VIXEE an Innovative Communication Infrastructure for Virtual Institutions

(Extended Abstract)

Tomas Trescak
Artificial Intelligence Research
Institute, IIIA, CSIC
Barcelona, Spain
ttrescak@iiia.csic.es

Marc Esteva
Artificial Intelligence Research
Institute, IIIA, CSIC
Barcelona, Spain
marc@iiia.csic.es

Inmaculada Rodriguez
Applied Mathematics
Deprtment, UB
Barcelona, Spain
inma@maia.ub.es

ABSTRACT

Virtual Institutions (VI) provide many interesting possibilities for social virtual environments, collaborative spaces and simulation environments. VIs combine Electronic Institutions and 3D Virtual Worlds. While Electronic Institutions are used to establish the regulations which structure participants interactions, Virtual Worlds are used to facilitate human participation. In this paper we propose Virtual Institution Execution Environment (VIXEE) as an innovative communication infrastructure for Virtual Institutions. Main features of the infrastructure are i) the causal connection between Virtual World and Electronic Institutions layers, ii) the automatic generation and update of VIs 3D visualization and iii) the simultaneous participation of users from different Virtual World platforms.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems; H.5.1 [**Multimedia Information Systems**]: Artificial, augmented, and virtual realities

General Terms

Human Factors, Management, Design

Keywords

Virtual Institutions, 3D Virtual Worlds

1. INTRODUCTION

Nowadays there is an increasing demand of applications supporting the participation of humans and software agents, which may engage in different activities to achieve their common or individual goals. Internet based and distributed software technologies, such as virtual worlds (VW) and multiagent systems (MAS), may support the engineering of this type of applications. Specifically, Virtual Institutions [1] (VI) combine Virtual Worlds and Electronic Institutions [2] (EI) to support the engineering of this type of applications.

Cite as: VIXEE an innovative communication infrastructure for Virtual Institutions (Extended Abstract), Tomas Trescak, Marc Esteva and Inmaculada Rodriguez, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1131-1132.
Copyright © 2011, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

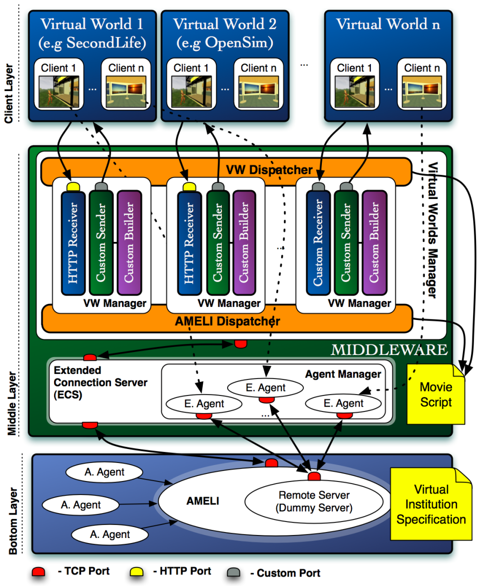

Figure 1: Architecture of the Virtual Institution Execution Environment

In this paper we propose VIXEE as an innovative Virtual Execution Environment which adds important extensions to previous Virtual Institution infrastructures. These extensions address generic and dynamic features. That is, our framework is able to allocate at run-time participants from different VW worlds and it can modify on the fly the 3D content of the Virtual Institution currently executing.

2. VIXEE ARCHITECTURE

Virtual Institution Execution Environment (VIXEE) has a 3-layered architecture (see Figure 1). Uses of VIXEE can be found in participatory simulation or any system where we need to mediate human to human or human to agent interactions.

2.1 Bottom Layer

The bottom layer is formed by AMELI the electronic institutions infrastructure that mediates agents' interactions while enforcing the nstitutional rules. AMELI is a general purpose infrastructure, as it can interpret any institution specification generated by ISLANDER, the EIs specification editor. Therefore it can be regarded as domain-independent. It is implemented in JAVA and uses two TCP ports for communication with the middleware.

2.2 Top Layer

The top layer consists of several 3D virtual worlds (universes). Each of the virtual worlds can be implemented in different programming language using different visualization technologies. The usual parts of the 3D virtual world is a 3D client and a 3D server. Such server communicates with the middleware using a protocol (e.g. TCP, HTTP). Our middleware implements a *multi-verse communication* mechanism that allows users from different virtual worlds to communicate between each other. Moreover, VIXEE uses our Virtual World Grammar (VWG) mechanism and its implementation in the Virtual World Builder Toolkit (VWBT) [5] to dynamically manipulate the 3D virtual world content. The toolkit automatically generates a 3D model loading a specification of a VI and using a VWG definition.

2.3 Middleware

The middleware causally connects the top and the bottom layer. Layers are causally connected because whenever one of them changes, the other one changes in order to maintain a consistent state [3]. We divide the middleware between the Extended Connection Server (ECS) and the Virtual World Manager (VWM).

2.3.1 Extended Connection Server (ECS)

ECS mediates all the communication with AMELI, and is an extended version of the original Generic Connection Server developed for the Itchy-Feet project [4]. The most important extensions are: support for multiple 3D virtual worlds; modified startup sequence, that allows to react on early EI events; and connection fail-safe mechanisms. An important part of ECS is the Agent Manager. For each avatar, participating in some 3D virtual world, an Agent Manager creates an external agent (E. Agent in Figure 1) in the middleware representing this avatar within the institution. Thus, when the avatar tries to perform an action which requires institutional verification this agent is used to send the corresponding message to AMELI. Hence, AMELI, perceives all participants as software agents. ECS uses three TCP ports, one to communicate with the VWM, the second one to listen for AMELI events and the third one is used by the Agent Manager to send external agents events to AMELI.

2.3.2 Virtual Worlds Manager (VWM)

VWM mediates all communication between 3D virtual worlds and ECS and dynamically manipulate the 3D representation of all connected virtual worlds. Virtual Worlds Manager consists of a set of Virtual World Managers, one for each connected virtual world (see Figure 1). Each Virtual World Manager consists of a triplet: a *receiver*, a *sender* and a *builder*. Each triplet is registered to a *VW Dispatcher*, responsible for mediation of virtual world events and an *AMELI Dispatcher* responsible for AMELI events received from ECS. Both dispatchers use our proposed *movie script* mechanism (see section 2.4) to select which action to perform depending on the context of an event.

2.4 Movie Script

To define the mapping between virtual world events and ECS protocol messages, and vice versa between ECS protocol messages and virtual world actions we propose a *movie script* mechanism. This mechanism supports the domain independence and facilitates simple and consistent definition of 3D virtual world behavior. Like a regular movie script it contains script lines. Each line holds a definition of specific context upon which a defined action will be executed. Formally we define a *movie script line* as a function which maps an event to a corresponding action $script_n : w \times i \times ag \times l \times c \to a$ where: (i) w is the layer where the event has taken place, that is either AMELI, or the identifier of a specific virtual world (ii) i is the electronic institution for which the event applies (iii) ag is the agent performing the event (iv) l is the location of the event, that is either some transition or scene (v) c is a *event descriptor*, that is a tuple: $c \in \{[n, 2^p]\}$ where n is the name of the message and 2^p is a list of message parameters, or message context (vi) a is the action which must be performed in response to the event occurrence. Action type differs depending on the originator of the event. If the event originator was a 3D virtual world, the action is a message sent to the ECS. If the event originator is AMELI, the action is usually a *sender* method that updates the virtual world visualisation.

3. CONCLUSIONS

We have presented a VIXEE an automated virtual institution execution environment that introduces many interesting features in the current line of research. First, the multi-verse communication supports the participation in the virtual institution of users from different virtual worlds. Second, our VWBT allows the dynamic manipulation of the virtual world content. Last, due to its dynamic and generic nature it is architecturally neutral allowing its use in multiple domains. VIXEE is also multi-platform solution that allows to run in any operating system supporting Java and Mono framework.

Acknowledgments

This work is partially funded by projects EVE (TIN2009-14702-C02-01 / TIN2009-14702-C02-02), AT (CONSOLIDER CSD2007-0022) and by the Generalitat de Catalunya under the grant 2005-SGR-00093. M.Esteva enjoys a Ramon y Cajal contract from the Spanish Government.

4. REFERENCES

[1] A. Bogdanovych. *Virtual Institutions*. PhD thesis, University of Technology, Sydney, Australia, 2007.

[2] M. Esteva. *Electronic institutions. from specification to development*. PhD thesis, Universitat Politecnica de Catalunya, 2003.

[3] P. Maes and D. Nardi, editors. *Meta-Level Architectures and Reflection*. Elsevier Science Inc., New York, NY, USA, 1988.

[4] I. Seidel. *Engineering 3D Virtual World Applications Design, Realization and Evaluation of a 3D e-Tourism Environment*. PhD thesis, Technischen Universitat Wien Fakultat fur Informatik, 2010.

[5] T. Trescak, M. Esteva, and I. Rodriguez. A virtual world grammar for automatic generation of virtual worlds. *The Visual Computer*, 26:521–531, 04/2010 2010.