when the television is switched on. Capabilities, which contain in a sense the agents knowledge about the world and the recipes or scripts to perform particular actions, can be extended by providing the agent with information about the world or by explicitly instructing the agent how to perform a particular action.

The domestic servant application, apart from being a non-trivial application of DLP and STEP involving path-following, complex motions and inverse kinematics to grasp objects, has triggered some interesting research issues by itself, in particular natural language processing and learning actions as parametrized STEP code fragments.

reusable gestures for musical conducting

Giving technology out of hands is always an interesting process. Suddenly one is confronted with completely different perspectives. What seems evident may become questionable. And an approach that always worked may suddenly reach its limit and fail. In our cooperation with the (former) Facial Animation Group[12] of CWI we experienced such differences in perspective and encountered limitations of our technology.

In [15], the application of STEP to define gestures for musical conducting is described. Conducting music involves the (timed) motion of both arms and hands, including the fingers. Whereas we conveniently ignored finger movements in our Tai-Chi demo, the conductor demo did not allow for such crudeness. As a result, the number of parallel threads needed to model the quasi-independent motion of all fingers presented a serious stumble-block. Fortunately, we succeeded in significantly reducing the number of independent threads needed, by introducing a new primitive that allows for the parallel motion of body elements (limbs and fingers) using iterated (quaternion-based) interpolation, which can be done using only a single thread. We must note nevertheless that, although an effective solution for this case, in general the combined motion of all joints jeopardizes the high level modeling approach advocated by STEP, which allows for the independent specification of the motion of body parts.

Another issue came up, however, having to do with the synchronization of gestures and sound. On the one hand, there is the problem of precisely synchronizing the gestures among themselves (as when conductiong with two independently moving arms), on the other hand, though, gestures must be synchronized with musical fragments in a very precise manner, on the beat so to say. This becomes even more important when facial animations must be lip-sync with spoken text. As discussed in section 4.3, this is where our technology shows some serious limitations which require significant redesign and the possible adoption of more powerful multimedia technology.

On the positive side, our cooperation with the CWI group demonstrates that STEP allows for the development of libraries of reusable gestures for fairly complex gestural tasks, as conducting music definitely is [16]. In all respects, the work done by the CWI people provides great impetus for the future development of our platform.

4. TECHNOLOGY – DECLARATIVE PROGRAMMING

[10]http://step.intelligent-multimedia.net
[11]Microsoft Speech SDK 5.1
[12]http://www.cwi.nl/projects/FASE/

In section 2 we have hinted at a preference for a programmatic approach to control the functionality of embodied agents. However, this may easily be misunderstood, and must certainly not be taken as to advocate a hackers approach to develop agent-based systems. To the contrary, in line with proposition 5, we adopted a fully declarative approach. Declarative, in this context, means that the code should be understood as a specification of the desired properties of the system, and not (only) by means of an operational interpretation based on some complex execution model. In fact, our platform is unique in that it is uniformly logic-based, both on the level of specifying the cognitive processes of the agents as in the specification of gestures and actions. And to the extent possible, even events within the 3D environment are controlled by a logical specification of sorts.

In this section we will describe the technological basis of our platform, as originally described in [2]. More extensive descriptions can be found in [17]. In section 4.3, however, we will discuss the issues concerning the technological limitations of our platform, as hinted at in [15].

4.1 The DLP+X3D platform

In [6], we have described a platform for virtual environments based on agent technology. In effect, our platform is the result of merging VRML with the distributed logic programming language DLP, using the VRML External Authoring Interface. This approach allows for a clear separation of concerns, modeling 3D content on the one hand and determining the dynamic behavior on the other hand. More recently we have adopted X3D as our 3D format. The VRML profile of X3D[13] is an XML encoding of VRML97.

The language DLP is a distributed object-oriented extension of Prolog [18]. It supports multiple inheritance, non-logical instance variables and multi-threaded objects (to allow for distributed backtracking). Object methods are collections of clauses. Method invocation is dealt with as communication by rendez-vous, for which synchronization conditions may be specified in so-called *accept* statements. The current implementation of DLP is built on top of Java.

To effect an interaction between the 3D content and the behavioral component written in DLP, we need to deal with two issues:

- control points: *get/set* – position, rotation, viewpoint
- *event-handling* – asynchronous accept

We will explain each of these issues separately below. In addition, we will indicate how multi-user environments may be realized with our technology.

control points The control points are actually nodes in the VRML scenegraph that act as handles which may be used to manipulate the scenegraph. In effect, these handles are exactly the nodes that may act as the source or target of event-routing in the 3D scene. As an example, look at the code fragment below, which gives a DLP rule to determine whether a soccer player must shoot:

```
findHowToReact(Agent,Ball,Goal,shooting) :-
   get(Agent,position,sfvec3f(X,Y,Z)),
   get(Ball,position,sfvec3f(Xb,Yb,Zb)),
   get(Goal,position,sfvec3f(Xg,Yg,Zg)),
   distance(sfvec3f(X,Y,Z),sfvec3f(Xb,Yb,Zb),DistB),
   distance(sfvec3f(X,Y,Z),sfvec3f(Xg,Yg,Zg),DistG),
   DistB =< kickableDistance,
   DistG =< kickableGoalDistance.
```

This rule will only succeed when the actual distance of the player to the goal and to the ball satisfies particular conditions. In addition to observing the state of the 3D scene using the *get* predicate, changes to the scene may be effected using the *set* predicate.

event handling Our approach also allows for changes in the scene that are not a direct result of setting attributes from the logic component. Therefore we need some way to intercept events. In the example below, we have specified an observer object that has knowledge of, that is inherits from, an object that contains particular actions.

```
:- object observer : [actions].
var slide = anonymous, level = 0, projector = nil.

observer(X) :-
   projector := X,
   repeat,
     accept( id, level, update, touched),
   fail.

id(V) :-  slide := V.
level(V) :- level := V.
touched(V) :- projector←touched(V).
update(V) :- act(V,slide,level).
:- end_object observer.
```

The constructor sets the non-logical variable *projector* and enters a repeat loop to accept any of the incoming events for respectively *id*, *level*, *update* and *touched*. Each event has a value, that is available as a parameter when the corresponding method is called on the acceptance of the event. To receive events, the *observer* object must be installed as the listener for these particular events.

The events come from the 3D scene. For example, the *touched* event results from mouse clicks on a particular object in the scene. On accepting an event, the corresponding method or clause is activated, resulting in either changing the value of a non-logical instance variable, invoking a method, or delegating the call to another object.

An observer of this kind is used in the system described in section 3.2, to start a comment (dialog) on the occurrence of a particular slide.

[13] http://www.web3d.org/fs_x3d.htm

4.2 STEP – a scripting language for embodied agents

STEP is a scripting language for humanoids based on dynamic logic. The STEP scripting language consists of basic actions, composite operators and interaction operators (to deal with the environment in which the movements and actions take place). See [17]

The basic actions of STEP consist of:

- *move* – `move(Agent,BodyPart,Direction,Duration)`
- *turn* – `turn(Agent,BodyPart,Direction,Duration)`

These basic actions are translated into operations on the control points as specified by the H-Anim 1.1 standard.

As composite operators we provide sequential and parallel composition, as well as *choice* and *repeat*. These composite operators take both basic actions and user-defined actions as parameters.

Each action is defined using the *script*, by specifying an action list containing the (possibly compound) actions of which that particular action consists. As an example, look at the definition of *walking* below.

```
script(walk(Agent), ActionList) :-
ActionList = [
  parallel([turn(Agent,r_shoulder,back_down2,fast),
         turn(Agent,r_hip,front_down2,fast),
         turn(Agent,l_shoulder,front_down2,fast),
         turn(Agent,l_hip,back_down2,fast)]),
  parallel([turn(Agent,l_shoulder,back_down2,fast),
         turn(Agent,l_hip,front_down2,fast),
         turn(Agent,r_shoulder,front_down2,fast),
         turn(Agent,r_hip,back_down2,fast)])
], !.
```

Notice that the *Agent* that is to perform the movement is given as a parameter. (Identifiers starting with a capital act as a logical parameter or variable in Prolog and DLP.)

Interaction operators are needed to conditionally perform actions or to effect changes within the environment by executing some command. Our interaction operators include: *test*, *execution*, *conditional* and *until*.

Potentially, an action may result in many parallel activities. To control the number of threads used for an action, we have created a scheduler that assigns activities to a thread from a thread pool consisting of a fixed number of threads.

XML encoding Since we do not wish to force the average user to learn DLP to be able to define scripts in STEP, we are also developing XSTEP, an XML encoding for STEP. We use *seq* and *par* tags as found in SMIL[14], as well as *gesture* tags with appropriate attributes for speed, direction and body parts involved. See [19]. As an example, look at the XSTEP specification of the *walk* action.

[14]http://www.w3.org/AudioVideo

```
<action type="walk(Agent)">
<seq>
 <par speed="fast">
 <gesture type="turn" actor="Agent" part="r_shoulder"
    dir="back_down2"/>
 ...
 </par>
 <par speed="fast">
 ...
 <gesture type="turn" actor="Agent" part="r_hip"
    dir="back_down2"/>
 </par>
</seq>
</action>
```

Similar as with the specification of dialog phrases, such a specification is translated into the corresponding DLP code, which is loaded with the scene it belongs to. For XSTEP we have developed an XSLT stylesheet, using the Saxon[15] package, that transforms an XSTEP specification into DLP.

Ontologies One particular feature of STEP that should be mentioned is the use of ontologies, to shield the programmer/developer from the low-level numerical specification of actual values for timing and motions in 3D space. Ontologies form an intermediate interpretative layer that assigns actual values to natural language-like temporal specifications such as *fast* and *slow*, and body movement specifications such as *down* and *up*. This intermediate layer is appropriately called an ontology since it characterizes the properties of the world in which the agent lives, taking into account the agents frame of reference.

4.3 On parallelism, synchronization and runtime performance

Complex humanoid gestures are of a highly parallel nature. The STEP scripting language supports a direct way of modelling parallel gestures by offering a parallel construct (*par*), which results in the simultaneous execution of (possibly compound) actions.

To avoid unconstrained thread creation, the STEP engine makes use of a thread pool, containing a fixed number of threads, from which threads are allocated to actions. Once the action is finished, the thread is put back in the pool.

This approach works well for most examples. However when many threads are needed, as in the conductor example (which requires approximately 60 threads), problems may occur, in particular when there are may background jobs.

When using parallelism three types of potential problems may be distingushed:

- synchronisation among gestures,
- (reliable) timing of gestures, and
- synchronisation of gestures with external events, e;g.audio.

Synchronisation among gestures requires that gestures that are meant to be executed simultaneously do indeed start and

[15]http://saxon.sourceforge.com

end at the same time. Unwanted delays however may occur, for example, when there are nested parallel constructs, due to the processing needed to unravel compound gestures. One remedy here is to optimize the Prolog engine, which is an ongoing effort. Another, equally obvious, remedy is to eliminate nested parallel constructs by some sort of flattening. These solutions require no or only a minor change to the STEP engine implementation. In effect, we have introduced a *par_turn* construct that allows to declare the parallel rotation of an arbitrary number of joints. However, such a construct, when applied exclusively, limits modelling to defining what can best be called successive 'frames' of a motion, which runs counter the approach advocated by STEP, namely defining gestures in a natural way, as concurrent motions of the human body.

To solve the problem of reliable timing in a more fundamental way would require not only a modification of the STEP engine, but also a rather different implementation of the DLP threads supporting the parallelism in STEP. Currently, the implementation only allows for *best effort* parallelism and does not provide the means for *deadline scheduling*.

Synchronisation with external events is possible, to the extent that such external events are supported by the VRML/X3D environment. That does at this stage, unfortunately, not include the synchronisation with audio events, unless an audio track is broken up in separate pieces.

Summarizing, as concerns the use of parallelism in modeling gestures we run into a number of problems which are either caused by exhausting the resources of the machine, inefficiencies in the execution of Prolog code, or limitations of the thread model supported by DLP. Partial remedies include, respectively, reducing the number of threads needed, eliminating nested parallel constructs, optimizing the Prolog execution engine, and a push-down of critical code of the STEP engine to the Java level. These measures can be taken without affecting the STEP engine or the underlying DLP thread model.

Nevertheless, there remains still some concern whether the inherent inefficiencies of the Java platform and the VRML Java External Authoring Interface can be overcome in a satisfying degree. Therefore, we have recently started to develop a version of DLP and STEP for the Microsoft .NET platform, which we believe that in combination with Direct3D based implementations of 3D graphic models allow for better control of the issues that we identified as problematic, in particular thread control, synchronization and interaction with a dynamic environment. As a preliminary result, we can mention that our Prolog implementation in the .NET C# language already runs ten times faster than our Java implementation on a standard set of Prolog programs.

5. EVALUATION – VALIDATING AGENT-BASED APPLICATIONS

Propositions 1 and 2 in the introductory section emphasize the need for meaningful content. In particular, as indicated in propositions 3 and 4, agent behavior must be meaningful and, in the perception of the user, be naturally embedded in its environment. In this section we comment briefly on our use of agents in presentational VR, as described in section 3.2. More importantly, however, we will outline how to approach the validation of agent-based systems from a user perspective, building on a theory of agents as fictitious characters. Finally, we will indicate some future research issues, including scenario(s) for the experimental evaluation of (the added value provided by) embodied conversational agents.

5.1 Narrative structure and presentations

Dialogs, as introduced in section 3.2, have a well-defined temporal structure, with alternating turns for the two (virtual) speakers. Mixed media, that is the combination of dialogs with rich media and virtual environments, may endanger the narrative structure of a presentation, due to conflicts between the narrative structures of the distinct media items. Slides are sequentially organized, and each slide may have levels that are displayed in a sequential fashion. The narrative structure of digital video may be arbitrarily complex, and may make full use of cinematographic rethorics. Navigation and interaction in virtual environments may be seen as a weak narrative structure, which may however be strengthened by guided tours or viewpoint transformations, taking the user to a variety of viewpoints in a controlled manner.

Our approach is clearly reminiscent to the notorious *Agneta & Frida* characters developed in the Persona[16] project. The Persona project aims at: investigating a new approach to navigation through information spaces, based on a personalised and social navigational paradigm [20]. The novel idea pursued in this project is to have agents (Agneta and Frieda) that are not helpful, but instead just give comments, sometimes with humor, but sometimes ironic or even sarcastic comments on the user's activities, in particular navigating an information space or (plain) web browsing.

In contrast with the Personas project, our (dialog) comments are part of a presentation which of itself has a definite narrative structure, in opposition to the 'random' navigation that occurs by browsing 'information spaces'. As a consequence, our comments may be designed taking the expected reaction of the audience into consideration. An interesting question is whether comments should be *consonant* with the information presented (drawing attention to particular aspects) or *dissonant* (as with ironic or sarcastic comments).

The characters and dialog text may be used to enliven the material. In this way, the students' engagement with the material may be increased [21]. Clearly, there is a tension between engagement and immersion. Immersion, understood as the absorption within a familiar narrative scheme (in our case the lecturer's presentation), may be disrupted by the presence of (possibly annoying) comments, whereas the same comments may lead the attention back to the material, or provide a foothold for affective reactions to the material [22]. Also, the audience might start to anticipate the occurrence of a dialog and possibly identify themselves with one of the characters.

In one of her talks, Kristina Höök observed that some users

[16]http://www.sics.se/humle/projects/persona/web

get really fed up with the comments delivered by Agneta and Frieda. Nevertheless, it also appeared that annoyance and irritation increased the emotional involvement with the task. For our presentations, we may ask how *embodied agents* may help in increasing the emotional involvement of the audience or, phrased differently, how dialogs may lead to emotional enhancement of the material [23]. An important difference with the Personas project is that our platform supports the actual merge of dialogs and the humanoid characters that deliver them in a unified presentation format, that is a rich media 3D graphics format based on X3D/VRML. As a consequence, the tension between immersion and engagement may be partially resolved, since the characters delivering the dialog may be placed in their 'natural' context, that is a virtual environment.

5.2 Embodied agents as fictional characters

In [24] the following observation was made:

> Observation: VR is fiction and fiction is as old as humanity.

This observation which is a different theoretical context (re)affirmed in [25], leads to the following, rather obvious, corollary:

> Corollary: subjective feeling of involvement with fictional situation is achieved by intentional suspension of disbelief.

The corollary is equally valid for embodied conversational agents. In [24] and [26], a theory is developed (PEFiC), concerning *Perceiving and Experiencing Fictional Characters*, that may serve as the basis for the experimental evaluation of user responses to embodied agents. In summary, PEFiC distinguishes between three phases, *encoding*, *comparison* and *response*, in analyzing the user's behaviour towards an agent. Encoding involves positioning the agent (or fictitious character) on the dimensions of *ethics* (good vs bad), *aesthetics* (beauty vs ugliness) and *epistemics* (realistic vs unrealistic). Comparison entails establishing personal relevance and valence towards the agent. Response, finally, determines the tendency to approach or avoid the character, in other words involvement versus distance.

Empirical support for the PEFiC-model was found using structured questionnaires with various character types and focusing on different aspects of the model. The characters against which the PEFiC-model was tested ranged from animated Web agents, such as Bonzi Buddy [27], to Hollywood feature film characters, [26], to pictorial representations of world leaders [28]. Underlying this range was a continuum of realism in depiction. Bonzi Buddy was the most unrealistic, cartoon-like, representation of a living creature, followed by fantasy figures played by real actors, such as Dracula and Superman, in the film studies. The film studies also investigated more realistic characters (e.g., Gandhi), while the study of world leaders explored the most realistic representations, that is, newspaper pictures of, for example, president Bush and Bin Laden.

5.3 Future research directions

Looking back, we succeeded in creating a platform for embodied agents that may potentially satisfy propositions 1-5, uniformly based on (some variant of) logic, suitable for creating web-based 3D agent applications. Looking to the future, there is still a lot of work to do. Technical work, to solve the problems we encountered with respect to synchronization and parallelism, as well as work to enhance the support for multimedia. At this stage it is, however, also important to define proper evaluation scenarios and to set new challenges against which to validate the future technological enhancements of our platform.

validation scenario(s) In section 5.1, we already (implicitly) identified two interaction paradigms, namely *pure navigation* and *guided tours* or presentations that are based on some narrative structure. Both paradigms can be augmented using embodied conversational agents, which in the case of navigation might merely give directions and, in the case of guided tours, may explain what is going on, possibly offering the user a choice of continuations. In cooperation with the Faculty of Social Sciences, we have submitted a research funding proposal to undertake such evaluation studies, based on the PEFiC theory outlined in section 5.2. In this proposal [29], we focused in particular on the relation between the type of agent (character) and the material presented (context), to determine how this relation affects the valence of the user towards the agent, resulting in the degree of distance or involvement experienced.

case study Apart from the evaluation studies mentioned above, we also plan to target a new case study, that brings many challenges with it. The project is to be done in cooperation with the *Dutch Cultural Heritage Institute* (ICN[17]) in the context of the *International Network for the Conservation of Contemporary Art* (INCCA[18]). Briefly, the idea is to develop so-called *digital dossiers* for individual artworks, allowing professionals to deal with the information involved in an integrated, highly interactive fashion. The following project assignment may serve as a characterization of the notion of digital dossier:

> *Create a VR that realizes a digital dossier for a work of a particular artist. A digital dossier represents the information that is available for a particular work of art, or a collection of works, of a particular artist. The digital dossier should be multimedia-enhanced, that is include photographs, audio and other multimedia material in a compelling manner.*

Note that *dossier* is an existing english word, which according to the *Webster New World Dictionary* has the following meaning

[17] http://www.icn.nl
[18] http://www.incca.org

- dossier (dos-si-er) [Fr ¡ dos (back); so named because labeled on the back] a collection of documents concerning a particular person or matter

It is closely related to the notion of archive, but there is a different focus:

- archive [...] 1) a place where public records are kept ... 2) the records, material itself ...

In relation to the evaluation studies mentioned before, we must investigate what role embodied agents may play in presenting such digital dossiers to the user, and how the agents should appear, that is what services are offered and what the actual presence of the character must be.

6. CONCLUSIONS

In this paper we presented a platform for embodied conversational agents, by reflecting on the motivations underlying its development. We characterized the functionality of the platform by discussing a selected number of applications, illustrating its features, and we included a discussion of its potential limitations. A brief description of our uniformly logic-based technology was given and we concluded by sketching how to validate the platform against actual user experience. Nevertheless, despite the relative maturity of the platform, we see many challenges for its future enhancement needed for the realization of new target demonstrators.

7. REFERENCES

[1] Johnson A., Moher T., Cho Y-J., Lin Y.J., Haas D., and Kim J. (2002), Augmenting Elementary School Education with VR, IEEE Computer Graphics and Applications, March/April

[2] Eliëns A., Huang Z., and 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 17/7/2002

[3] Eliëns A. (2000), *Principles of Object-Oriented Software Development*, Addison-Wesley Longman, 2nd edn.

[4] Visser C. and Eliëns A. (2000), A High-Level Symbolic Language for Distributed Web Programming. Internet Computing 2000, June 26-29, Las Vegas

[5] Huang Z., Eliëns A., van Ballegooij A., De Bra P. (2000), A Taxonomy of Web Agents, IEEE Proceedings of the First International Workshop on Web Agent Systems and Applications (WASA '2000), 2000.

[6] Huang Z., Eliëns A., Visser C. (2002), 3D Agent-based Virtual Communities. In: Proc. Int. Web3D Symposium, Wagner W. and Beitler M.(eds), ACM Press, pp. 137-144

[7] Huang Z., Eliëns A., Visser C. (2002b), STEP – a scripting language for Embodied Agents, PRICAI-02 Workshop – Lifelike Animated Agents: Tools, Affective Functions, and Applications, Tokyo, 19/8/2002

[8] Harel D. (1984), Dynamic Logic. In: Handbook of Philosophical Logic, Vol. II, D. Reidel Publishing Company, 1984, pp. 497-604

[9] Huang Z., Eliëns A., Visser C. (2001), Programmability of Intelligent Agent Avatars, Proceedings of the Agent'01 Workshop on Embodied Agents, June 2001, Montreal, Canada

[10] Eliëns A., Dormann C., Huang Z. and Visser C. (2003), A framework for mixed media – emotive dialogs, rich media and virtual environments, Proc. TIDSE03, 1st Int. Conf. on Technologies for Interactive Digital Storytelling and Entertainment, Göobel S. Braun N.,n Spierling U., Dechau J. and Diener H. (eds¿), Fraunhofer IRB Verlag, Darmstadt Germany, March 24-26, 2003

[11] Eliëns A., Huang Z., Visser C. (2002), Presentational VR – *What is the secret of the slides?*, in preparation

[12] Huang, Z., Eliëns, A., and Visser, C. (2003c), *STEP: a Scripting Language for Embodied Agents*, in: Helmut Prendinger and Mitsuru Ishizuka (eds.), Life-like Characters, Tools, Affective Functions and Applications, Springer-Verlag, (to appear).

[13] Huang, Z., Eliëns, A., and Visser, C. (2003d), "Is it within my reach?" – an agents perspective , Proc. Intelligent Virtual Agents 2003, Irsee, September 15-17, 2003, J.G. Carbonell and J.Siekmann (eds.), LNAI 2792, Springer, pp. 150-158

[14] Hildebrand M., Eliëns A., Huang Z. and Visser C. (2003), Interactive Agents Learning their Environment, Proc. Intelligent Virtual Agents 2003, Irsee, September 15-17, 2003 J.G. Carbonell and J.Siekmann (eds.), LNAI 2792, Springer, pp. 13-17

[15] Ruttkay Z., Huang Z. and Eliëns A. (2003a), The Conductor: Gestures for Embodied Agents with Logic Programming, Joint Annual ERCIM/CoLogNet Workshop on Constraint and Logic Programming, Budapest, Hungary, 30 June - 2 July, 2003

[16] Ruttkay M., Huang Z. and Eliëns A. (2003b), Reusable gestures for interactive web agents, Proc. Intelligent Virtual Agents 2003, Irsee, September 15-17, 2003 J.G. Carbonell and J.Siekmann (eds.), LNAI 2792, Springer, pp. 80-87

[17] Huang Z., Eliëns A., Visser C. (2003a), Implementation of a scripting language for VRML/X3D-based embodied agents, Proc. Web3D 2003 Symposium, Saint Malo France, S. Spencer (ed.) ACM Press, pp. 91-100

[18] Eliëns A. (1992), DLP – A language for Distributed Logic Programming, Wiley

[19] Huang Z., Eliëns A., Visser C. (2003b), XSTEP: A Markup Language for Embodied Agents, Proc. CASA03, The 16th Int. Conf. on Computer Animation and Social Agents

[20] Munro A., Höök, K. and Benyon D.R. (1999), Footprints in the Snow. In: Social Navigation of Information Space, Springer

[21] Morgan (2000), Theory and models for creating engaging and immersive ecommerce websites, Proc. of the 2000 ACM SIGCPR Conf. on Computer Personnel Research, April 2000, pp. 77-85

[22] Dijkstra K., Zwaan R. Graesser A.and J. Magliano (1994), Character and reader emotions in literary texts, Poetics 23, pp. 139-157

[23] Astleneir H. (2000), Designing emotionally sound instruction: the FEASP approach. Instructional Science 28, pp. 169-198

[24] Konijn E.A. and Hoorn J.F. (2003), Perceiving and Experiencing Fictional Characters, an integrative account, Japanese Psychological Research, 45, 4, pp. 250-268

[25] Bolter J.D and Grusin R. (2000), *Remediation – Understanding New Media*, MIT Press

[26] Konijn E.A. and Hoorn J.F. (2004), Some like it bad. Testing a model for perceiving and experiencing fictional characters, Media Psychology Vol. X, pp. XX

[27] Hoorn, J. F. (2003), The Role of Social Norm in User-engagement and Appreciation of the Web Interface Agent Bonzi Buddy, Tech. Rep. VU Amsterdam

[28] Konijn, E. A., and Hoorn, J. F. (2003), Pushing the Ethic, Aesthetic, and Epistemic Borders in Meeting Mediated People. Int. Communication Association, Theme Sessions: Local and Everyday Praxis in Virtual Borderlands, San Diego, California, May 23-27, 2003.

[29] Kleinnijenhuis J., van der Veer G., Konijn A. and Hoorn J.F. (2003), Designing the user experience of Web as receivers of mass communication, VUBIS Funding proposal, Faculty of Sciences and Faculty of Social Sciences, Free University Amsterdam