2

Controlling and Assisting Activities in Social Virtual Worlds

I. Rodriguez, A. Puig and M. Esteva
Applied Mathematics Department. University of Barcelona, Spain
Artificial Intelligence Research Institute. Barcelona, Spain

1. Introduction

Since its beginning, web technology has advanced from a text-based to a visual-based interaction. This evolution has been facilitated by both high speed internet connections and PC's graphical power. Virtual world (VW) technology began as standalone applications (e.g.. virtual simulations) but soon evolved into web-based applications. Nowadays, home users for entertainment and wide-spread enterprises or institutions for business can exploit virtual worlds to develop remote activities between friends, employees, clients, teachers or students (Sherman, 2002). Then, virtual worlds have clear applications in e-governance, e-learning and e-commerce, and therefore it is mandatory to study mechanisms ensuring the assistance and the control of activities taking place in these applications.

We focus on social virtual worlds populated by participants that act in order to achieve common and individual goals (Bartle, 2003). Due to the type of activities taking place in a Social Virtual World (SVW), the virtual environment should be prepared to be a dynamic space where participants are informed about activities' evolution and where norms are used to organize participants' actions, to define actions' consequences and to prevent undesired participants behaviours. We rely on electronic institutions (Esteva, 2003) to set up participants' valid interactions and on virtual objects, named intelligent objects, populating the virtual world, to enforce norms and to give assistance to participants (Rodriguez, 2008). This work exploits the Virtual Institution (VI) concept which is a combination of both multiagent and virtual world technologies (Bogdanovych, 2007).

We have designed a general framework of object behaviour control tied with an IA based external module and prepared to be exploited by several virtual world platforms. This is done creating a specific module to capture participant interactions on objects populating the virtual world and connecting this module with an external and generic one in charge of deciding what should be the virtual object action. Decision depends on an organization-based multiagent system (MAS) which, as said before, establishes the valid interactions participants may have and the consequences of those interactions. Our main objectives are:

- Establish participants' roles, activities and norms by means of a multiagent system named electronic institution. Participants can be both software agents and humans.

- Use of intelligent virtual objects with an external module named *iObject* manager i) to inform participants about activities evolution and ii) to decide whether participants comply established norms.
- Design an object behaviour control scheme applicable to different VW (Virtual World) platforms.
- Exploit the virtual nature of the spaces, and the objects populating these spaces, allowing to represent things impossible for their real world counterparts.

The dynamic feature of current VW platforms only rely on users who are free to dynamically change aspects of the virtual world by means of built-in tools and scripting features. Our proposal is to extend the ability of a VW to dynamically change itself and exploit the virtualness of the space supporting the presentation of information, which would be impossible to do in the real world, and so provide a better support to participants on their activities.

This chapter is structured as follows. Section 2 presents related work in areas related with norms enforcement in virtual worlds and multiagent systems combined with virtual environments. Section 3 describes how our system models activities taking place in normative virtual worlds and uses intelligent objects to guide and control the user during the activities. Section 4 presents the developed intelligent objects framework and finally section 5 presents conclusions and future work.

2. Related work

2.1 Norms in web based communities

Most of well-known virtual communities -such as Second Life, Active Worlds, Entropia and others- require participants to agree to the company's terms of service in the signing up process (Linden, 2008). Participants should understand the terms and conditions to which they are agreeing as a member of that community. Most people don't read or are otherwise immune due to the lack of real consequences. There are some types of incorrect behaviours that we think can be addressed programatically, that is contemplated in the design of the VW platform and ensured at deployment time.

We propose to use intelligent objects (iObjects) as elements helping users to comply norms and if it is necessary to prevent forbidden actions. For example, to block entry to people who is less than 18 years old in a special virtual room. WonderDAC is an extension module developed for Wonderland that allows to show or hide parts of a VW depending on the user and group profile (Wright and Madey, 2008). In contrast to WonderDAC, developed to control discretionary access basing on users and group permissions, our approach is more general allowing, for example, the control of access to spaces based on the historic of user activities. For example, a norm establishes that a participant can not enter to the projection room unless he has bought a ticket for that room and session.

Part of our inspiration for a general interaction approach for objects populating a social virtual world comes from the smart objects proposal (Kallmann et al., 2000) (Abaci et al, 2005) and the posterior work done by Jorinssen (Jorissen et al., 2004) (Jorissen et al., 2005). Nevertheless, our approach is different to those because they worked with their own virtual environments named ACE (Agent Common Environment) and ALVIC (Architecture for Large-Scale Virtual Interactive communities), respectively. In this way, their object interaction approach is general in the sense it is independent of the final application but can not get out of their concrete virtual platform (they have their own scripting language and engine). Our interaction framework for control and assist activities in SVW has been designed to be applied to different VW platforms such as Wonderland and Second Life. In this way, rendering and event capture continue being controlled in the concrete VW platform but the behaviour decision is managed in an external and generic manager connected with an AI based module, i.e. Electronic Institution.

Virtual worlds can be seen as singular information spaces where the virtual nature of the 3D space (e.g.. floor) and the furniture (e.g.. noticeboard) can be exploited in a special manner not possible for their real counterparts. For example, in the real world it is not possible to dynamically change tiles colour in a floor to represent an agree/disagree position of participants in a discussion. This has been done in a recent work (Harry & Donath, 2008). We aim to incorporate an added value to virtual objects allowing to give valuable information to participants. As an example in section 4.1, a door is visualized either green or red depending on the user trying to pass through. Accessibility issues can also be addressed in these information spaces, for example a noticeboard object adapts letter size depending either on user profile and on the distance between the user and the panel. Exploiting these native properties of virtual objects, we create rich and expressive social spaces.

We extend the dynamic conception of current VW platforms in which users are free to dynamically change aspects of the virtual world by means of built-in tools and scripting behaviours (Friedman et al., 2007) (Sun, 2008). Part of the unexplored feature of virtual spaces is their ability to be adapted in architectural terms. Our proposal is to extend the ability of a VW to dynamically change itself and exploit the virtualness of the space supporting the presentation of information, which would be impossible to do in the real world, and so provide a better support to participants on their activities.

2.2 Combining multiagent systems and virtual environments

A system that incorporated intelligent agents within virtual environments was mVITAL (multi-agent VITAL) which allowed the definition of agent societies so that intelligent agents could communicate through simple speech acts, co-operate and help each other to achieve goals (Vosinakis et al., 1999) (Anastassakis et al. 2001a) (Anastassakis et al. 2001b) . The mVITAL viewer allowed human supervisors to observe the activity inside the environment. We propose to allow the user not only to supervise but to control his avatar and communicate with a regulated multi-agent system in order to test whether his actions are allowed. We have used the so-called iObjects in order to provide facilities for avatar-object interaction and the visualization of the social virtual world execution context. A detail description of iObjects integration at MAS level by means of an Interaction Language can be found in (Rodriguez et al. 2007).

Several researches integrated BDI (Belief, Desire and Intention) agents within virtual worlds. Torres et al. developed an interface that allowed a BDI-based agent reasoning system to be used for guiding the behaviour of articulated characters in a virtual environment (Torres et al., 2003). ACE (Agent Common Environment) was designed for virtual human agent simulations. It provided pre-built commands to perceive and actuate meanwhile the reasoning processing is defined by means of a collection of external modules (i.e. python scripts)(Kallmann et al, 1998), (Kallmann et al, 2000). Virtual agents were used to enhance Customer Relationship Management (CRM). eGain's virtual assistants interact in plain English over the Web with online users (Osterfelt, 2001). They combined 3D graphical representations and artificial intelligence to assist customers to locate information or place orders. Our system provides assistance to the participant also by means of 3D graphical representations (i.e. iObjects). An iObject allows the user to be aware of current execution state (e.g.. data visualized on an intelligent noticeboard), enforcing norms (e.g. let to pass through a door depending on user previous activities) and adapting object's features depending on user profile (e.g.. adapts the font's size of a noticeboard depending on user's visual capacity).

Guyot and Honiden's approach merged multiagent systems and role-playing games (MAS/RPG) (Guyot, 2006). They compared agent-based participatory simulations and the MAS/RPG approach and explained the advantages of their approach: "actions and interactions can be registered and used for learning purposes, the gap between the agent model and the participants can be decreased and the user interface with an assistant agent may improve the understanding of the model by the participants". Our system, exploiting iObjects in the context of social virtual worlds, aims to work along those advantages too.

Another research conceives the organisation infrastructure of a multiagent system in terms of agents and artifacts (Kitio et al.. 2007). They distinguish between organizational artifacts, which provide organization's functional aspects, and organizational agents, which provide decision aspects of organizations management. Artifacts and iObjects, although both arise with a similar objective, that is, to model "entities" used to develop activities in the institutions, they are situated in different levels of abstraction. Artifacts facilitate agent activities at a organizational MAS level and iObjects facilitate user interactions at 3D world level.

3. Modelling activities in a social virtual world

3.1 Our approach: a hybrid system with software agents and humans in 3D virtual worlds

Conventional virtual communities are populated by avatars representing human participants connected to the virtual world. We focus on a hybrid approach due to the heterogeneous nature of participants as they can be software agents and humans. Our system is based on Bogdanovych approach which utilizes this hybrid nature of participants in the so named Virtual Institutions (Bogdanovych, 2007) (Bogdanovych et al., 2008). Despite of the hybrid system complexity, it has advantatges as the human participant controls its avatar in a concrete activity happening in a concrete 3D scene (e.g.. asking for information in an e-goverment information office) but it could launch an agent software, that in his behalf, should perform another activity in another 3D space (e.g.. filling an administrative form in the tax office). Then, it is needed to set up roles, activities, norms and obligations of participants in the social virtual world as described in the next section.

3.2 An organization based Multiagent System

We are interested in social virtual worlds which emulate activities in a real institution. For the specification of the institutional rules, we use electronic institutions (Esteva, 2003), a well-known MAS methodology. The institutional rules establish the valid interactions agents may have and the consequences of those interactions. Specifically, institution designers should define the following components (the formalization of these components can be found in (Arcos et al. 2005)):

- Dialogical framework. It establishes the common ontology and communication language to allow agents to exchange knowledge and understand each other.
- Social structure. It establishes the roles that the agents may play within the institution and the relationships among them. Each role defines a pattern of behaviour within the institution.
- Scenes. Each scene defines an interaction protocol among a set of roles. The protocol, specified by a finite state machine (FSM), establishes the valid interactions that agents may have. The nodes of the FSM represent the different conversation states, while the arcs are labelled with messages of the communication language or timeouts. A scene specification also defines at which states agents, depending on their role, can join or leave.
- Performative structure. It defines the role flow policy among scenes, that is, how agents depending on their role can move among the different scenes. The performative structure is specified as a graph. Graph's nodes are scenes and transitions, and arcs are labelled with the roles that can progress through them. Transitions are a kind of routers that permit to express synchronisation, parallelisation and choice points for agents moving between scenes.
- Norms. They capture the consequences of agents' actions within the institution. Such consequences are modeled as commitments (obligations) that agents acquire as the result of their actions. It is worth mentioning that the specification also includes the definition of the information model that the institution uses to keep the state of participants and activities going on at run time. For instance, an auction house may keep for each buyer her current credit and the list of purchased goods. This is specified as a list of attributes (or properties) associated to some of the previous elements. The specification of the institutional rules is supported by ISLANDER, the electronic institutions specification tool (Arcos et al. 2005).

At design time, the specification focusses on macro-level (rules) aspects of agents not in their micro-level (players) aspects. No assumptions are made at specification time about the internal architecture of participating agents. Hence, participants can be human and software agents. Electronic institutions infrastructure at run-time is named AMELI which is in charge of guaranteeing the participants do not violate the institutional rules established at design time.

3.3 Intelligent objects to control and assist participants' activities

An Electronic Institution models roles and activities as they happen in a real institution. Therefore, a Social Virtual World gives a 3D appearance to an EI specification, participants (both humans and software agents) are represented as avatars in the virtual world and some participant actions can be controlled and assisted by means of iObjects. The virtual world is generated from multiagent system specification (using ISLANDER tool) as described in (Bogdanovych, 2007).

iObjects are entities having both visualization properties and decision mechanisms, that help to improve human participation in a VW in the following ways:

- Representation of execution context. They provide an effective mapping of the institutional state (i.e. current good price in an auction) into the 3D virtual world. Hence, it facilitates participants perception of the current state and its changes.
- User participation. To some extent iObjects are similar to real world objects in appearance and the way to use (interact with) them. Hence, they provide an intuitive way to participate in the institution by interacting with the iObjects populating the virtual world. For instance, by opening a door to leave a room or by pressing an accept button in a remote control to accept an offer from another agent within a negotiation process.
- Enforcement of norms. iObjects collaborate with the other elements of the run time environment in the enforcement of the institutional rules. Furthermore, they can inform users when a norm has been violated and, optionally, they can guide a user in order to avoid a new wrong action.
- Guide and learn of user actions. They can incorporate a knowledge base to guide user participation (i.e. actions) inside the virtual environment. An iObject with learning abilities may gain knowledge about user actions within the simulated environment and after that, apply this knowledge to facilitate future user participation.

An iObject may have several sensors (which allow to capture events from the environment) and some effectors (which allow to act upon the environment). In the context of normative and social virtual words, by environment we mean both the virtual world and AMELI. AMELI is the component keeping the execution state and capable of verifying than an action complies with the institutional rules. An iObject central component is a decision module which determines, taking into account sensors inputs, iObject's effectors actions.

Though their sensors, iObjects can perceive events occurring at the virtual world due to avatar actions and movements. For instance, *touching* sensors allow iObjects to perceive avatars interacting with them, while *proximity* sensors allow them to react to avatars presence. An iObject can also interpret *gesture* events which allow it to act according to avatar gestures, for example a shaking head meaning "I disagree" in a e-business meeting or a raising hand meaning "I want to bid" in a auction house. Another source of events for iObjects is AMELI. That is, iObjects should be aware of changes in the execution state, in Figure 1 named *state* variables. For example, changes in the interaction context within a scene (e.g. current price of a good in an auction house), the fulfilment of a pending obligation by a participant, or *norms* changes (e.g.. a door has been opened to everyone because a scene activity has finished). When an iObject's sensor captures an event from the environment as consequence an iObject's effector reacts to the event. It is worth mentioning that in some cases, although the required reaction can be situated in the virtual world (e.g. opening a door), that reaction may depend on the compliance of the avatar action with the institutional rules. If this is the case, the iObject requests for institutional verification of the action to AMELI by using its *enforce norm* effectors. Then, the door will only open if the avatar is allowed to leave the room, which is checked by contacting AMELI. Furthermore, iObjects can also be informed about the result, executed or failed, of the actions for which they requested institutional verification, in this way, they can inform the user about the result of the action in a friendly way.

Effectors act upon the virtual world changing several properties of the iObject itself: the *aspect* (e.g. color, geometry, textures), the *information* that some types of iObjects provide (e.g. notice board) and *transformation* properties (e.g. position, rotation and scale). For example, an intelligent e-business room may scale if there is an increasing number of clients populating the space, or if it is difficult to overcome the change of its dimensions by a scaling transformation may even replicate itself. An iObject's effectors may also maintain AMELI informed about changes of the *current state* of execution, for example a door iObject informs that an avatar has moved from one scene to another one.

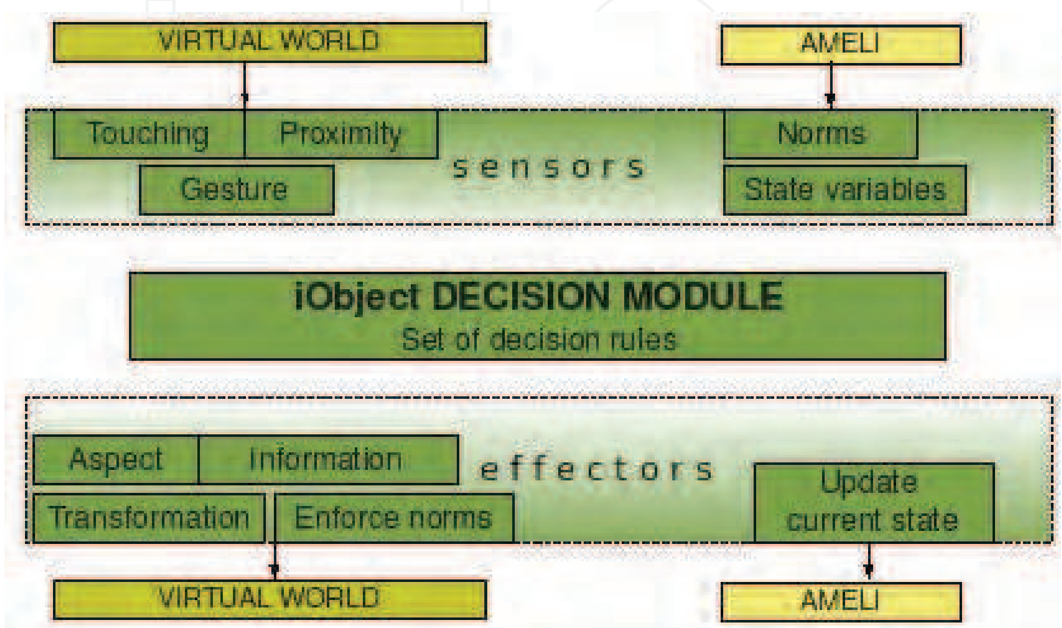


Fig. 1. Intelligent object structure

Every iObject may have some of the following features: *actionable*, *state modifier*, *self-configurable*, *learnable*. *Actionable* iObjects offer the avatar the possibility to act on them. An example of actionable objects are remote controls or a touch screen. iObjects are *state modifiers* if they may change the execution state, as for instance a door or a remote control. In the first case, because there are avatars moving from one scene to another, and in the second one by modifying the current winning bid within an auction. On the contrary, a brochure, a touch screen or an item on sale are merely informative. A *self-configurable* iObject (e.g.., a brochure or an item on sale) adapts its features according to changes in its environment. Finally, a *learnable* iObject may discharge the electronic institution infrastructure of doing the same norm checking several times. For example, a door iObject may learn a pattern of norm enforcement (i.e. circumstances such as role and agent's state that let an avatar pass through the door) so that next time it would not be necessary to query the MAS organizational infrastructure.

As can be seen in Figure 2, the human participates by controlling an avatar in the virtual world. Among other actions the avatar can interact with the different iObjects within the virtual world. The user can perceive the different iObjects in the virtual world to be aware of the execution context and use this information to decide what actions to do. Figure 2 distinguishes between iObjects at *scene/institution level* and *participant level*. The first one correspond to the iObjects belonging to the scene infrastructure (e.g. noticeboard) or institution infrastructure (i.e. door). Figure 3 shows a notice board iObject showing

information about good and its price, red salmon at 3 euros, of the current round within an auction room.
iObjects at *participant level* give the user personal information about his participation in the SVW. Hence, each user perceives their own iObjects at this level containing their information. They are placed in the user interface but not in the virtual world. At this level, there are three types of iObjects, namely the backpack, the information model notice board, and the historial. The backpack keeps the user pending obligations, which are shown by clicking with the mouse on the backpack. The information model notice board shows the current values of the user information model attributes which depend on his role. For instance, within an auction house buyer attributes may be his current credit and the list of purchased goods. The historical shows a register of the user participation (e.g.. actions) within the institution.

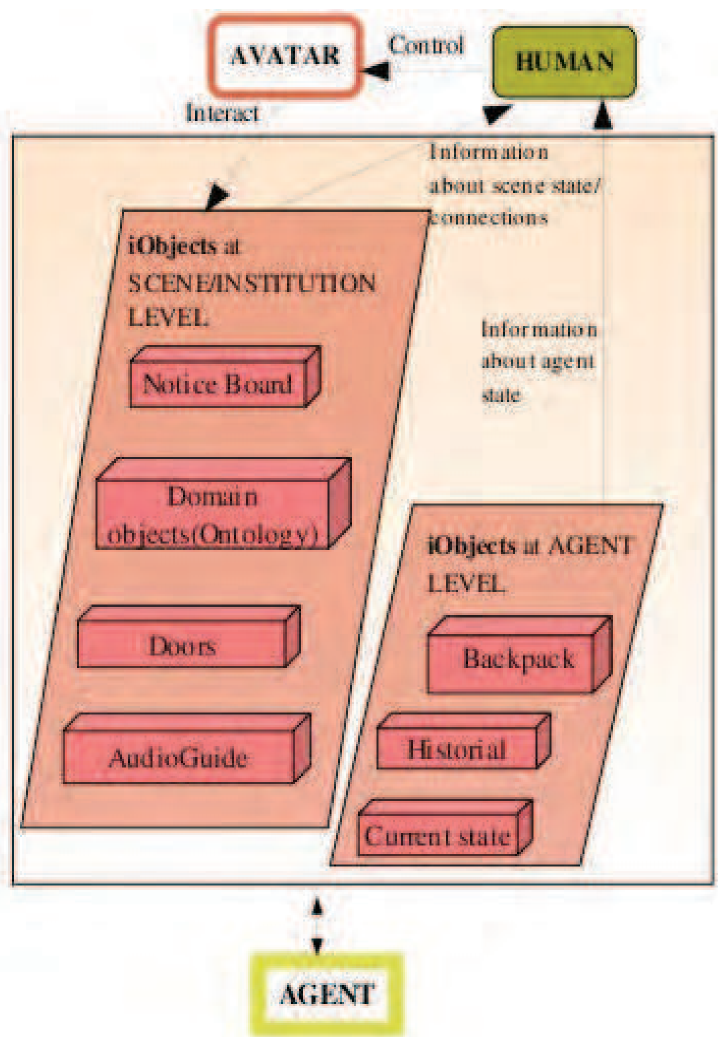


Fig. 2.Intelligent objects at scene and participant level

Fig. 3. Noticeboard iObject at Fish auction room

4. Generic framework to enforce norms in SVW

4.1 General description

We have developed a framework for generic behaviour management of virtual intelligent objects. It decouples event provider from event dealer (i.e. behaviour handling) and, compared to traditional virtual worlds, allows a better support for normative and dynamic social virtual worlds. In a conventional VW, clients take charge of rendering, interaction and behaviour handling (i.e. event capture and treatment). On the server side, digital assets are stored (in proprietary or standard format), and the server propagates client changes to the rest of connected users. The main drawback of this architecture is that an object behaviour has to be reprogrammed when VW platform changes.

Our approach gets behaviour handling out of the VW platform. It is treated in an external module named iObjects manager. An iObject is a 3D entity populating the virtual environment which is exploited in two ways: it allows normal interaction as it would do in the real world (e.g. approach/touch a door to open) and its virtual nature gives an added value to the provided information (e.g. adapts dynamically color or size). More information on iObjects can be found in (Rodriguez et al. 2008) . In the virtual world, iObjects ensure participants norm compliance and give the user assistance during his participation.

iObjects' manager is designed to be used by several virtual world platforms. To do that, it is needed to develop an extension module, iObjects extension in Figure 4, in the VW platform that will communicate with the generic manager using a socket connection. Next section presents the prototype we have developed in Wonderland virtual world (from Sun Microsystems) and presents some simulation results.

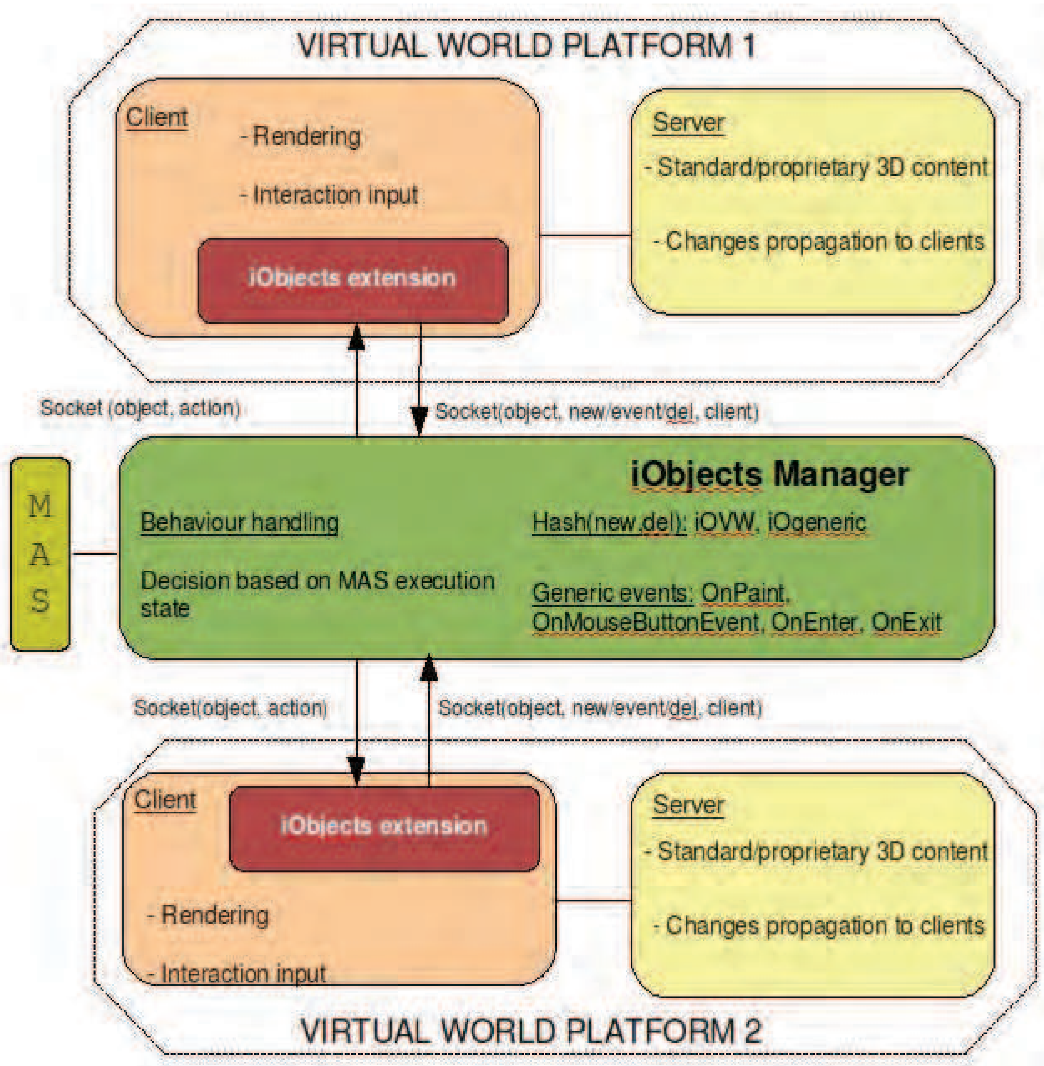


Fig. 4. Generic approach to enforce norms in a SVW

As can be appreciated in Figure 4, an interaction with an iObject is captured in the virtual world client and it is sent, using a socket message, to the iObjects manager. The message indicates client identifier, object and event used to interact with the object. The iObject manager decides which iObject action (e.g.. change color, size, trigger animation) has to be sent back to the VW. This decision is based on a response given by an organization-based multiagent system which establishes norms and possible interactions. The manager maintains a hash with iObject identifier in the concrete VW (ioVW) and its generic counterpart (iOgeneric). Currently, generic iObject events contemplated are OnPaint, OnMouseButton, OnEnter (an avatar enters in an area near to object's position) and OnExit (an avatar leaves an area near to object's position). Note that it is needed to do a mapping between concrete VW events and generic ones contemplated by the iObjects manager.

4.2 Prototype

There are several VW platforms to develop an interactive virtual environment, Second Life, Active Worlds and Wonderland, to name a few. All of them consist of similar components such as avatars, buildings, scripting components and built-in features. We chose

Wonderland because it was conceived to work with 3D standards, it is open-source and multi-platform (java-based). We have developed our prototype in WL 0.4 where 3D content is represented in X3D standard format. WL 0.5 works with COLLADA, a well-extended 3D interchange format. We are now migrating to version 0.5 available only for developers.
Once selected the VW platform, we studied how to incorporate iObjects in WL and how to capture an event (i.e. an object interaction) in WL and communicate it to the external and generic iObject manager. In particular, our prototype presents results obtained using an iObjectDoor. In WL, doors are merely holes allowing to pass through them to avatars and so change from one room to another one. An iObjectDoor adds an additional nuance letting pass through it only avatars having permission, that is, avatars who comply with the norms established by the multiagent system refered in Figure 4. Figure 5 shows two simulations exploiting norm compliance for an *iObjectDoor* in Wonderland. Client 1 (named c1a) sees the door in green because he complies with the norm allowing to enter the next room. Client 2 (named c2) sees it in red because he does not comply the norm. Note that both snapshots correspond to the same door in the same virtual world but thanks to a multi-view scheme both clients see the same door with different colors depending on their permission to pass through it. Avatars without access permission have always the collision control enabled so that they can never get closer to the door.

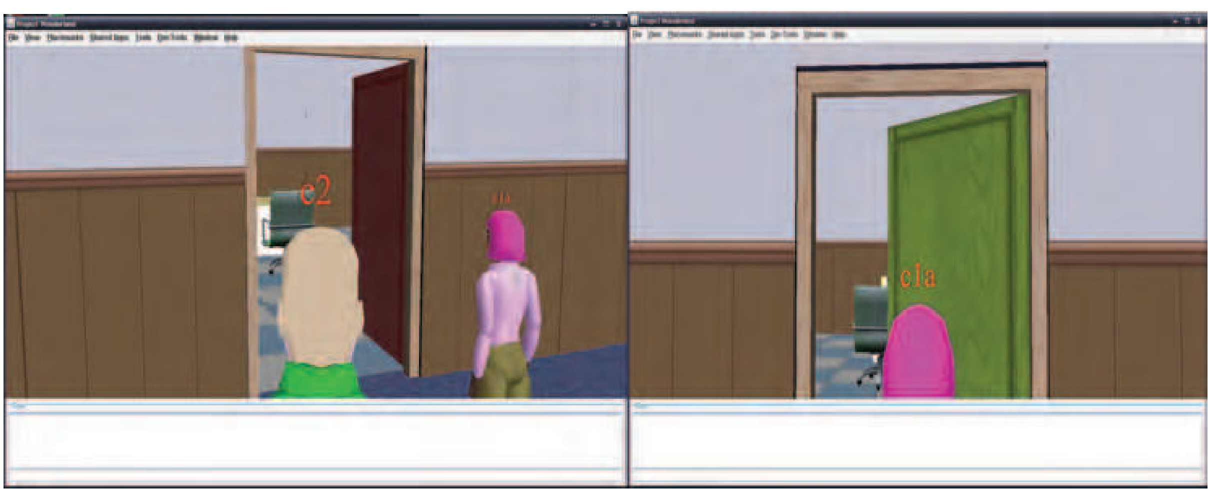


Fig. 5. Controlling norm compliance by means of an iObjectDoor in Wonderland (on the left: Client1, on the right: Client2)

Virtual worlds can be exploited as dynamic information spaces, for example, adapting the visualization of a virtual object depending on the participant profile or previous activities. As mentioned before, we propose an iObject multi-view scheme by keeping different 3D models of the iObject. All clients share an indexed set of visual representations (red door, green door, glazing door, etc.), but only one is active for each client in a given moment. Figure 6 shows the multi-view scheme of an iObjectDoor in Wonderland. On the left side, Client 2 sees glazed red door because he has permission to see the next room but not to pass through it. On the right side, Client 3 has both permission to see and to pass through it.

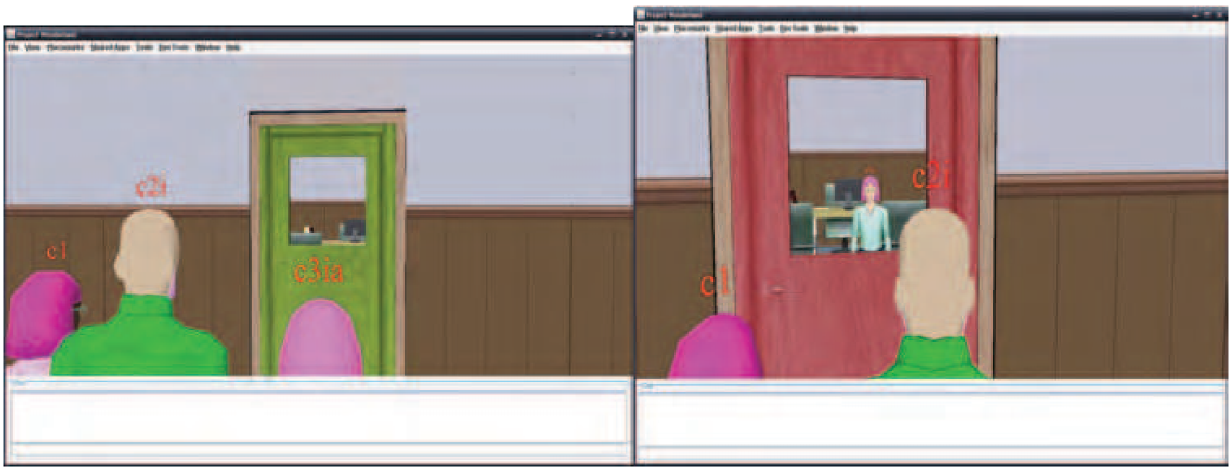


Fig. 6. Snapshots showing multi-view scheme. Client 2 and client 3 views of the iObjectdoor on left and right pictures, respectively

When an avatar is near to the door and clicks the mouse over the door, the iObjectDoor captures the event and asks the iObjects manager whether the client complies norms allowing to access the room (e.g.. in an auction, the buyer has paid registration fee). Then, if the answer is affirmative, the iObjectDoor runs the local animation and notifies it to the server so that the rest of clients also visualize it.

5. Conclusions

In this paper we have presented a system that merges multi-agent systems and virtual environments in order to model roles, activities and norms characterizing a social virtual world. We contribute with an intelligent object framework to enforce established norms and provide feedback and guide the participants on their activities. We propose a generic behaviour management for these objects populating a virtual world. We get behaviour handling out of the VW platform so that it is performed in an external module named iObjects manager allowing to be exploited by different virtual world platforms. An iObject is a 3D entity populating the virtual environment which is exploited in two ways: it allows normal interaction as it would do in the real world (e.g. approach/touch a door to open) and its virtual nature gives an added value to the provided information (e.g. adapts dynamically color or size depending on the client). We have the interoperability between different virtual world technologies in mind and so provide a general solution in which participants can be connected from different immersive environment platforms.

As future research, there is an interesting work to do regarding iObjects role at design time, i.e. when the 3D virtual world is generated from an institution specification. Shape grammars, semantic annotation and template based techniques could help us to generate and populate an initial design efficiently. In particular, we are in an initial stage of shape grammar exploitation as an alternative method for layout plan generation. As another issue of future research, iObjects could also incorporate sound sensors to obey voiced commands. We also plan to extend the iObject module with new types of intelligent objects (e.g. notice-board, brochure) and test its functionality in other VW platforms such as SL or Active Worlds.

6. Acknowledgements

Partially funded by projects IEA (TIN2006-15662-C02-01), AT (CONSOLIDER CSD2007-0022), EU-FEDER, CICYT TIN2008-02903 and by the Generalitat de Catalunya under the grant 2005-SGR-00093. M.Esteva enjoys a Ramon y Cajal contract from the Spanish Government.

7. References

Abaci T., Ciger J., and Thalmann D.(2005). *Planning with smartobjects*. In Int. Conference in Central Europe on Computer Graphics, pages 25–28.

Anastassakis G., Panayiotopoulos T., and Ritchings T. (2001a). *Virtual agent societies with the mvital intelligent agent system*. In International Workshop on Intelligent Virtual Agents,pages 112–125.

Anastassakis G., Ritchings T., and Panayiotopoulos T. (2001b). *Multi-agent systems as intelligent virtual environments*. Lecture Notes in Artificial Intelligence, 2174:381–395.

Arcos J. L., Esteva M., Noriega P., Rodriguez-Aguilar J.A., and Sierra C. (2005) *Environment engineering for multiagent systems*. Engineering Applications of Artificial Intelligence, 18(1):191–204, January 2005.

Bartle R. (2003), *Designing Virtual Worlds*. New Riders.

Bogdanovych A., Esteva M., Simoff S., Sierra C., and Berger H (2008). *A methodology for developing multiagent systems as 3d electronic institutions*. In M. Luck and L. Padgham, editors, Agent-Oriented Software Engineering VIII., volume 4951 of Lecture Notes in Computer Science, pages 103–117. Springer Berlin / Heidelberg.

Bogdanovych. A. (2007) Virtual Institutions. PhD thesis, University of Technology, Sydney, Australia.

Esteva M. (2003). Electronic Institutions: from specification to development. PhD Thesis Universitat Politècnica de Catalunya (UPC), 2003. Number 19 in IIIA Monograph Series. IIIA, 2003.

Friedman D., Steed A., and Slater M. (2007), *Spatial social behavior in second life*, Lecture Notes in Computer Science, Springer, vol. 4722, pp. 252–263.

Guyot P. and Honiden S.(2006) *Agent-based participatory simulations: Merging multi-agent systems and role-playing games*. Journal of Artificial Societies and Social Simulation, 9(4), Oct.

Harry D. and Donath J. (2008), *Information spaces: Building meeting rooms in virtual environments*, in Conference on Human Factors in Computing Systems, pp. 3741–3746.

Jorissen P., Wijnants M., and Lamotte W. (2004), *Using collaborative interactive objects and animation to enable dynamic interactions in collaborative virtual environments*, Conference on Computer animation and Social Agents

Jorissen P., Wijnants M., and Lamotte W. (2005), *Dynamic interactions in physically realistic collaborative virtual environments*, IEEE transaction on visualization and computer graphics, vol. 11, no. 6, pp.649–659.

Kallmann M. and Thalmman D. (1998) *Modeling objects for interaction tasks*. In Proc. Eurographics Workshop on Animation and Simulation, pages 73–86.

Kallmann M., Monzani J., Caicedo A., and Thalmann D. (2000), *A common environment for simulating virtual human agents in real time*, in Proc.Workshop on Achieving Human-Like Behavior in Interactive Animated Agents.

Kallmann M., Monzani J., Caicedo A., and Thalmann D. (2000). *A common environment for simulating virtual human agents in real time*. In Proc. Workshop on Achieving Human-Like Behavior in Interactive Animated Agents.

Kitio, R., Boissier O., Hubner J., Ricci A. (2007) *Organisational artifacts and agents for open multi-agent organisations: giving the power back to the agents*. In European Workshop on Multi-agent Systems.

Linden, Lab (2008). *Second life term of services*, http://secondlife.com/corporate/tos.php.

Osterfelt S. (2001) *A customer service odyssey*. In DM Review Magazine.

Rodriguez I., Puig A., Esteva M., Sierra C., Bogdanovych A., and Simoff S. (2008), *Intelligent objects to facilitate human participation in virtual institutions*, in Conference on Web Intelligence.

Rodriguez I., Salamo M., Lopez-Sanchez M., Cerquides J., Puig A., and Sierra C.(2007). *Completing the virtual analogy of real Institutions via iobjects*. In Congres Internacional de l'Associacio Catalana d'Intelligencia Artificial.

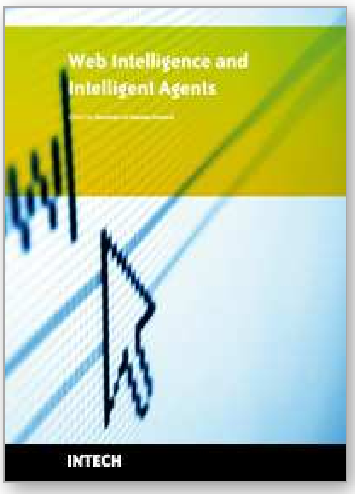
Sherman W. R. and Craig A. B. (2002) *Understanding virtual reality: interface, application, and design*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Sun Microsystems (2008). *Wonderland from Sun Microsystems*. https://lg3d-wonderland.dev.java.net/.

Torres J. A. , Nedel L. P. , and BordiniR. H. (2003) *Using the bdi architecture to produce autonomous characters in virtual worlds*. In IVA, pages 197–201.

Vosinakis S., Anastassakis G., and Panayiotopoulos G. (1999) *Diva: Distributed intelligent virtual agents*. In Workshop on Intelligent Virtual Agents.

Wright T. E. and Madey G. (2008), *Wonderdac: An implementation of discretionary access controls within the project Wonderland*, in Tech report. Univ. of Notre Dam.



Web Intelligence and Intelligent Agents
Edited by Zeeshan-Ul-Hassan Usmani

ISBN 978-953-7619-85-5
Hard cover, 486 pages
Publisher InTech
Published online 01, March, 2010
Published in print edition March, 2010

This book presents a unique and diversified collection of research work ranging from controlling the activities in virtual world to optimization of productivity in games, from collaborative recommendations to populate an open computational environment with autonomous hypothetical reasoning, and from dynamic health portal to measuring information quality, correctness, and readability from the web.