Establishing Social Order in 3D Virtual Worlds with Virtual Institutions

Anton Bogdanovych and Simeon Simoff
School of Computing & Mathematics
University of Western Sydney
Sydney, NSW, Australia
{a.bogdanovych, s.simoff}@uws.edu.au

ABSTRACT

An important security aspect of *Virtual Worlds* (in particular Virtual Worlds oriented towards commercial activities) is controlling participants' adherence to the social norms (rules of behavior) and making them follow the acceptable interaction patterns. Rules of behavior in the physical world are usually enforced through a post factum punishment, while in computer-controlled environments like Virtual Worlds we can simply block the actions that are inconsistent with the rules and eliminate rule violations as such. In order to facilitate enforcing the rules in such automatic manner and allow for frequent rule changes, the rules have to be expressed in a formal way, so that the software can detect both the rules and the actions that can potentially violate them. In this chapter we introduce the concept of Virtual Institutions that are Virtual Worlds with normative regulation of interactions. For development of such systems we employ the Virtual Institutions Methodology that separates the development of Normative Virtual Worlds into two independent phases: formal specification of the institutional rules and design of the 3D interaction environment. The methodology is supplied with a set of graphical tools that support the development process on every level, from specification to deployment. The resulting system is capable of enforcing the social norms on the Virtual Worlds' participants and ensuring the validity of their interactions.

KEYWORDS

3D Virtual Worlds, Virtual Institutions, Normative Multiagent Systems, Artificial Intelligence, Regulations, Social Norms.

INTRODUCTION

Every day in the physical world we participate in a number of institutions. Once we enter a work place, shop or university we realize the change of the context and start obeying the rules of the environment we have entered. Our behavior is highly influenced by these rules, which range from not strictly enforceable and rather implicit social conventions (like etiquette) to more explicit and usually strictly controlled instructional *norms* (like having to walk through a metal detector in an airport or to pay for the purchased items before exiting a shop).

The *institutions* are trusted third parties, which establish the rules of the interactions and administer strict control in regards to their enforcement (North, 1990). The establishment of the institutions helps in reducing the complexity in the decision making of the participants as well as in increasing the trust between individuals and improving the security of their interactions (Schotter, 1981). Government is one of the most prominent examples of an institution. The rules it tries to enforce are various laws present in the country the government is representing.

In human societies there are two available rule enforcement mechanisms: preventive measures, which prevent the rule violation from happening; and sanctions, which are used to punish the rule violator (Fehr & Fischbacher, 2004). While all the possible efforts are put into developing preventive measures, the physical world is so complex that it is impossible to prevent many rule violations from occurring. Therefore, the majority of the rule-control mechanisms used in the physical world is sanction based. The rule enforcement mechanisms of a government, for example, include the employment of armed forces like police or the army.

The *participants* of non-gaming *Virtual Worlds* like Second Life also have to deal with institutions when they access virtual classrooms, sell virtual goods or attend research conferences. Unlike the physical world the environment of a Virtual World is both computer-controlled and computer-generated. Furthermore, the complexity of a Virtual World is much lower than the complexity of the physical world. In such environments it is more efficient to employ preventive rule enforcement measures rather than use sanctions. The institutional rules of a Virtual World can potentially be expressed in a formal way and their enforcement can happen automatically by blocking all the actions that are inconsistent with the institutional rules.

Despite such exciting prospects, Virtual Worlds in their current form are rather anarchical environments developed on an ad-hoc basis. Instead of being properly formalized and automatically enforced, institutional rules are often assumed to be part of the common sense knowledge of the users. Some of them are expressed in the terms of services of a particular software product. When it comes to regulation, the rules are still often enforced in a sanction-based manner: by issuing warnings to the rule violators or even banning them if the violation reoccurs. To our knowledge, none of the existing Virtual Worlds offers a centralized technological solution that would enable structuring the interactions of participants and establishing social order in an automatic manner. Moreover, there is no widely used methodology being employed by Virtual Worlds developers that is structured around formalizing the rules of participants' interactions. Even for those cases where the rules of behavior are controlled by the code of the software, having no clear methodology and no formal representation of the rules makes it extremely difficult to introduce rule changes in the system.

Many existing Virtual Worlds are developed for engaging users into some sort of entertainment activities where participants value their virtual experience because of the freedom of interactions, and lack of norms and boundaries. Introducing the explicit social norms and the computational control of those might not be an appropriate measure for such environments. However, Virtual Worlds also proved to be a very promising technology for so-called "serious games" in domains like electronic commerce, education, tourism, etc. In these domains the absence of formal rules of behavior and reliable methods to enforce those is rather unfortunate and is, in our opinion, one of the key contributors to the present low acceptance of the Virtual Worlds technology by the industry. Therefore, the remainder of the chapter is focused on developing technological facilities for structuring the interactions of participants (the term "*participants*" is used throughout the chapter to refer to both human users and computer-controlled characters participating in a Virtual World through their avatars) in the Virtual Worlds oriented towards 'serious' games, not the general purpose Virtual Worlds.

A key contribution of this work is the concept of *Virtual Institutions*. The key idea behind Virtual Institutions is to treat *Virtual Worlds* as *Normative Multiagent Systems* (Boella, Torre, & Verhagen, 2007) that cater for mixed society of software *agents* and humans. Normative Multiagent systems is a field of research within Artificial Intelligence that suggests to treat complex computational systems as societies of autonomous entities (agents) which abide by a formally defined set of social norms and interaction protocols. The implementation focus of Normative Multiagent Systems is on programming the environment in terms of the rules of behavior of participants and acceptable interaction protocols rather than on modeling the behavior of individual agents. Such focus permits ignoring the internal architecture of the individual agents' architecture and concentrate on the system design instead.

Conceptually, *Virtual Institutions* are Virtual Worlds with normative regulation of interactions. Technologically, the current implementation of the Virtual Institutions concept incorporates a three-layered deployment framework and a design methodology. The methodology separates the development of Virtual Worlds based on the concept of Virtual Institutions into two independent phases: specification of the institutional rules and design of the 3D interaction environment. It is supplied with a set of graphical tools that support the development process on every level, from specification to deployment. The proposed approach is not only useful for helping humans to establish social order in Virtual Worlds but also for helping autonomous agents to reduce the uncertainty about the world, understand and learn the rules of the interactions and interpret the actions of the others. Although in the current form Virtual Institutions are only concerned with establishing explicit rules they can also be employed for establishing implicit social conventions.

Further in this chapter we present the concept of Virtual Institutions, establish the case in favor of defining and operating Virtual Worlds as Normative Multiagent Systems, outline extensive details of the Virtual Institutions Methodology that should be used for design of Normative Virtual Worlds and describe the deployment architecture we have developed. The presentation is concluded by illustrating the Virtual Institutions concept with an example, summarizing the contribution and outlining the directions of future work.

VIRTUAL INSTITUTIONS

Virtual Institutions is a paradigm that facilitates establishing social order in Virtual Worlds. The concept of Virtual Institutions is defined as follows:

Virtual Institutions are 3D Virtual Worlds with normative regulation of interactions.

More precisely, we propose to separate the development of Virtual Worlds based on the concept of Virtual Institutions into two independent phases: specification of the interaction rules and design of the 3D Interaction environment. For producing more efficient designs such separation is widely used in architecture (Maher, Simoff & Mitchell, 1997), from where we adapt elements of activity/space modeling applied to Virtual Worlds. Apart from design efficiency, in our case this separation has the following advantages:

- it helps in achieving clear distribution of the development tasks between system analysts and designers;
- explicitly focusing the attention of system analysts on interactions enables them to analyze the system in details and elicit substantial requirements specification before creating the visualization, which is useful for detecting critical issues and errors at an early stage;
- it makes the specification of the interaction rules independent of particular Virtual Worlds technology used for the visualization of the system, permitting a quick and easy portability to new visualization platforms.

To be able to support the conceptual separation between the design of a Virtual World and normative control of the interactions within this space we conceptually model Virtual Institutions as two conceptual layers: Visual Interaction Layer and Normative Control Layer.

The Visual Interaction Layer maps to the domain of 3D Virtual Worlds. It is concerned with audio and visual aspects of the multimedia, as well as with visualization of the interactions of participants in the 3D Virtual World.

The Normative Control Layer maps to the domain of Normative Multiagent Systems. Electronic Institution (Esteva, 2003) is the particular type of a normative multiagent system that we employ for this case. An Electronic Institution is understood as a computational environment that is enacted in accordance with a certain formal specification. The specification includes common language, a hierarchy of acceptable roles, the role flow policy, norms of behavior and interaction protocols. In the Normative Control Layer we employ Electronic Institutions for enabling institutional control of the interactions within the Visual Interaction Layer.

Not every Virtual World requires normative control of interactions as well as not every physical world institution needs 3D Visualization. Systems that involve high degree of interactions, which need to be structured in order to avoid violations, may need institutional modeling. From these, only institutions, where 3D visualization of active components is possible and beneficial, are worth visualizing in Virtual Worlds.

Systems that could benefit from both interaction control and 3D visualization, provided by the concept of Virtual Institutions, should be built following the Virtual Institutions metaphor, presented in the next section.

Virtual Institutions Metaphor

Virtual Worlds are inspired by the metaphor of architecture. They often employ physical structures like buildings, rooms, walls, etc. to represent different kinds of activities and to separate them from one another (Russo Dos Santos et al., 2000). The metaphor of architecture is convenient as humans are mostly familiar with the concept of a building. Therefore, the metaphor of architecture seems like a good choice for Virtual Institutions and can be applied to the same (or an even wider) range of problems that we are concerned within this chapter. However, Virtual

Worlds that are expected to support normative regulation of interactions require the introduction of some design constraints to the metaphor used by such Virtual Worlds.

Virtual Institutions can be viewed as a virtual space which we call 3D Interaction Space. This space can correspond to an arbitrary 3D Virtual World populated by avatars and various objects. Inside the 3D Interaction Space a set of buildings are located, where each of the buildings represents an Electronic Institution. The appearance of the 3D Interaction Space outside the buildings can be arbitrary and the behavior of avatars is not controlled by the institutional norms. However, there are restrictions on appearance and interactions inside the buildings.

As it is hard to provide a physical metaphor for the 3D Interaction space we introduce some sort of a substitute. We present the metaphor of a garden, as a place surrounding the institutional buildings. This metaphor is well known to humans as gardens often surround residential buildings in the physical world. In this way the 3D Interaction Space can be described as the combination of the garden and institutional buildings.

Employing the building metaphor for the visualization of an institution is motivated by the fact that many institutions familiar to the participants from the physical world (like universities, courts, banks etc.) also have a brick and mortar representation. As in the physical world the walls of a virtual building create visible boundaries for norm enactment.

Each institutional building is associated with its unique set of interaction rules, which are controlled by the specification of the corresponding Electronic Institution. The participants are visualized as avatars and each of them is assigned with at least one role. Only participants with specific roles can enter the institutional buildings and once there should act according to the specification of the corresponding institution. The concept of a role is widely used in Virtual Worlds. In many game based Virtual Worlds a role reflects the fact of being a part of a selected group and determines different abilities of the participants associated with it. In non-game based Virtual Worlds the role is normally used to distinguish between fee-paying subscribers and participants with trial membership.

Further elaborating the metaphor, we see each of the institutional buildings being divided into a set of rooms that are separated from each other by walls and doors. The doors are opened or closed for a participant depending on the role and the institutional state. Walls and doors are often employed in the physical world to restrict access to some activities; hence, this choice of metaphors is consistent with physical world institutions.

Figure 1 outlines the details of the Virtual Institutions metaphor presented so far. With the help of the 3D Virtual Worlds technology the metaphor of Virtual Institutions can be visualized. All of the concepts contributing to the metaphor (Garden, Buildings, Rooms, etc.) will have their respective graphical representations inside the resulting 3D Virtual World.

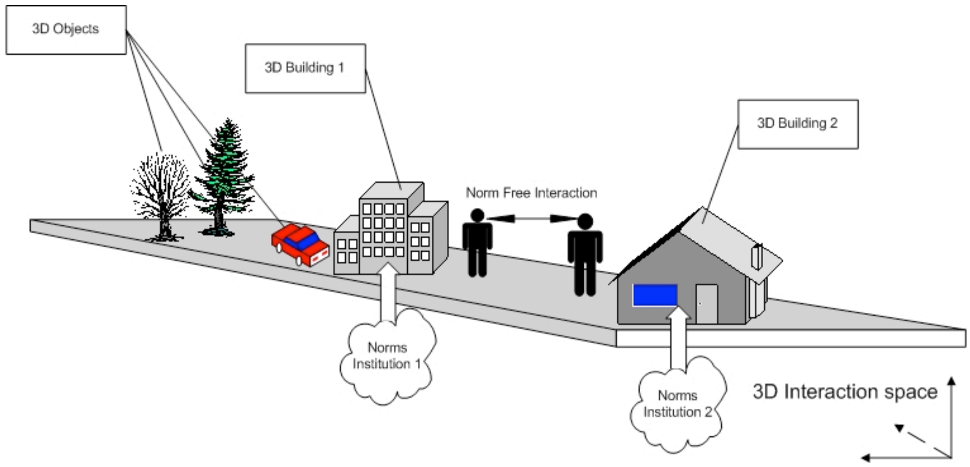


Figure 1. Virtual Institutions Metaphor

The findings in the area of Virtual Worlds design research suggest that during the construction of a 3D representation of a Virtual World, it is important to keep the benefits of traditional 2D interface design in mind (Bowman, Kruijff, LaViola, & Poupyrev, 2001). Participating in a 3D environment, where users can manipulate 3D objects, does not necessarily mean exclusion of 2D interface elements. In fact, the interaction with 2D interface elements offers a number of advantages over a 3D representation for particular tasks (Nielsen, 1998). Most efficient selection techniques, for instance, are widely realized in 2D, whereas, the selection process in a 3D user interface must consider the user's viewpoint and distance to the object.

Based on these findings we suggest enhancing the Virtual Institutions metaphor with two additional components (which in the resulting Virtual World will be visualized as additional 2D elements). The first component is a map of the 3D Interaction space, which inside a building is transformed into a building map. The map metaphor is widely used in various Virtual Worlds to assist navigation through the virtual space. Virtual Worlds can be harder to navigate than the physical worlds, and employment of maps proved to be very useful in Virtual Worlds (Darken & Sibert, 1996). Consequently, as Virtual Institutions are visualized as Virtual Worlds, we introduce maps navigational maps of Virtual Institutions.

Another additional component that contributes to the overall metaphor and which is also visualized in 2D is what we call "backpack with obligations". Electronic Institutions provide technological facilities for participants to collect obligations while acting inside the institution. If a participant does not fulfill some obligations, the institution may impose some restrictions to participant's activities. We introduce the backpack metaphor to inform participants about the reasons for imposed restrictions and to provide them with convenient facilities to visualize their commitments (so that enforcement of restrictions can be promptly avoided). This metaphor is taken from military-oriented games, in which it is usually the case that a player is given a mission, which in turn consists of a set of submissions. Any time during the game, the player can click on his/her backpack to see the mission related details. In a similar way, the backpack can be used in Virtual Institutions to show the acquired obligations to the participants.

Next we describe the components of each of the Virtual Institutions conceptual layers in terms of the new metaphor.

Visual Interaction Layer

Visualization of the Virtual World and its participants as well as providing the participants with interaction facilities is associated with the following set of concepts.

3D Interaction Space

It represents the generated 3D Virtual World, and there is no possibility for participants to move beyond it. The only way to leave it is by disconnecting from the Virtual World client. Once someone enters it, he/she will become embodied as an avatar and will be physically located inside. To enhance the believability of the visualization the space is usually populated with a number of various 3D objects, depending on the domain of the virtual institution(s) in it. In the default case, a 3D Interaction Space is decorated as a garden, where the objects enhancing the believability are trees, bushes, cars etc. A special type of object inside a garden is the building, which metaphorically represents the institution. Agents enter the institution by entering the building. Anywhere outside the institutional building interactions among participating avatars are not regulated and every event that happens inside this space is immediately visualized without any prior validation. In other words the space outside the buildings is the unregulated part of the virtual world.

Institutional Building

An institution is represented as a building in the 3D Interaction Space, and the interactions within the building are regulated by the specified institutional rules. The Electronic Institution is seen as a computational infrastructure that establishes a set of norms on the behavior of participants, who can be either humans or autonomous agents. Every event that a participant requests through an input device (typically but not limited to pressing keys on the keyboard or operating with the mouse) is first sent to the institutional infrastructure for validation. If the institution permits event's execution - the corresponding action is performed and visualized, otherwise the event is blocked and dismissed. The institution provides context-based explanations of the reasons why the event can not be processed. Each institutional building has a single entrance door, through which the participants can enter it.

Avatars

The participants of the 3D Interaction Space are visualized as *avatars*. We distinguish between the following two types of avatars: avatars for users (the institution customers) and avatars for the institutional employees. For the users' avatars an initial set of default appearances is provided, but those appearances can be changed later. The institutional employees are usually represented by autonomous agents that play internal roles in the corresponding Electronic Institution. They are assumed to have similar appearance which goes inline with the dress code of the institution they are employed with. As mentioned earlier outside the institutions the avatars are free to execute any possible action (the set is limited by the functionality of the respective virtual world) and their communication is not moderated by any of the institutions. Once they enter the institutional building they can only execute the actions that are permitted by the corresponding institution for the role in which they have entered. In some of the rooms it is allowed by the institution to split the user into several alteroids (avatars). The concept of an *alteroid* is similar to the concept of a thread in object-oriented programming. Splitting into alteroids for an agent means that two instances of the given agent are created inside a transition, which can simultaneously move into two different scenes and act there independently. Each time a new alteroid is created a user should decide which to choose to control and a new autonomous agent is executed to take control over the one that was controlled before. This functionality allows a user to employ autonomous agents for performing some routine tasks on user's behalf, while the user may be involved into some other activities.

Rooms

Every institutional building consists of a set of rooms, each one representing a different activity. The rooms are supposed to be represented as a set of rectangular boxes closed by walls from every side. Each of the rooms has at least one door, through which it can be entered. Once a participant enters a room, the only possibility for the corresponding avatar to move outside the room boundaries is to walk through one of the doors embedded into the walls of the room. To be able for a user to instantly move from one room to another it is necessary for those two rooms to be connected via a door. Unlike in many Virtual Worlds, in Virtual Institutions it is not always the case that a user can enter the room through a door and then will be able to exit it through the same door. On the contrary, when the room is entered the entrance door may be automatically locked for the user. This depends on the specification of the underlying Electronic Institution, whether it allows the backward movement between the rooms or not. In order to provide a consistent experience to the users we suggest that system designers should include the backward movement everywhere where it is possible.

Doors

The Doors are used to connect different rooms in the institutional building. The institutional rules and the execution state determine which agents (avatars) depending on their role can progress through the door. This is strictly controlled by the Electronic Institution.

Map

In order to simplify the navigation of the users, every institution is supplied by the map of the building. The map usually appears in the upper-right corner of the screen as a semitransparent schematic plan. Each of the available rooms is shown on the map and the human-like figures show every user the positions of all the alteroids a user is associated with. While moving through the institution the positions are updated accordingly.

Backpack with obligations

While acting in an institution a user may acquire some commitments. An example of such a commitment may be that a user who just won the auction will not be able to directly leave the institution, but is committed to visit the payment room before leaving. These commitments are expressed in the specification of the underlying Electronic Institution and are fully controlled by the system. In order to have a simple way to present those commitments to a user we use the metaphor of a backpack used in many computer games. The backpack is usually present in the lower right part of the screen and a user may decide to hide it or show it back after hiding. Clicking on the backpack will result in a user being presented with the textual list of the acquired commitments.

Events/Actions/Messages

Although, we anticipate that the users may use all sorts of different devices for navigating virtual worlds, in a standard case a participant of a 3D Interaction Space is able to control the avatar and change the state of the Virtual World by pressing keyboard buttons, moving a mouse or clicking mouse buttons. Those physical actions executed by a user in the real world generate events inside the Virtual World, which are then visualized as actions executed within the 3D Interaction Space. The events that a user is trying to execute inside an institutional building are not directly visualized. Before visualization every event is transformed into a message understandable by the institution and sent to the institutional infrastructure for validation. The action is performed only if a given message is consistent with the current state of the institution and it is not against the institutional rules to visualize the corresponding action.

Normative Control Layer

The description of the Visual Interaction Layer provided the explanation of the Virtual Institutions metaphor in terms of the Virtual Worlds domain. In order to explain how the institutional control of the processes inside the Virtual World is achieved we next present the mapping of the concepts presented above to the domain of Electronic Institutions. Further in this section we provide an explanation how each of the concepts described in the previous section is expressed in terms of Electronic Institutions.

A detailed explanation of the components constituting an Electronic Institutions specification is presented in (Esteva, Cruz, & Sierra, 2002). Table 1 presents a short overview of the mapping between the concepts of Electronic Institutions and the corresponding concepts of the Virtual World.

Institutions

In Virtual Institutions every building is seen as an institution. This metaphor is borrowed from the physical world, where most of the institutions are brick and mortar. Entering or exiting such an institution changes the behavior of participants, including their conversation style. Most of the processes inside each brick and mortar institution are controlled by the institutional infrastructure.

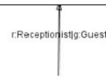
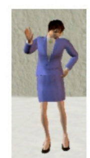
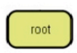
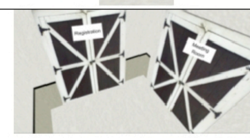
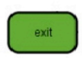
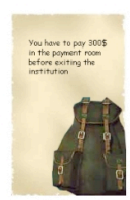
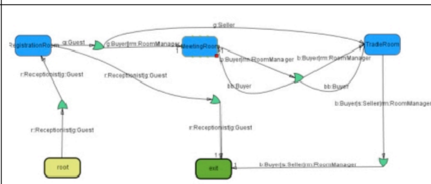
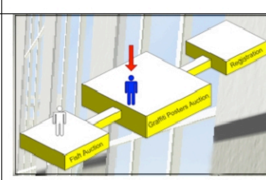
Scenes and Transitions

Scenes and Transitions in Virtual Institutions are represented as rooms and corridors. A scene in Electronic Institutions is seen as the protocol that describes a basic activity. By giving a brick and mortar representation to this concept we suggest representing each of such basic activities as a separate physical room inside the institutional building.

Transitions in Electronic Institutions serve the purpose of a middle point between two different scenes and sometimes are used for synchronization of agents. In a case when a transition is used for synchronization purposes we suggest to visualize it as a room of a special kind. Probably the most appropriate appearance for such a room would be a waiting room similar to the waiting rooms in the airports. That's the place where participants will have to wait for someone else to join them if the institution demands so.

In the Electronic Institutions specification for each of the scenes the number of participants for each of the accepted roles is always specified. We use this information to determine the “size” of the room in terms of the units of the Virtual World. The more avatars can enter the scene the bigger it should be. Hence, the size of the room is proportional to the sum of the maximum number of participants of all accepted roles.

Table 1: Mapping between 3D Virtual Worlds and Electronic Institutions

Specification Element	**Example**	**Virtual World Element**	**Example**
Scene		Room	
Transition		Room/No representation	
Connection		Door	
Number of participants		Size of the room	
Agent		Avatar	
Message	(inform (? RoomManager) (all Buyer) closed)	Action	
Root Scene		Room/No representation	
Exit Scene		Garden	
Obligations	((obl !x (inform (!x buyer) (?y buyer_ac)(payment !price)) buyer-settlement))	Backpack	
Performative Structure		Map	
Data Types in Ontology		3D Objects or No Representation	

Performative Structure and Virtual Institution layout

The original idea behind the Performative Structure is to define the main activities and specify the role flow of participants between those activities (scenes). The closest concept present in the Virtual Worlds domain is the map of the building. As it naturally happens in the Virtual Worlds, the map for each of the participants is personalized. The participants are usually only interested in the places that they can access and have much less interest in the rest. This allows us to personalize the map in a way that only the scenes that a participant is able to access are visualized there.

Every Performative structure contains special types of scenes called “root” and “exit”. These scenes are not associated with any processes inside the institution and cannot have a protocol specified for them. The root scene only serves the purpose of being the entrance point into the institution. In some cases when a root scene is connected to more than one transition it is useful to visualize it as a small room with a set of doors. Otherwise, when there is only one transition connected to it we suggest avoiding creating unnecessary rooms and do not visualize it. The entrance door of the institution in this case will lead straight into the room connected to the root scene via the first transition.

The exit scene defines the exit point. It is never visualized. Reaching the exit scene means leaving the institutional building. So, once a participant walks through a door that corresponds to the connection leading to the exit scene, the corresponding avatar will be located outside the institutional building and the exit door will be closed (as there can be no return connection from the exit scene).

Connections

The connections in the Performative Structure graph are used to determine the flow of participants between a scene and a transition. In the Virtual World we see them visualized as doors connecting different rooms. Many connections should not be visualized at all. For example, when in the specification there are several incoming connections that define entering a scene by the agents playing different roles, there is no need to create a separate door inside the corresponding rooms for each of them. In such a case only one door will be present and all the agents will use the same door for entering the room.

In the case when a transition is not visualized all the connections leading to it should not be visualized too. In such situation the rooms that are connected with each other via this transition will have a direct door connection.

Obligations

The obligations the participants acquire in an institution are represented as the backpack, which on opening displays the textual description of the obligations.

Data Types in Ontology

To be able for the agents operating in an Electronic Institution to “understand” each other it is necessary for them to establish common language. Therefore, the ontology defines a set of data types that the agents should operate with (by sending messages to each other). The majority of the Virtual Worlds inhabitants are humans, who usually are pretty good in establishing a common language without any ontological help. Mixed societies of agents and humans are a different ball game. The data types in the ontology are used to describe objects that the participants operate while interacting inside the institution. Although some data types may refer to intangible concepts, the majority of them should be represented as 3D objects. Associating data types from ontology with 3D models in the Virtual World helps autonomous agents to recognize these objects, manipulate them, track actions upon them and refer to them in conversations with humans.

Virtual Institutions Methodology

For building Virtual Institutions we propose using the Virtual Institutions methodology outlined in Figure 2. This methodology covers the whole development process and is also supplied with the tools for deployment of Virtual Institutions. This methodology should be employed by system architects and software developers (both are further called users). In general, applying Virtual Institutions methodology requires 7 steps to be accomplished:

1. Eliciting Specification Requirements.
2. *Specification* of an Electronic Institution.
3. *Verification* of the specification.
4. *Automatic Generation* of the corresponding 3D environment.
5. *Annotation* of the Electronic Institution specification with components of the 3D Virtual World.
6. *Integration* of the 3D Virtual World and the institutional infrastructure.
7. Enabling Implicit Training.

Completing the above steps will result in defining both Normative Control Layer and Visual Interaction Layer of the corresponding Virtual Institution. As shown in Figure 2, the specification requirements for the Normative Control Layer are derived on Step 1 of the methodology. The Normative Control Layer is created on Step 2 and Step 3. The development of the Visual Interaction Layer is completed after applying Step 4, Step 5 and Step 6. In case a system designer wishes to enable programming agents through implicit training mechanisms Step 7 should also be completed. Next we present a detailed overview of each of these steps.

Step 1. Eliciting Specification Requirements. The initial step of the methodology is the analysis of the application domain by system architects with the goal to elicit specification requirements. This step should result in the creation of the Software Requirements Document. In this document the key activities, roles of the participants and basic scenarios are outlined. The suggested methods for eliciting system requirements are interviews, questionnaires as well as other means of applied exploratory and descriptive research. Once the specification requirements are established, Step 2 should be used to formalize them. A detailed methodology for completing this step is outlined in (Bogdanovych, Rodriguez, Simoff, Cohen, & Sierra, 2009).

Step 2. Specification. In the current technological solution, this step is conceptually the same as specifying the institution using the Electronic Institutions methodology (Arcos, Esteva, Noriega, Rodrguez-Aguilar, & Sierra, 2005) and should also be executed by the system architects. It establishes the regulations that govern the behavior of the participants. This process is supported by ISLANDER (Esteva et al., 2002), which permits to specify most of the components graphically, hiding the details of the formal specification language and making the specification task transparent. The specification is determined by the following sets of rules:

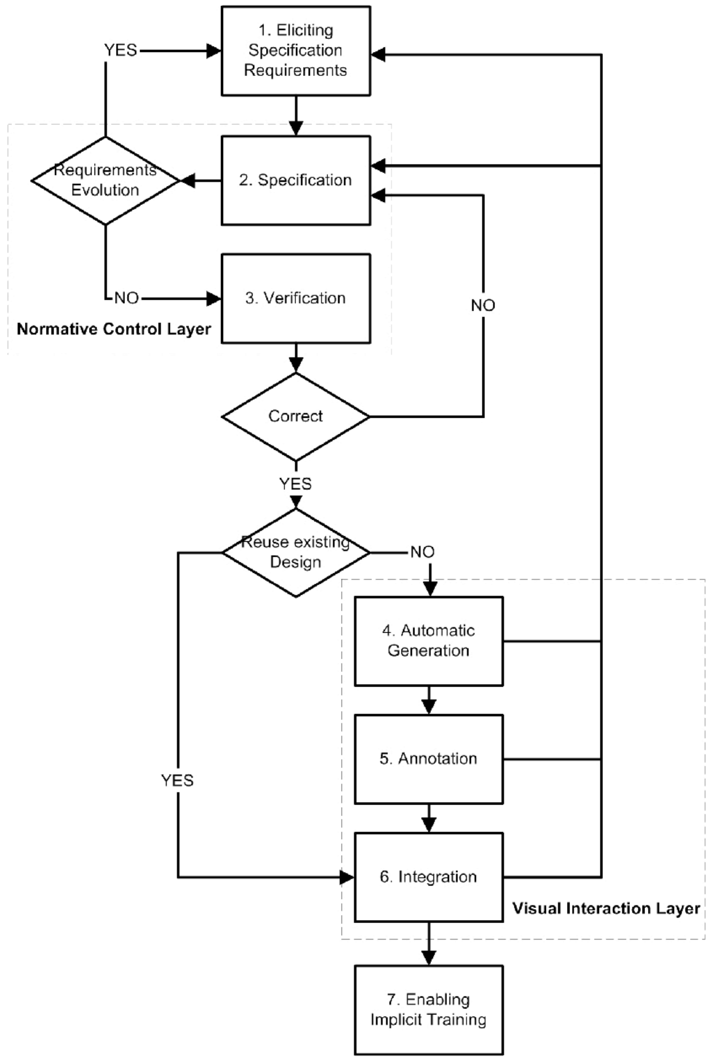


Figure 2. Methodology steps

- Conventions on language, labeled as "Dialogical Framework", which specify the roles allowed for participants, the relationships amongst those roles and a common language.
- Conventions on interactions, labeled as "Scenes", which define conversation protocols for a group of roles, where a protocol is defined as a final state machine. Each scene requires a definition of its participating roles and their population, its conversation protocol, and the states at which the participants can either leave or enter the conversation. The scenes of an institution define the valid interactions that the participants are allowed to have and set the context wherein the exchanged illocutions must be interpreted.
- Conventions on activities, labeled as "Performative Structure", which determine the types of dialogues in which participants can be engaged. Once the key activities (scenes) are identified, the role flow between different scenes is set. Each of the scenes is further associated with an interaction protocol, where a protocol is defined as a final state machine.
- Conventions on behavior, labeled as "Norms", which determine the consequences of participants' actions in different scenes. These consequences are regarded as

commitments acquired while acting in the environments and have to be fulfilled later on. These commitments restrict future activities of the participants. They may limit the possible scenes that can be entered and possible actions that can henceforth be executed.

If during the formalization it became evident that the system requirements elicited on Step 1 are insufficient or if for some other reason their evolution is required, the system architect has a choice to return to Step 1 for refining the system requirements. If no requirements evolution is necessary and the specification of the system is complete, its validity should be ensured on Step 3 of the methodology. For additional technical details of Step 2 see (Esteva, 2003).

Step 3. Verification. One of the advantages of the formal nature of the methodology is that the specification produced on the previous step can be automatically verified for correctness by ISLANDER. The tool verifies the scene protocols, the role flow among different scenes and the correctness of norms. This verification is static in nature, meaning that the specification has to be finalized before the verification can take place (in contrast to trying to verify on the fly during the specification process). The verification starts with the validation of the correctness of the protocol defined by each scene. This includes checking that the state graph of each scene is connected, that each state is reachable from the initial state and that there is a path from each state to a final state. It is also ensured that the messages associated to the arcs of the state graph are correct with respect to the Dialogical Framework.

The Performative Structure establishes how participants can legally move between different scenes. As we do not want them to get blocked in any scene or transition it is verified that from each scene and transition participants always have a path to follow, and that from any scene or transition there is a path to the final scene (so that all participants can leave the institution).

Finally, ISLANDER checks whether the norms are correctly specified and whether the participants can fulfill their commitments. As commitments are expressed in terms of actions that have to be carried out in the future, it is verified that those actions can be performed. For instance, if there is a norm that defines that a participant has to pay for acquired products, it is checked that from the scene where the products are purchased there is a path that allows reaching the payment scene.

The verification permits to detect errors in the institutional rules before starting the design and development of the Visual Interaction Layer. If such errors are found, the developers should go back to Step 2 to correct those. If the specification contains no errors there are two options how to proceed. If there is an existing design that fits the Electronic Institutions specification, i.e. the 3D visualization of the environment has already been created for an earlier design, then the developers may skip the next two steps and continue with Step 6. Otherwise, the generation step (Step 4) should be executed.

Step 4. Automatic Generation. On the generation step the 3D Virtual World and its floor plan can be created in a fully automatic way. Not only the institutional specification defines the rules of the interactions, but also helps to understand which visualization facilities are required for participants to operate in the institution. Some elements of the specification have conceptual similarities with building blocks in 3D Virtual Worlds, which makes it possible to create an automatic mapping between those. On this step of the methodology the Electronic Institution Specification is automatically transformed into the skeleton of the Virtual World. The scenes and transitions are transformed into 3D rooms, connections become doors, and the number of participants allowed to be in a scene determines the size of each room.

A method used for the automatic generation step is described in (Drago, Bogdanovych, Ancona, Simoff, & Sierra, 2007). After finishing this step the generated visualization has to be annotated on Step 5.

Step 5. Annotation. A fully immersive and visually rich Virtual World cannot be automatically created just on the basis of this specification. In order to make the generated skeleton appealing it has to be enriched with additional graphical elements on the annotation step. These additional elements include textures and 3D Objects like plants, furniture elements etc.

Apart from the elements defined in the Performative Structure most of the components of the Electronic Institutions specification do not require visualization. Such components are ignored on the generation step. In some situations, however, the visualization of these components is useful. For example, data types defined in the ontology can be represented by 3D objects. If this is the case, on the annotation step a system designer should create 3D models for these objects and manually insert them into the corresponding rooms.

This step of the methodology does not usually require the involvement of the system architects and should rather be executed by software developers and graphic designers. After this step the user can return to Steps 1 and 2 to refine the specification requirements or the specification itself, or can continue with Step 6. More details about the annotation step can be found in (Bogdanovych, 2007).

Step 6. Integration. On the integration step the execution state related components are specified. This includes the creation of the set of scripts that control the modification of the states of the 3D Virtual Worlds and mapping of those scripts to the messages, which change the state of the Electronic Institution. Firstly, the scripts that correspond to the messages from the agent/institution protocol need to be defined. These include entering or leaving a scene or transition, entering or leaving an institution, etc. Next, the scripts that correspond to the specific messages defined in the ontology on the specification step must be created. In case there were any 3D objects representing the ontology data types, the actions upon which require validation - the mapping between these objects and the corresponding data types in the ontology has to be established.

Making the integration a separate step of the methodology stimulates the development of the scripts in the form of design patterns, which are generic enough to be reused in other systems.

After accomplishing this step the generated 3D Virtual World is ready to be visualized and the Virtual Institutions infrastructure will be executed to take care of the validity of interactions between participants, verify the permissions of participants to access different scenes and make sure that all the institutional norms and obligations are imposed. This step is particularly important for the case when the system requires using an already existing design. For existing designs the integration step cannot happen automatically and, currently, manual integration is the only possible way to enable the technological connection between the Visual Interaction and Normative Control layers.

Similar to the previous step, integration should be conducted by software developers. After this step the user can return to Steps 1 and 2 to refine the specification requirements or the specification itself, or can continue with Step 7. Additional details of the integration step are presented in (Bogdanovych, 2007).

Step 7. Enabling Implicit Training. Virtual Institutions provide unique facilities for development of autonomous agents. We propose the method of Implicit Training (see (Bogdanovych, Simoff, & Esteva, 2008) for details), which is a central technology behind the decision making of the autonomous agents in Virtual Institutions. With the help of Implicit Training the agents can learn sophisticated human-like behaviors from observation of human actions in the 3D environment. Following this approach in many cases it becomes much more efficient to train the autonomous agents than to program them. In contrast to programming, such approach is much less resource consuming and a lot more flexible.

In case of Implicit Training the Normative Layer of Virtual Institutions forms the basis for the decision graph of the agent, where possible illocutions become the nodes of this graph. For each of those nodes it is possible to specify whether implicit training is conducted or not. This process is completed on the Enabling Implicit Training step of the methodology.

Deployment

Applying the Virtual Institutions Methodology described in the previous section results in the creation of the Electronic Institution Specification (the central component of the Normative Control Layer) and the 3D model of the Virtual World (containing the institutional building that represents this specification). The logical connection between the two layers is achieved on the integration step. In order to maintain this connection and make the Virtual Institution functional an additional software layer was required. In this section we present the resultant architecture for deployment of Virtual Institutions.

Our deployment architecture consists of three independent layers as shown in Figure 3. First layer here is the Normative Control Layer. It uses the AMELI system (Arcos et al., 2005) to regulate the interactions of participants by enforcing the institutional rules established on the specification step. AMELI maintains the execution state of the institution and uses it along with the specification to guarantee that participants' actions do not violate any of the institutional constraints. Each participant is introduced into the institution via an institutional agent called "governor". The governor blocks the actions that are inconsistent with institutional rules and explains those rules to the respective participant on demand.

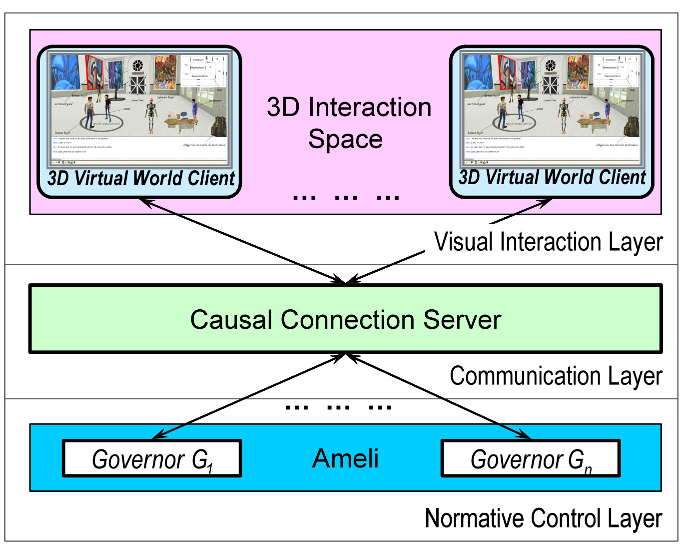


Figure 3. Runtime Architecture

Second layer is the Communication Layer. Its task is to causally connect the institutional infrastructure with the visualization system and transform the actions of the visualization system into messages, understandable by the institutional infrastructure and the other way around. This causal connection is achieved via the Causal Connection Server (Bogdanovych, Berger, Sierra, & Simoff, 2005), which uses the Action-Message table created on the integration step to establish the mapping between actions of the Visual Interaction Layer and messages of the Normative Control Layer. The causal connection is happening in the following way: an action executed in the 3D Virtual World (that requires institutional verification) results in a change of the institutional state in the AMELI layer, as well as every change of the institutional state is reflected onto the 3D Virtual World and changes its state.

The third layer, the Visual Interaction Layer, is used to visualize the 3D Virtual World for the users. The 3-layered approach not only supports the conceptual subdivision of a Virtual Institutions into the Normative Control Layer and Visual Interaction Layer as it was presented in the previous section, but also has a number of practical benefits:

1. The interactions inside the 3D Virtual World become structured, secure and predictable, as everything that needs control is verified by AMELI and will happen as specified.
2. The Visual Interaction Layer can be easily replaced (i.e. when a more advanced visualization platform appears on the market) with minimal changes required for the rest of the system.
3. The changes in the Normative Control Layer can be automatically reflected onto the Visualization layer or will require minimal manual adjustment.
4. A number of different visualization platforms (possibly implemented via different technologies) can be simultaneously connected to the Causal Connection Server and share the same institution.
5. Some of the software agents that do not interact with the humans can directly connect to the institution via the Normative Control Layer and operate (interact) there, keeping their observable presence (room location, actions within the room, etc) in 3D, so that other participants (humans) will be able to observe their presence and actions in the 3D Virtual World. This is important in terms of awareness and consistency of the environment, and markedly differentiates the Virtual Institutions technology.

CONCEPT ILLUSTRATION: TRADING INSTITUTION

In order to illustrate the concept of Virtual Institutions let consider the following scenario.

Scenario: *Imagine a fishmonger who is very interested in contemporary art. He is a regular customer of a Virtual Institution and uses its fish market auction for buying and selling fish. One of the rooms in this institution serves as the gallery for graffiti posters. The artist is present in the room and is looking forward to conversations with visitors. The fish monger enters the poster exhibition and spends his time browsing through the art works, while his other alteroid, driven by an autonomous agent, participates in the fish market auction and buys fish on his behalf.*

Institution Formalization

In this section we demonstrate how the formal specification of the Trading Institution from the above described scenario can be created using the ISLANDER tool (Esteva et al., 2002). The specification of the Trading Institution forms the basis for the Normative Control Layer of the corresponding Virtual Institution.

Dialogical Framework

The specification starts by defining the roles of the accepted participants. Each role specifies a pattern of behavior within the institution. The participants can play multiple roles at the same time and can change their roles. There are two types of roles: internal roles - played by the staff agents to which the institution delegates its services and tasks, and external roles - performed by external agents.

Figure 4 outlines the relationships amongst the roles in the Trading Institution.

"Receptionist" and "RoomManager" are internal roles in this institution, while the rest are external roles. The Receptionist is an institutional employee, whose task is to greet external participants and verify their registration details. The RoomManager is another employee responsible for starting the interactions in the Meeting Room and controlling the execution of each auction conducted in the Trading Room.

All of the external participants are playing the role "Guest". This role has two subroles: "Buyer" and "Seller". This fact is expressed in Figure 4 by two arrows marked as "Sub". Having subroles for a given role means that in the Trading Institution each Guest can become a Buyer or Seller as a result of some action (i.e. entering correct registration details) and these roles will be accepted everywhere where the "Guest" role is accepted.

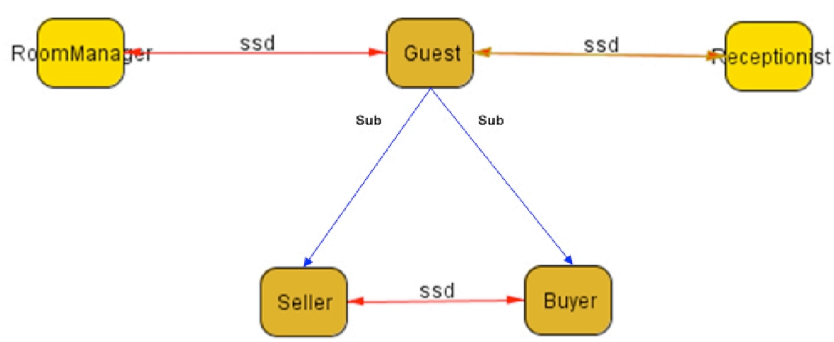


Figure 4. Roles in the Trading Institution

The two-directional arrows marked with "ssd" (static separation of duties) in Figure 4 define incompatibility between roles. In our case, once a user has entered the Trading Institution as a Guest it is not possible for this user to become a Receptionist or RoomManager. And vise-versa, a Receptionist or RoomManager cannot change their roles to become a Guest or any of its subroles. The same condition is true for Buyer and Seller. A Guest who has registered as a Buyer cannot be a Seller anymore. This security feature of linking identities to roles in each moment of time during the operation of the institution is a trade off limitation.

Once the roles of the participants are defined in order for them to be able to interact with each other we need to define a common ontology, acceptable illocutionary particles, valid communication language expressions and content language.

The ontology defined within the Dialogical Framework specifies the acceptable data types and functions (structuring the message content) that can be used inside the illocutions uttered by participants. There are standard data types: Integer, Float, String, Agent, etc. which can be instantly used. Other data types, which are compositions of the basic data types, can also be defined. The ontology used for the Trading Institution is shown in Figure 5.

This ontology specifies a single custom data type "Good", which is a tuple of the following form: (goodID : float; goodName : String). The rest of the ontology components are functions to be used in the illocutions inside scene protocols. These functions in our example are marked with an "f" symbol and include: accept, approached, closed, etc.

The illocutionary particles for this institution include: "inform" and "request".

The communication language used by the agents consists of the expressions of the following form:

$$(i\ (\alpha_i, r_j)\ \beta, \gamma, \tau)$$

In this expression:

i - is an illocutionary particle (e.g. request, inform);

α_i - can be either an agent variable or an agent identifier;

r_j - can be either a role variable or a role identifier;

β - represents the addressee(s) of the message and can be:

- (α_k, r_k) the message that is addressed to a single agent.
- r_k the message that is addressed to all the agents playing the role r_k.
- "all" the message that is addressed to all the agents in the scene.

γ - is an expression in the content language;

τ - can be either a time variable or a time-stamp;

An example of an expression pattern sent by a Guest agent to a Receptionist in the Registration Room looks like:

(request(?x Guest)(!y Receptionist)(login ?user ?pwd))

This particular expression means that agent "x" playing the role Guest sends a login request to agent "y" with user identifier and email address as the parameters of this request.

In this expression "?x" stands for a free occurrence of the variable and "!x" is its application occurrence, which means that it has to be replaced by the last bound value of variable "x".

Figure 6 shows the components of this regular expression. The dialog window on the left hand side illustrates the possible illocutionary particles that are defined in the Trading Institution. These include "request" and "inform". For an agent to be able to construct an expression as in the equation above, the "request" illocutionary particle should be defined in the Dialogical Framework of the institution.

The "Function type" dialog window in Figure 6 illustrates the definition of the "login" message inside the ontology. This definition simply specifies the number of parameters (two) and their type (both are of type String).

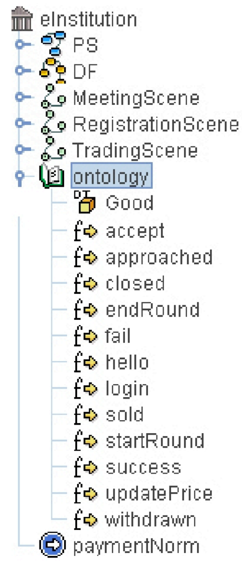


Figure 5. Ontology used for the Trading Institution.

Performative Structure

The Performative Structure defines the relationships between basic activities (scenes) of the institution. These relationships can be: causal dependency (e.g. a Guest agent must go through the registration scene before going to any further scenes); synchronization points (e.g. synchronize a Buyer and a Seller before starting a negotiation scene); parallelization mechanisms (e.g. a Buyer agent can go to multiple auction scenes); choice points (e.g. a Buyer leaving the registration scene can choose which auction scene to join); the role flow policy (which participants can access which scenes depending on their roles). Figure 7 outlines the Performative Structure of the Trading Institution.

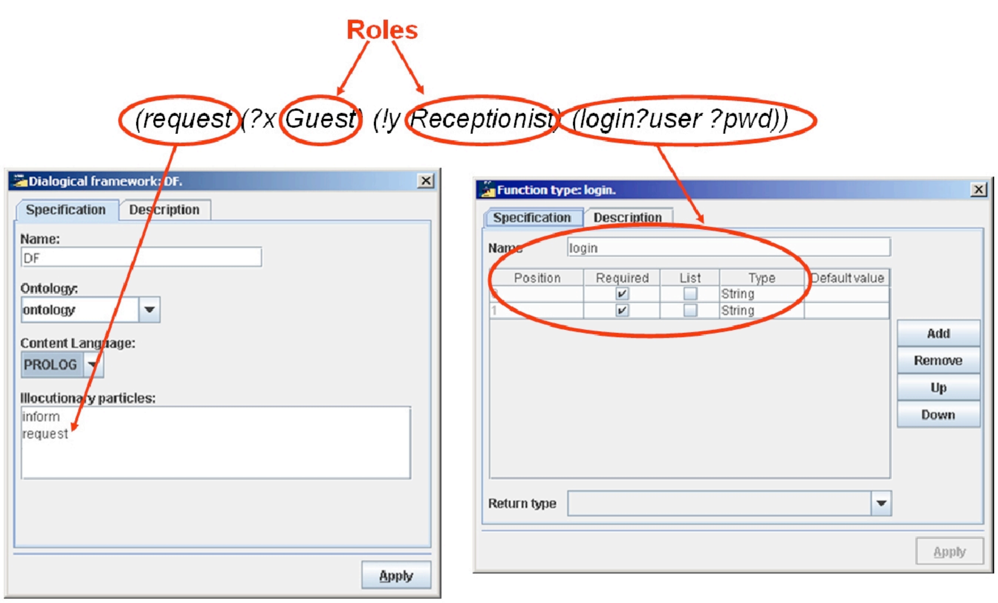


Figure 6. Illocutions, Content Language and Communication Language

The rectangular shapes in Figure 7 represent scenes, the arcs connecting them are connections and triangular shapes are transitions. Three of the scenes “Registration”, “Meeting” and “Trade” in this Performative structure are functional components. Another two scenes: “root” and “exit” are only used to show the entrance point (root scene) and exit point (exit scene) of the institution processes. When an agent joins the institution it is immediately moved to the root scene. Reaching the exit scene means leaving the institution.

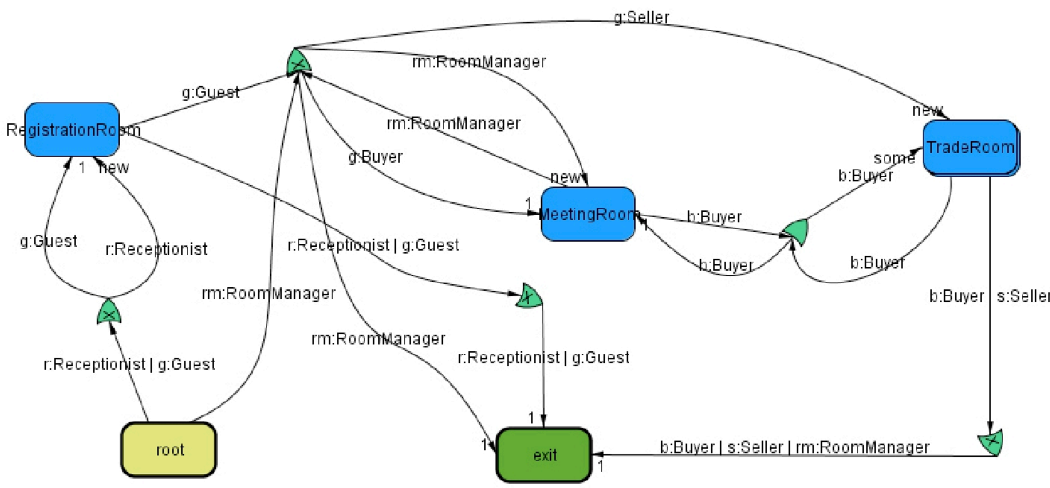


Figure 7. A Performative Structure of the Trading Institution

The transitions are used for rerouting agents between scenes and for agent synchronization. Most of the transitions in Figure 7 are marked with an “x” symbol (exclusive choice point), which means that the agents can only follow one path from such a transition. One of the transitions doesn't have this symbol (choice point). This is because with this transition it is permitted for an agent to follow multiple paths. In the particular case of the Trading Institution the choice point transition connects MeetingRoom and TradeRoom scenes. A Buyer agent is allowed to perform the stay-and-go operation in the TradeRoom (which will be further shown in the protocol of this scene). This means that the Buyer agent can split into alteroids inside this transition. There is also another type of transitions - the synchronization and parallelization point available in Electronic Institutions (this type wasn't used in our example). Such transitions can be used to force splitting agents into alteroids or force synchronization for agents with different roles.

The TradeRoom scene's appearance differs slightly from the rest of the scenes and appears to be slightly “bumpy”. This illustrates that for this scene it is allowed to have multiple executions. To have multiple executions means that a number of instances of a scene with the same protocol are created, each of them is associated with a state and the agents can join or leave these instances depending on the state and permissions.

The labels indicated above the connections define the role flow of participants. In the case of the Trading Institution the following role flow dynamics are possible:

Agents with the “Guest” role can access the RegistrationRoom scene from the root through a corresponding transition. After successful admission in the RegistrationRoom in the next transition Guests can change their role and become either Buyer or Seller. If the admission wasn't successful - the Guests are not allowed to enter any further scenes and can only leave the institution.

If admitted as Buyers, Guests change their role and can enter the MeetingRoom. From the MeetingRoom Guests can move to the TradeRoom and then either leave the institution or return back to the MeetingRoom. The label “some” on the arc connecting the TradeRoom with the corresponding transition shows that a Buyer can enter more than one instance of the TradeRoom scene.

If a Guest agent is admitted as a Seller, it can create a new instance of the Trade room and conduct an auction there. The Sellers are not allowed to enter the MeetingRoom and from the RegistrationRoom can only go to the TradeRoom or exit.

Agents with the “Receptionist” role activate the RegistrationRoom scene. This is expressed by the label “new” on the incoming connection into the RegistrationRoom. Before the scene is activated other participants cannot enter it. As the task of the Receptionist is to control the registration of the guests, from the RegistrationRoom it can only exit the institution and cannot access any other scene.

Agents with the “RoomManager” role activate the MeetingRoom. They can only access this scene. When the MeetingRoom scene is terminated the RoomManager agent can exit the institution through the corresponding transition.

Scene Protocols

The RegistrationRoom scene is used for letting the Guest agents identify themselves. It accepts only one “Receptionist” agent and twenty “Guest” agents to be simultaneously present within this scene. The protocol for the RegistrationRoom scene is presented in Figure 8. This figure also shows the illocutions that may trigger the state changes in the scene.

This protocol outlines 5 different states through which the RegistrationRoom scene can evolve (W0 - W4). State W0 is the initial state. As soon as the scene is activated it is switched into W0. In this state the agents with roles “Receptionist” and “Guest” can enter the scene. This is expressed through the labels “+Receptionist” and “+Guest” above the state.

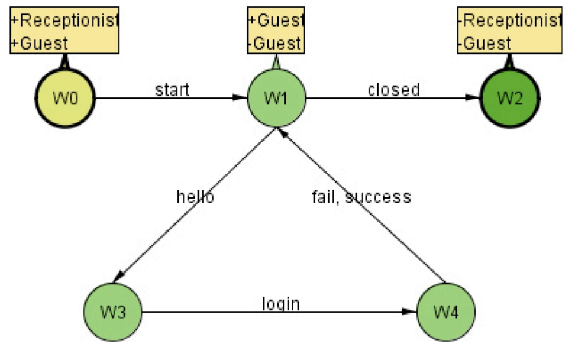
As a result of a timeout (“start” illocution) the scene will immediately change its state to W1. While the scene is in this state all the agents with the role “Guest” can enter or leave this scene.

The final state of the scene (the state in which scene is destroyed and can not be accessed anymore) is “W2”. This state can be reached from state “W1” through “closed” illocution. This illocution occurs when a Receptionist sends the “closed” message to all Guests. In the final state all the agents are forced to leave. The fact that all of them are indeed able to do so is expressed by the labels “-Receptionist” and “-Guest” above the state W2.

The registration process happens between states W1, W3 and W4. Each of the Guest agents that enter the scene while it is in W1 state will receive the identification request from the Receptionist (“hello” illocution). As the result of this illocution the scene will evolve to state W3. The Guest should respond with its login and password by sending the “login” illocution. This will change the state of the scene to W4.

If the registration details of the Guest are acceptable - the Receptionist will inform the Guest about it by executing the “success” illocution. If the registration details sent by the Guest are incorrect - the Guest will be notified about the failure to register (“fail” illocution) and can try to register once again. Either of these two illocutions will change the state of the scene to W1.

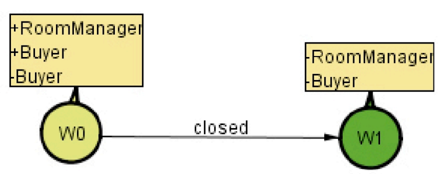
Notice that while RegistrationScene is in states W3 and W4 it is impossible for any agent to leave or enter the scene.



start: (timeout[0])
hello: (request(?x Receptionist)(?y Guest) hello)
login: (inform (!y Guest) (!x Receptionist) (?login ?pwd))
success: (inform (!x Receptionist) (!y Guest) success)
failure: (inform (!x Receptionist) (!y Guest) (fail?reason))
closed: (inform (?x Receptionist) (all Guest) closed)

Figure 8. Scene Protocol for RegistrationRoom

The MeetingRoom scene is to be used for social interactions between Buyers. The protocol and illocutions for this scene are presented in Figure 9.

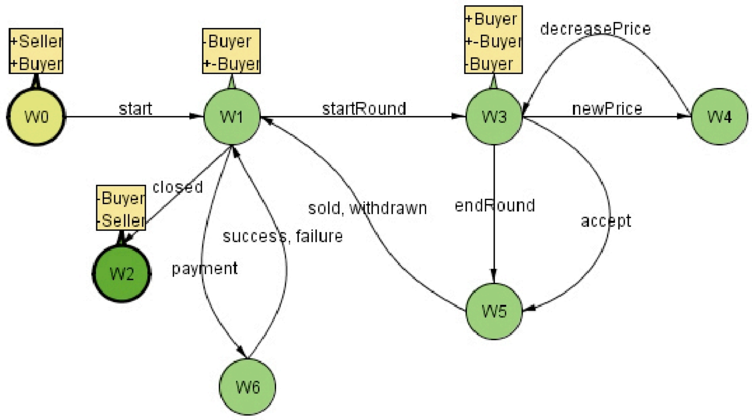


closed: (inform (?x RoomManager) (all Buyer) closed)

Figure 9. Scene Protocol for MeetingRoom

It consists of two states (W0 and W1). In state W0 agents with roles “RoomManager” and “Buyer” can enter the scene and agents with the role “Buyer” can leave the scene. It is allowed to have 1 receptionist agent and up to 50 Buyers inside the scene. The scene will evolve to the state “W1” when a RoomManager agent will decide to destroy the scene by sending the “closed” illocution. In this case all the agents will be forced to leave the scene and the scene will be terminated. Such simple protocol is due to the fact that in the particular implementation we are not concerned with controlling the socially acceptable behavior of the agents and no restrictions are imposed on how they interact in the MeetingRoom scene.

The TradeRoom scene is used for conducting different kinds of auctions following the downwards-bidding protocol. The protocol for the TradeRoom scene and the corresponding illocutions are presented in Figure 10.



start: (inform (?x Seller) (all Buyer) start)
closed: (inform (!x Seller) (all Buyer) closed)
startRound: (inform (!x Seller) (all Buyer) (startRound ?good ?price ?bidding_time))
endRound: (inform (!x Seller) (all Buyer) (endRound))
sold: (inform (!x Seller) (all Buyer) (sold !good !y !price))
withdrawn: (inform (!x Seller) (all Buyer) (withdrawn !good))
newPrice: timeout[!bidding_time/100]
decreasePrice: (inform (!x Seller) (all Buyer) (updatePrice ?price))
accept: (request (?y Buyer) (!x Seller) accept)
payment:(request (?y Buyer) (!x Seller) (payment ?price))
success: (inform (!x Seller) (!y Buyer) success)
failure: (inform (!x Seller) (!y Buyer) (failure ?reason))

Figure 10. Scene Protocol for TradeRoom

This scene has six possible states. In the initial state (W0) agents with the role “Buyer” and agents with the role “Seller” can enter the scene. From the initial state the scene can evolve to the state W1. This happens as the result of a Seller notifying all the Buyers that the auction is about to start by executing the “start” illocution. While the scene is in this state Buyers can enter and leave the scene.

The final state of this scene may be reached from state W1 through the “closed” illocution. This illocution is executed in case a Seller has no more products to sell and decides to close the auction and destroy the corresponding scene.

When a Seller decides to submit a good to be sold at the auction, the state of the scene is changed from W1 to W3 by sending the “startRound” illocution. Here the good being sold, initial price for the good and the time for the auction are announced. In the downward bidding auction the price is constantly decreasing from the initial price to the lowest price a Seller is ready to accept. This process is expressed through “newPrice” and “decreasePrice” illocutions. The bidding time is divided into 100 segments. Each 1/100 of the bidding time the timeout (“newPrice” illocution) is red and the scene evolves to W4. The price is then updated by the Seller and the scene reverts to W3.

The buyers are to react to the price changes and as soon as they see the price they are ready to pay - they should notify the Seller that they are ready to accept the current price. If a buyer agent accepts the price, it executes the “accept” illocution and terminates the auction. As the result, the scene evolves to the state W5. It will also evolve into W5 if none of the buyers have decided to accept the good within the given time frame or if the seller decides to finish the round for some other reason ("endRound" illocution).

At the end of the auction the seller either announces the winning Buyer ("sold" illocution) or withdraws a good from the auction ("withdraw" illocution). As a result of this, the scene reverts to the state W1, where Buyers can enter and leave and a new auction round for another good can be initiated by the seller.

Notice that in states W1 and W3 all participants playing the role Buyer can exercise the stay-and-go operation (this is expressed in the scene protocol with the label "+-Buyer" above the corresponding states). Such label in a scene means that the scene is in an acceptable state for agents to split into alteroids. If an agent decides to execute the stay-and-go operation then one of the alteroids remains in the original scene and all other alteroids can leave the scene and move somewhere else.

Norms

Norms define the consequences of agents' actions within the institution. Such consequences are captured as obligations. "Obl(x; ϕ; s)" means that agent x is obliged to do "ϕ" in scene s.

Norms are a special types of rules specified by three elements:

- Antecedent: the actions and Boolean expressions over illocution scheme variable that provoke the activation of the norm.
- Defeasible antecedent: the actions that agents must carry out in order to fulfill the obligations.
- Consequent: the set of obligations expressed as pairs of scene and illocution schema.

In the trading institution there is only one Norm present (paymentNorm). This norm is presented in Figure 11.

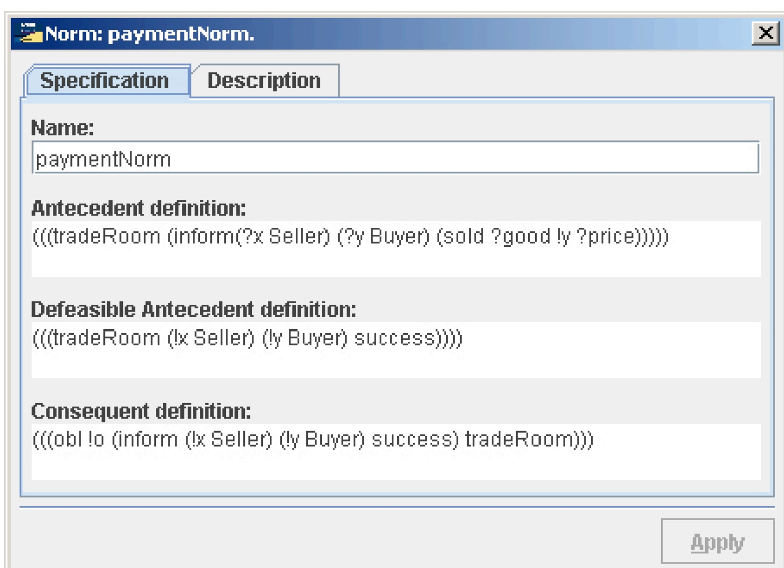


Figure 11. Norm Example

Payment norm expresses the obligation of a buyer to pay for a good that was purchased in the TradeRoom. The antecedent in this norm specifies that the norm should be activated in the TradeRoom scene as soon as a seller announces that a particular good was sold to a given buyer. Once the antecedent is fired - the buyer acquires an obligation to commit the action expressed in the defeasible antecedent. The defeasible antecedent outlines the actions that should be done to withdraw the acquired obligation. In our case the obligation can only be withdrawn if in the TradeRoom the buyer agent receives a confirmation about successful payment. The consequent shows the actual obligation the agent acquires when the antecedent is fired. In our case this obligation is not something the buyer agent has to do, but something its actions will trigger.

Norms provide an additional mechanism to control the validity of agent interactions. One of the intentions behind introducing this formalization element into Electronic Institutions was to facilitate control over the obligations an agent may acquire while moving through different scenes. For example, it is possible to have an institution similar to the Trading Institution, where payment would be completed not in the TradeRoom but in another scene. Apart from norms in this case there is no mechanism to actually control that the payment was made, while with the use of norms it is possible to establish such inter-scene validation.

Visualization

The Visual Interaction Layer of the Trading institution corresponds to the 3D Virtual World outlined in Figure 12. The Virtual World contains a landscaped garden, within which the institutional building is located. The building consists of five rooms, three of which correspond to Registration, Meeting and Trading scenes in the Electronic Institution specification. The other two rooms represent the transitions connecting these scenes in the specification. The rooms are separated by doors, where doors are either locked or open for a participant depending on its role and the state of the scene.

The Virtual World for this institution is automatically generated from the Electronic Institution specification. In the Visual Interaction Layer the Electronic Institution specification determines the skeleton of the Virtual World. In the Normative Control Layer the specification serves as the basis for the execution of the infrastructure, enforcing the interaction constraints, controlling the state of conversations, and providing permissions for different roles.

Figure 12. The Virtual World of the Trading Institution

The Performative structure forms the basis for the map of the institution and the 3D model of the institutional building. Each of the scenes ("Registration Room", "Meeting Room" and "Trade Room" in this case) lay the foundations for the generation of the 3D models of the rooms. The size of each room is determined by the maximal number of participants specified for each scene. The root and exit scenes do not have any visual representation. Transitions are transformed into special types of rooms (corridors) connecting the scenes. Connections (arcs of the Performative Structure graph) are represented as doors. Each door is initially locked, and will be opened as soon as the participant is granted the permission to enter the corresponding scene or transition by the institution.

After the automatic generation the rooms in the newly created Virtual Institution are not furnished, however, the doors, transitions and door labels are present. This institution is fully functional, which means that all the security issues of the institution will be imposed (e.g. permissions, protocols, obligations). The agents are able to freely interact and take part in conversations; the consistency of those conversations and interactions with the institutional rules is guaranteed by the infrastructure and the possibility to split into alteroids is also granted. To make the institutional building visually appealing the textures and additional objects are added at a later stage.

Each participant enters the system appearing as an avatar in the garden. The initial role given to each of the participants is "Guest". As defined in the specification any "Guest" can enter the institution. Therefore, the entrance door of the institutional building is always open for all avatars. Entering the institutional building for an avatar means entering the Registration Room. The "root" scene and the transition connecting it to the "Registration" scene are not visualized. After successful registration in the Registration Room the participant's role changes and, depending on this new role, appropriate doors are open. Opening of the doors happens independently for each particular avatar, and the same doors may be locked for a different avatar if respective permissions are not granted.

Once inside the institutional building a participant moves throughout its different rooms. The Registration Room serves as a reception desk. Once the desk is approached a participant is welcomed into the institution by the Receptionist agent and asked to enter login and password details to be able to proceed further. Figure 13 illustrates the Registration Room.

Figure 13. Registration Room Inside the Trading Institution

From the Registration Room every participant can either leave the institutional building by walking back through the entrance door into the garden or enter the corridor (transition) behind the registration room, which connects it to the Meeting Room. The door that connects the corridor with the Meeting Room is initially locked. Only when the user is correctly identified as a “Buyer” this door will be unlocked. If the user is identified as a “Seller” the door will not be open, as Sellers are not permitted to enter the Meeting Room. The exit door from the corridor in this case will contain a number of buttons on it (as in an elevator). Each button is marked with the name of the auction currently conducted inside the Trading Room. Pressing a button results in teleporting the Seller into the selected instance of the Trading Room.

The interior fragment of the Meeting Room is presented in Figure 14. To match the scenario presented above, the room is decorated as a graffiti poster gallery. The posters are presented in a conventional way (hanging on the walls) as it is usually done in physical world galleries. The participants of the gallery are represented as avatars. The avatar controlled by an autonomous agent has a robot-like appearance. All the other avatars are human participants with customizable appearances.

Additionally to the main 3D part, the Virtual World is enhanced with some two-dimensional elements. The chat window serves the purpose of the communicator between participants. To reduce the information overload and engage people into spatial interactions the maximum audibility distance is specified. Only the avatars within the audibility distance can participate in a conversation. The avatars that are not within this distance cannot “hear” the conversation. Due to this fact, the female-looking avatar present in the left part of the window is not disturbed with the conversation between the artist and the fish monger. This approach also provides means to address privacy issues: humans can clearly observe the participants of each conversation and may not give away secure information if there are undesirable participants present.

Figure 14. Meeting Room Inside the Trading Institution

Another important 2D part of the interface is the map of the institution. It is only visible if the mouse pointer is moved to the right border of the screen. The large rectangular blocks represent rooms and the smaller ones correspond to transitions. The solid figure with the arrow on top of it displays current location of the human-driven avatar within the institution. The non-solid figures represent the autonomous agents that are in the participant's subordination. As it was already mentioned, Virtual Institutions permit situations where a participant can split herself/himself into a number of alteroids. Only one of the alteroids can be controlled by a human and all the others are controlled by autonomous agents. Autonomous agents act autonomously trying to fulfill the task specified by a human. They may move around and even walk between different rooms. If in some situation an autonomous agent is unable to proceed with the given task due to the lack of intelligence, the figure representing this agent on the map starts to blink attracting the attention of the human.

The human is able to control any of the alteroids at any time by clicking the corresponding figure on the map. This will lead to switching to a different view (determined by the position and head rotation of the alteroid). The control over the avatar that the human was controlling before is automatically passed on to an autonomous agent and the appearance of this avatar is changed to the default appearance of an autonomous agent (robot look). In the given scenario the autonomous agent (non-solid figure) represents the fish monger in the fish market auction, while the fish monger drives the avatar (solid figure) through the Meeting Room.

The backpack with obligations is another two-dimensional element of the user interface. It helps the human to remember the obligations towards the institution that have to be fulfilled. The backpack automatically opens and the pending obligation is displayed if the situation of not fulfilling an obligation makes it impossible to proceed to another scene or state in the institution. The participant can also see the obligations on demand by clicking on the backpack icon.

As it is expressed in the institutional specification, the Trading Room is allowed to have multiple executions. Therefore, it is represented as a number of similar rooms placed on top of each other. Inside each of these rooms a different type of auction is conducted. One such auction is shown in Figure 15. The room instance is functioning as a fish auction. A seller is conducting an auction with lobsters being the current offer. The buyers are located around the seller's desk, waiting for an acceptable price. The red (dark in grey scale printing) robot on the left hand side is the fishmonger’s autonomous agent from our scenario.

Figure 15. Trading Room Inside the Trading Institution

Once the offer is announced, the big screen behind the seller is updated to show the picture of the current product, the remaining time left for bidding and current price. As specified in ISLANDER the auction follows the Dutch auction protocol. In this protocol the auction is conducted for a given period of time and the initial price for a good is set to be slightly higher that the desired price of the seller. Within the given time frame for a given auction round the price decreases to the minimal acceptable price. As soon as the price drops to the point when one of the buyers is ready to accept it - the buyer raises his hand to notify the auctioneer that he wants to purchase the advertised product. The auctioneer immediately announces the winner and if there are still any goods lefts to be sold - continues with the next round. After a successful purchase the buyer who won the previous round is required to approach the seller and finalize the purchase. The obligation to pay for the good is assigned to the buyer by the institution and can be observed through the buyer's backpack. Until this obligation is fulfilled this buyer will be unable to leave the institution. The product that was not sold during the round is withdrawn from the auction.

CONCLUSIONS AND FUTURE WORK

We have presented the concept of Virtual Institutions, which are 3D Virtual Worlds with normative regulation of participants' interactions. The concept of Virtual Institutions includes two logical levels: Visual Interaction Layer (responsible for the visualization of participants and for providing them with interaction facilities inside the space of the Virtual Institution) and Normative Control Layer (responsible for establishing and controlling the enforcement of the interaction rules within the Virtual Institution).

The design of these two logical levels is covered by the Virtual Institutions Methodology. To support this methodology we have developed a feasible and economically efficient technological solution, which utilizes two existing technologies - Electronic Institutions technology and Virtual Worlds technology and includes tools for formalization and deployment of Virtual Institutions.

The majority of existing virtual worlds are developed on an ad-hoc basis. Virtual Institutions is one of the first formal methodologies that addresses in a more systematic way the development of Virtual Worlds. As a formal methodology, it has a number of advantages. Firstly, employing formal methods forces the system designer to analyze the system in details before implementing it. Secondly, it permits to detect the critical points and errors at an early stage. Thirdly, the methodology clearly distinguishes and separates the two important aspects of the design of a virtual institution: the design of institutional rules and the design of institution visualization in Virtual Worlds. As a result these two processes can be run in parallel. Another advantage of using this methodology is that the supplied tools make the development faster, helping to achieve some tasks automatically. Moreover, due to this distributed multilayer architecture possible updates of the system can be accommodated faster. This in combination with the execution infrastructure permits a quick and easy portability of the system to new visualization platforms.

In their current form Virtual Institutions do not allow violating the institutional rules, however, to facilitate learning of the rules by the users it is often required to enable violation of some of the non-critical rules. In our further development of the Virtual Institutions we have focused on enabling such “soft norms” and their enforcement through sanctions.

REFERENCES

Arcos, J. L., Esteva, M., Noriega, P., Rodrguez-Aguilar, J. A., & Sierra, C. (2005). An Integrated Developing Environment for Electronic Institutions. In R. Unland, M. Klusch, & M. Calisti (Eds.), *Agent related platforms, frameworks, systems, applications, and tools* (pp. 121–142). Birkhaeuser Basel.

Boella, G., Torre, L. van der, & Verhagen, H. (2007). Introduction to normative multiagent systems. *Normative multi-agent systems* (pp. 71–79). Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany.

Bogdanovych, A. (2007). *Virtual institutions*. Doctoral dissertation, University of Technology, Sydney, Australia.

Bogdanovych, A., Berger, H., Sierra, C., & Simoff, S. (2005). Narrowing the Gap between Humans and Agents in E-commerce: 3D Electronic Institutions. In *Proceedings of the 6th International Conference on Electronic Commerce and Web Technologies (EC-Web'05)* (pp. 128–137). Springer- Verlag.

Bogdanovych, A., Rodriguez, J. A., Simoff, S., Cohen, A., & Sierra, C. (2009). Developing Virtual Heritage Applications as Normative Multiagent Systems. In *proceedings of Agent Oriented Software Engineering Workshop (AOSE 2009)* organized by AAMAS 2009 (pp. 121–132). Berlin, Heidelberg: Springer.

Bogdanovych, A., Simoff, S., & Esteva, M. (2008). Training Believable Agents in 3D Electronic Business Environments Using Recursive-Arc Graphs. In *Third international conference on Software and Data Technologies* (ICSoft 2008) (pp. 339–345). INSTICC.

Bowman, D. A., Kruijff, E., LaViola, J. J., & Poupyrev, I. (2001). An Introduction to 3-D User Interface Design. Presence: *Teleoperators and Virtual Environments*, 10 (1), 75–95.

Darken, R. P., & Sibert, J. L. (1996). Wayfinding strategies and behaviors in large virtual worlds. In Chi '96: *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 142–149). New York, NY, USA: ACM Press.

Drago, S., Bogdanovych, A., Ancona, M., Simoff, S., & Sierra, C. (2007). From Graphs to Euclidean Virtual Worlds: Visualization of 3D Electronic Institutions. In G. Dobbie (Ed.), *Australasian Computer Science Conference (ACSC2007)* (Vol. 62, pp. 25–33). Ballarat Australia: ACS.

Esteva, M. (2003). *Electronic institutions: From specification to development.* Doctoral dissertation, Institut d'Investigaci´o en Intellig`encia Artificial (IIIA), Spain.

Esteva, M., Cruz, D. de la, & Sierra, C. (2002). ISLANDER: an Electronic Institutions editor. In *First international conference on Autonomous Agents and Multiagent Systems (AAMAS 2002)* (pp. 1045–1052). Bologna: ACM Press.

Fehr, E. & Fischbacher, U. (2004). Social Norms and Human Cooperation. *Trends in Cognitive Sciences*, 8 (4), 185–190. Available at http://dx.doi.org/10.1016/j.tics.2004.02.007.

Maher, M. L., Simoff, S., & Mitchell, J. (1997). Formalizing Building Requirements Using an Activity/Space Model. *Automation in Construction*, 6 , 77–95.

Nielsen, J. (1998). 2D is Better Than 3D. *Jakob Nielsen's Alertbox*.

North, D. C. (1990). *Institutions, Institutional Change and Economic Performance*. Cambridge, U.K.: Cambridge University Press.

Russo Dos Santos, C., Gros, P., Abel, P., Loisel, D., Trichaud, N., & Paris, J.-P. (2000). Mapping Information onto 3D Virtual Worlds. In *Proceedings of the International Conference on Information Visualization* (pp. 379–386). Washington, DC, USA: IEEE Computer Society.

Schotter, A. (1981). *The Economic Theory of Social Institutions*. Cambridge, USA: Cambridge University Press.

ADDITIONAL READINGS

Rodriguez-Aguilar, J.A. (2003). On the design and construction of agent-mediated electronic institutions, volume 14 of Monografies de l'Institut d'Investigació en Intelligència Artificial. Consejo Superior de Investigaciones Científicas, 2003.

Weiss, G. (1999). Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. MIT Press, 1999.

Arcos, J. L., Esteva, M., Noriega, P., Rodrguez-Aguilar, J.A. and Sierra, C. (2005). An Integrated Developing Environment for Electronic Institutions. In Rainer Unland, Matthias Klusch, and Monique Calisti, editors, Agent Related Platforms, Frameworks, Systems, Applications, and Tools, Whitestein Book Series, pages 121–142. Birkhaeuser Basel, 2005.

Cascio, J., Paffendorf, J., Smart, J., Bridges, C., Hummel, J., Hurtsthouse, J., and Moss, R. (2007). Metaverse Roadmap: Pathways to the 3D Web. Report on the cross-industry public foresight project, July 4, 2007.

Damer, B. (1998). Avatars! Exploring and Building Virtual Worlds on the Internet. Peachpit Press, 1998.

Esteva, M., de la Cruz, D., and Sierra, C. (2002). ISLANDER: an Electronic Institutions editor. In First International Conference on Autonomous Agents and Multiagent systems, pages 1045–1052, Bologna. ACM Press, 2002.