Developing Virtual Heritage Applications as Normative Multiagent Systems

A. Bogdanovych[1], J. A. Rodriguez[2], S. Simoff[1], A. Cohen[3], and C. Sierra[2]

[1] School of Computing and Mathematics, University of Western Sydney, Australia
{anton, simeon}@it.uts.edu.au
[2] IIIA, Artificial Intelligence Research Institute, CSIC, Spanish National Research Council† {jar, sierra}@iiia.csic.es
[3] Federation of American Scientists, 1725 DeSales Street, Washington, DC, USA
acohen@fas.org

Abstract. The majority of existing virtual heritage applications are focused on detailed 3D reconstruction of historically significant sites and ancient artifacts. Recreating the way of life of ancient people is only considered by some researchers, who employ crowd simulation for this task. Existing crowd simulation algorithms are not suitable for modeling complex individual behaviors and role dependent agent interactions with other participants in the Virtual World. To address this problem we suggest treating 3D Virtual Worlds as Normative Multiagent Systems and propose the Virtual Institutions Methodology that can be used for design and deployment of Virtual Worlds that require complex interactions involving both humans and autonomous agents. To highlight the usefulness of this approach we illustrate how Virtual Institutions are employed in the development of the Uruk prototype, which integrates 3D Virtual Worlds and Artificial Intelligence in the domain of cultural heritage.

1 Introduction

Non-gaming Virtual Worlds like Second Life [1] have become an important area of research during the last few years. Many researchers stress the significance of this technology, considering it being the next stage of the World Wide Web evolution (Web 3.0). Gartner has predicted that 80% of the Internet users will be participating in non-gaming Virtual Worlds by the end of 2011 [2]. Already now millions of people spend an average of around twenty hours a week in various Virtual Worlds [3]. Furthermore, studies in South Korea have indicated that the majority of Koreans prefer 3D Virtual Worlds to television [4].

Two promising application domains for non-gaming Virtual Worlds are cultural heritage and education. In heritage 3D graphics is often used to reconstruct lost sites of high historical significance. In education the interest in Virtual Worlds is particularly strong in relation to history. In both of these domains researchers are normally focused on reconstructing destroyed or damaged buildings (e.g. the Roman Coliseum). While such an approach allows to examine the

† Thanks to projects JC2008-00337, TIN2006-15662-C02-01, CONSOLIDER CSD2007-0022

architectural details of the heritage site in three dimensions, it still does not help a general observer to understand how this site has been enacted in the past.

Populating history or heritage oriented Virtual Worlds with avatars that behave similar to the ancient citizens of the reconstructed places has a potential to provide visitors with a more engaging experience. Using human experts to control the avatars and simulate the behavior of ancient citizens is very costly, while employing Artificial Intelligence for this task (which is a much more affordable option) hasn't received appropriate attention from research. The majority of researchers that are working on populating virtual heritage sites with avatars employ so-called virtual crowds [5] for this task. Such crowds normally consist of a large number of autonomous agents (represented as avatars) dressed appropriately for the selected period of time and appearing as local citizens of the reconstructed area. The state of the art in combining crowd simulation and 3D heritage reconstruction can be observed on the example outlined in [5]. Here, a 3D reconstruction of the ancient City of Pompeii is "made alive" using a large number of avatars that walk around the city avoiding collisions. While providing a visitor with some understanding about the appearance of the ancient people, this approach poorly elaborates on specific behavioral characteristics of those people. The agents employed in [5] do not use the objects in the environment and are not engaged into historically authentic interactions.

Through the use of Virtual Institutions technology [6] we intend to bring virtual heritage to a new level by making it more dynamic and interactive. Instead of just having virtual crowds walking around the city we suggest populating virtual heritage sites with autonomous agents that reenact the most typical daily activities of the reconstructed society. Creating such agents is quite a challenging task as the degree of interaction is quite high, the agents have to depend on other agents, play different roles, synchronize their activities with other agents and even solve some tasks in a teamwork manner while actively using the objects in the virtual environment. One of our research hypotheses is that for an autonomous agent to be able to demonstrate similar complexity of actions as ancient humans, the complexity of agent's environment must be reduced. This assumption is based on the suggestion made by Russell & Norvig that the agents ability to successfully participate in some environment and extend its intelligence there is highly dependent on the complexity of this environment [7]. It is suggested that situating the agent in a fully observable, deterministic and discrete environment helps the agent to tackle the famous frame problem of AI [8].

As an example of a fully observable, deterministic and discrete environment, we consider Virtual Institutions [6] (previously known as 3D Electronic Institutions), which are 3D Virtual Worlds with normative regulation of participants' interactions. In Virtual Institutions the environment is formalized in terms of norms of acceptable behavior of participants, interaction protocols and role flow of participants. Every agent has access to this formalization, which helps it to reason about its own actions and the actions of other participants (either humans or agents) as well as to understand the consequences of these actions. The Virtual Institutions technique that we employ for such environment formalization is based on Electronic Institutions [9] widely used in the Multiagent Systems for structuring the interactions of the agents participating in open systems.

Overall, the Virtual Institutions approach to the development of applications for virtual heritage is to treat 3D Virtual Worlds as Normative Multiagent Systems. Our work builds on top of the research published in [10]. The original *methodology* presented in [10] has been applied to a real world problem and, as a result of this, *was revised*. The updated methodology includes *new steps* and also features a *detailed explanation of the first two steps*. The most significant *contribution* of this paper is providing the *evaluation* of the Virtual Institutions methodology by developing a prototype in the domain of cultural heritage.

The remainder of the paper elaborates on the details of the contributions presented above. In Section 2 we present the concept of Virtual Institutions, the corresponding methodology and technology. Section 3 provides the motivation for using Virtual Institutions in the domain of cultural heritage, illustrates the application of the Virtual Institutions methodology to this domain and outlines all the development steps. Finally, Section 4 presents some concluding remarks.

2 Virtual Institutions

We consider Virtual Institutions [6] being a new class of Normative Virtual Worlds, that combine the strengths of 3D Virtual Worlds and Normative Multiagent Systems, in particular, Electronic Institutions [9]. In this "symbiosis" the 3D Virtual World component spans the space for visual and audio presence, and the normative component takes care of enabling the formal rules of interactions among participants. Through the normative component a Virtual World is separated into a number of logical spaces (scenes). Only participants playing particular roles are admitted to a scene and can change the state of this scene. Once admitted the participants should follow the interaction protocol specified for each scene and are unable to violate this protocol. The institution doesn't take away agent's autonomy by forcing it to act in a specific manner, but prohibits violating institutional rules and enforces a particular interaction protocol.

2.1 Virtual Institutions Methodology

The Virtual Institutions methodology [10], outlined in Fig. 1 a), covers the entire development process. Its application 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 (if needed).
5. Annotation of the Electronic Institution specification with components of the 3D Virtual World.
6. Integrating the 3D Virtual World into the institutional infrastructure.
7. Enabling Implicit Training

One of the key contributions of this paper is the description of the methods that should be utilized on steps 1 and 2 of the methodology – as outlined in Fig. 1 b). Therefore, here we elaborate on these steps, while referring to the initial work published in [10] for the detailed description of other steps.

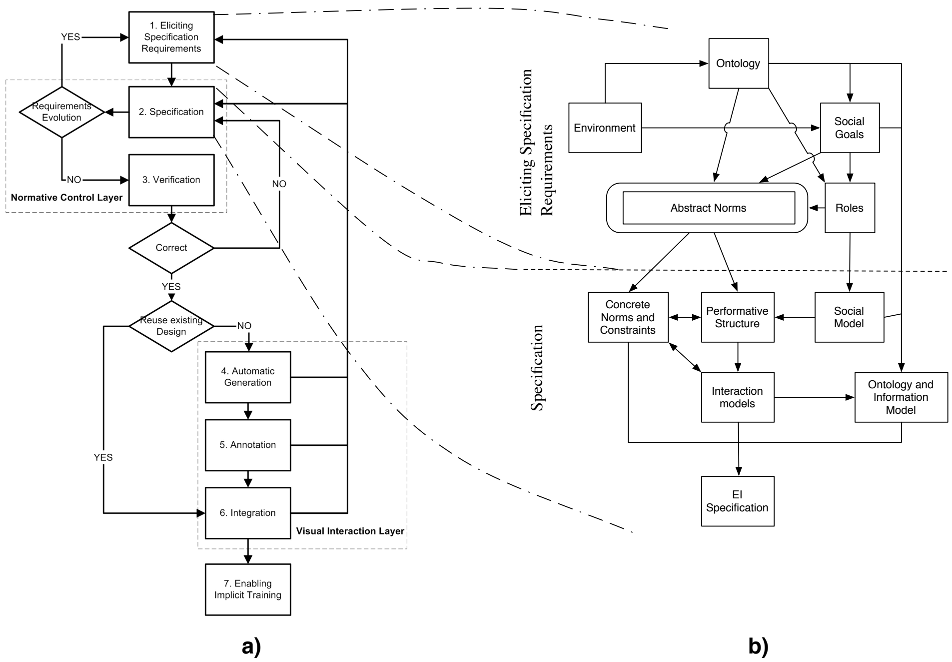

Fig. 1. Virtual Institutions Methodology.

Step 1. Eliciting Specification Requirements. This step of the methodology aims at producing the Software Requirements Document where the key activities, roles of the participants, and basic scenarios are outlined. Upper part of the Fig. 1 b) presents the details of this step. The key task here is enforcing institution designers to provide an answer to the following core issues regarding a virtual institution: where is it situated (environment), what is it designed for (social goals), how it is expected to succeed (abstract norms), which social roles are required to enact it and how can participants interact (ontology).
<u>1.1 Environment.</u> Firstly, a system architect must consider *where* the resulting system is to be deployed, in which *physical* and *social environment* it has to perform its activities, what are the factors that will affect its behaviour that it can or cannot control? We take the stance that all objects in the environment are beyond the *complete* control of the institution to be defined. Agents will interact with that environment by *observing* and by *affecting* those objects in some way. In order to cope with the mentioned lack of control we will assume that all of them are agentified and that the perception and action upon them is always dealt with in the context of a dialogue with the agent that wraps them.
<u>1.2 Ontology</u>. To define institutional norms we must express normative behaviour in some language, which in turn requires the definition of some *ontology*, namely the definition of what agents are going to talk about. The ontology can be represented as a set of concepts and relations among them. The ontology has also to accommodate the objects already found when specifying the environment.
<u>1.3 Social goals.</u> Typically institutions are regulated environments where humans

interact to attain some global objectives, which we call *social goals*. An institution aims at continuously satisfying its social goals and will not allow any agent to behave in a way that prevents from attaining them. The specification of social goals (e.g. in first-order logic or natural language) must employ the ontology defined in the previous step. The output of this step is a list of social goals.
1.4 Roles. Next, we identify the *roles* (patterns of agent behaviour) in the institution, which are determined to a large extent by the social goals. What the institution aims at achieving indicates which capabilities and behaviours are needed. Roles, at this stage, are a set of identifiers with some associated capabilities expressed as the message types (i.e. ontology terms) the role is capable of dealing with. The output of this step is a list of roles and their capabilities.
1.5 Abstract norms. The idea of an abstract norm is that of expressing a *generic restriction* on how the agents incarnating the identified roles should behave. The roles and the ontology previously analysed provide the language needed to write the expressions of the abstract norms. To avoid any confusion, the designer should clearly distinguish between abstract norms and social goals because abstract norms are assumed to refer to the *behaviour of the agents*, and not to the ultimate *purpose* of the institution, which is represented by the social goals. Abstract norms cannot prevent agents from satisfaction of any social goal. The output of this step is a list of abstract norms.
Step 2. Specification. This step establishes the regulations that govern the behavior of the participants. This process is supported by ISLANDER [11], which permits to specify most of the components graphically, hiding the details of the formal specification language and making the specification task transparent. The details of this methodology step are outlined as a number of substeps in the lower part of Fig. 1 b) that we detail below.
2.1 Social model. The initial task of the system architect is enriching the role model by adding further information to turn it into a *social model*. A distinction between internal and external roles must be made. Roles whose raison-d'être is to support the achievement of the social goals and to enforce the norms will be marked as *internal* roles and the rest as *external* roles. The following relationships between the roles relevant for the institution are specified: (i) hierarchy, (ii) ssd, static separation of duties, and (iii) dsd, dynamic separation of duties.
2.2 Performative structure. Next the definition of the set of dialogical activities permitted for different roles entails a sequence of substeps:

1. Starting from the abstract norms, the designer must define a list of scenes (activities in the institution) along with their participating roles.
2. For each scene in the list, its creation conditions (i.e. which role(s) initiates the scene), and whether it can be multiply enacted or not must be specified.
3. Based on the abstract norms, the designer gathers together scenes into a performative structure by specifying: (i) the flow of agent roles, namely which roles from which scenes can get into other scenes; (ii) the role change policy, namely whether agents are allowed to change roles when moving out of a scene into another scene. The result is a graph connecting scenes whose edges are labelled with expressions encoding the role flow and role change policies.

2.3 Interaction model. Once the performative structure is defined the designer must associate some *interaction model* to each scene, namely a specification of the dialogue in the scene (v.g. through a finite state machine). An interaction model must contain the conversation states, the illocutions exchanged between agents that permit transitions between conversation states, and the constraints that restrict these transitions. Moreover, this substep also requires to set the minimum and maximum number of agents playing each role allowed in the scene as well as the conversation states where new agents are permitted to join and participating agents are permitted to leave. Sometimes, when going through this step, we may identify new roles and thus a revised version of the social model is intermingled with this substep. Moreover, we may also identify refinements in the ontology that help us better express the constraints restricting transitions. This substep and the next one are very interrelated.
2.4 Concrete norms and constraints. A further refinement still applies to the performative structure and its scenes' interaction models. Back to the abstract norms, we may find that some of them can be translated into simple *constraints* that will limit agents in two ways: (i) by not permitting them to say certain things (by adding a constraint into some interaction model); and (ii) by not permitting them to move to a certain scene (by adding a constraint into the performative structure). Some abstract norms, however, will require a more sophisticated representation because the effect of an agent action (illocution) implies that certain other action(s) is *actually* done.
2.5 Ontology and information model. Finally, from the initial ontology definition and the concrete messages as expressed in the different interaction models of the institution, the *ontology* can be completed. Further analysis of the social model, and the specification of the performative structure and its scenes' interaction models helps to complete the identification of the *attributes* (along with their types) of: roles, objects in the environment, scenes, and the insitution itself.

Step 3. Verification. One of the advantages of the formal nature of the Virtual Institutions methodology is that the specification produced on the previous step can be automatically verified for correctness by ISLANDER [11]. The tool verifies the scene protocols, the role flow among the different scenes and the correctness of norms (see [10] for details). If any errors are found, developers must return to step 1 to correct those. If the specification contains no errors, there are two options. If the 3D Visualization of the environment is already created (reuse of the existing design) then the developers may skip the next step and continue with Step 4. Otherwise, the generation step, Step 3, should be executed.

Step 4. Automatic Generation. In some cases it is desirable to generate the initial skeleton of the Virtual World from the specification. In such cases the Virtual World can be generated automatically using the method presented in [12]. The resulting Virtual World has to be annotated on Step 5.

Step 5. Annotation. The Specification defines the rules of the interaction and has nothing to say about the appearance of the specified elements. On step 5 the specification is enriched with appearance related graphical elements. These additional elements include textures and 3D Objects like plants, furniture elements

etc. This step of the methodology does not usually require the involvement of the system architects and should rather be executed by designers and 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 6.
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. 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.
Step 7. Enabling Implicit Training. Having the complete formalization of the agent environment makes it possible to use imitation learning as a key technique for achieving human-like behavior of the agents. The institutional specification forms the basis for the decision tree of the agent, where possible illocutions become the nodes of this tree (see [13]). On Step 7 for each of those nodes we specify whether implicit training is conducted or not.

2.2 Deployment

A virtual institution is enabled by a three-layered architecture presented by conceptually (and technologically) independent layers [6]. The Normative Control Layer employs AMELI [14] to regulate the interactions between participants by enforcing the institutional rules. The Communication Layer causally connects the institution dimensions with the virtual world [6]. The Visual Interaction Layer (currently supported by Second Life [1]) visualizes the Virtual World.

3 City of Uruk: Virtual Institutions in Cultural Heritage

Uruk is a joint research project between the University of Western Sydney and the Federation of American Scientists. The aim of the project is to recreate the ancient city of Uruk from the period around 3000 B.C. in the Virtual World of Second Life letting the history students experience how it looked like and how its citizens behaved in the past. The Virtual World of Second Life provides a unique collaborative environment for history experts, archaeologists, anthropologists, designers and programmers to meet, share their knowledge and work together on making the city and the behavior of its virtual population historically authentic.

3.1 Significance of Uruk

Uruk was an ancient city located in present day Iraq (circa 250 km south of Baghdad). Many historians consider Uruk being one of the first human built cities on Earth. By 2900 B.C. Uruk is believed to be one of the largest settlements in the world and one of the key centers of influence of the Sumerian culture. Uruk played a major role in the invention of writing, emergence of urban life and development of many scientific disciplines including mathematics and astronomy.

3.2 The Prototype

The prototype aims at showing how to enhance the educational process of history students by providing them with a possibility to immerse into an accurate replica of the daily life of the ancient citizens of Uruk and gain quick understanding of the advance of technological and cultural development of ancient Sumerians. Ultimately, the students may become part of the virtual society and will have to interact with agents and other humans to solve the assigned tasks.

The 3D reconstruction of the city was produced within the Virtual World of Second Life based on the results of archeological excavations and available written sources. Both modeling of the city and programming of the virtual humans populating it were conducted under the supervision of subject matter experts.

We have selected fishermen daily life of ancient Uruk to illustrate how Virtual Institutions can enable immersive experience in the life of ancient societies. We created four agents that represent members of two fishermen families. Each family consists of a husband and wife. Every agent has a unique historically authentic appearance and is dressed appropriately for 3000 B.C.

The agents literally "live" in the virtual world of Second Life. Their day is approximately 15 minutes long and starts with waking up on the roof of the building – Fig. 2 a). Although, most of the buildings in Uruk had ventilation holes the temperatures inside (especially during summer) could become quite unpleasant and most of the citizens would prefer sleeping on the roof top in the evening, where it would have been much cooler. The wives wake up first to collect some water from the well and prepare breakfast for their husbands. The husbands start their day with a morning chat while waiting for the breakfast to be prepared (in the current prototype eating and cooking are not implemented).

a) b) c)

Fig. 2. The City of Uruk Prototype.

After breakfast the fishermen would collect their fishing gear and walk towards the city gates – Fig. 2 b). Outside the gates on the river bank they find their boat which they will both board and start fishing. One of the agents would be standing in the boat with a spear trying to catch the fish and the other agent would be rowing. Fig. 2 c) illustrates the fishing process.

After fishing, the men exit the boat, collect the fishing basket and spear and bring them back to their homes. This daily cycle is then continuously repeated with slight variations in agent behavior.

3.3 Development of the Prototype

The Virtual Institutions Methodology was employed for the development of the prototype. Here we provide a step-by-step description of how it was followed.

Step 1: Eliciting Specification Requirements. In order to come up with realistic specification requirements the system developers have conducted joint meetings with 2 subject matter experts. The decision on the roles of participants, scenes they can participate in and interaction protocols have been identified through conversations with subject matter experts. It was agreed that having four agents in the system is sufficient for the first prototype. The roles of these agents are: two fishermen and two fishermen wives. The identified scenes that must be present in the environment are: Home1, Home2, FirePlace, Well, Chat and Fishing. Below we provide a fragment of the resulting Software Requirements Document produced as the result of applying this step of the methodology.

Environment contains: boat, house, well, fire, spear, fish, river, fishing basket.

Ontology Relationships. River has fishes, baskets contain fishes, men own boats.

Social goals. Daily provision of food to cater for the city's needs.

Abstract norms. Fishermen must go daily fishing for a limited number of hours; The number of fishing baskets per household is limited to avoid overexploitation of natural resources; Women are in charge of housework; Men alone cannot go fishing; Fishing is not allowed at night; Hoarding is prohibited and punished; All men under some age are obliged to fish;

Social model. Fishermen are in static separation of duties with wives.

Performative structure. Fishermen wake up, chat, fish, eat, sleep, and back again. Wives wake up, set up a fire, collect water, cook, sleep, and back again.

Interaction model. Description of fishing. At least 2 men on a boat. When both men on a boat, its state changes to "sailing". While sailing the fishermen holding a spear can throw it to catch fishes. If a spear gets a fish, the fisherman can remove the fish from the spear and put it in the fishing baske. The action of throwing the spear is an illocution, catching a fish is treated as an event generated by the spear. The rest treated as properties of roles, spear, and fishing baskets.

Concrete norms and constraints. Men are obliged to fish under the age of 35; one fishing basket per household max; fishing permitted between dawn and dusk; at least 2 men in a boat (this can be considered before); if hoarding (more than one fishing basket), fishermen can be prohibited to go sailing for a week.

Step 2: Specification. Based on the information obtained at Step 1 the formal specification of the underlying Electronic Institution was produced with ISLANDER [11]. Fig. 3 shows the Performative Structure of the specification. The nodes of this graph feature the identified scenes and the arcs define the role flow of participants amongst these scenes. Arcs labelled with "new" define that the participant with the role name appearing above the arc is initializing the scene and no other participants can enter it before the initialization occurs. The "bumpy" appearance of Home1 and Home2 suggests that these scenes permit multiple execution (which in our case corresponds to having different floors in the building, so that the agents can sleep inside their homes and on its roof).

Apart from the four roles (Fisherman1, Fisherman2, Wife1 and Wife2) that were identified at Step 1, the Performative Structure also includes the following roles (Fire, Boat, House1, House2, Well). These roles correspond to dynamic objects that can change the state of the environment by performing some actions in it. The interaction of the agents with such objects must be formalized appropriately in the specification of the institution to ensure correct behavior.

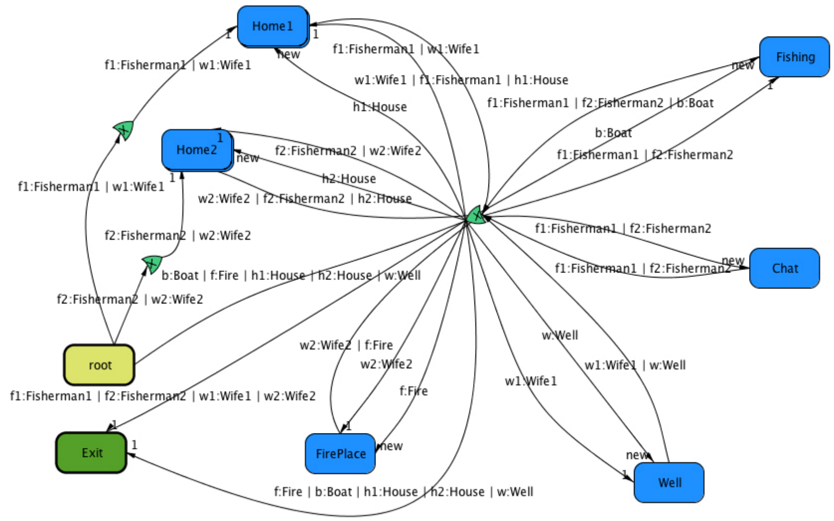

Fig. 3. Uruk Performative Structure.

The "root" and "exit" scenes are not associated with any patterns of behavior and simply define the state of entrance and exit of participants into the institution. Apart from "root" and "exit" each of the scenes present in the Performative Structure is associated with a Finite State Machine defining the interaction protocol for the participants that are accepted into the scene. In order to change the scene state the participant has to perform an illocutionary act (by sending a message to the institutional infrastructure). To give an example of the scene formalization Fig. 4 outlines the scene protocol for the Fishing scene and outlines the illocutions that change the states of this scene.

The scene protocol here defines in which sequence agents must perform the actions, at which point can they join and leave the scene and what should they do to change the scene state. In Virtual Institutions we consider every action that changes the state of the institution being a speech act (text message). Every action (i.e. grabbing an object or clicking on it) a participant performs in a Virtual World is automatically transformed into a speech act similar to those presented in the lower part of Fig. 4.

Once the scene is initialized its initial state becomes "W0". While the scene is in this state Fisherman2 and Boat can join the scene and both Fisherman1 and Fisherman2 can leave the scene (the "Boat" joins the scene automatically once it is activated). Fisherman1 can only enter the scene when it evolves to state "W1" as the result of Fisherman2 informing all the scene participants that he has successfully entered the boat. This happens once the corresponding avatar boards the boat. After entering the boat Fisherman1 must notify the participants about this fact so that the scene can evolve to "W2". For this to happen the corresponding agent must send the "f1:enterBoat" illocution to Fisherman2 and Boat. Then Fisherman1 may request to start the fishing, which would bring the scene into "W3". The result of this is the change of the boat property from "standing" to "afloat", Fisherman2 will start rowing and the boat object will move. While in "W3" the only action that can be done is informing all the

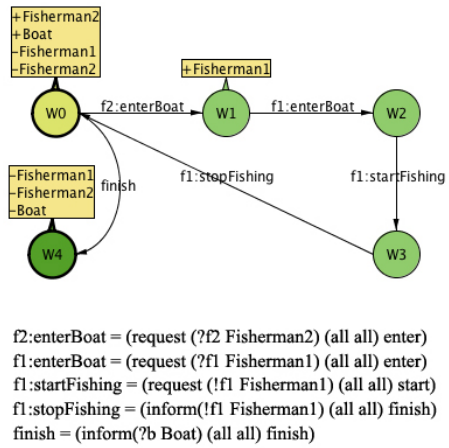

Fig. 4. Fishing Scene: Interaction Protocol.

participants by Fisherman1 that fishing is finished. When the fishing is finished Fisherman2 must return the boat to the initial poistion, park it there, drop the paddles, take the fishing basket and exit the boat. Fisherman1 will also have to exit the boat. No participants can leave the scene in this state and must wait until the scene evolves to "W0". While the scene is in "W0" again the Boat object may stop the scene by sending the "finish" illocution. Doing so would mean deactivating the scene and making it impossible for the participants to join it and act on it (no participant will be able to sit inside the boat).

Similar to the Fishing scene the interaction protocols have to be specified for other scenes present in the Performative Structure. We would like to point out that the scene protocol does not define how the actual fishing should take place, but simply provides the key states within the scene so that the agents can have a forma understanding of the performed actions.

Step 3: Verification. This step provides the possibility to ensure that the resulting specification is correct. ISLANDER provides an automatic way of verifying the correctness of the specification as well as the error notification system. Until the specification is error free it must be further revised in ISLANDER.

Step 4. Automatic Generation. In our case one of the system requirements was having a historical accurate reconstruction of the city based on the results of archaeological excavations. For such a case automatic generation is not appropriate and, therefore, Step 4 and 5 have been skipped. The 3D design of the city of Uruk was created manually under supervision of subject matter experts.

Step 6. Integration. At this step the dynamic objects are supplied with the corresponding LSL (Linden Scripting Language) scripts to enable interaction dynamics. In order to be able to maintain the causal connection between the institutional state and the state of the Virtual World the actions that change the state of the Virtual World are mapped to illocutions that change the institutional state. Programming the agents so that they can act in the environment and use the institutional specification for the decision making is also done at this step.

Step 7. Enabling Implicit Training. This functionality is currently missing in the described prototype. In the future this step will be used to let the agents learn state transitions from human experts controlling the avatar.

4 Conclusion

We have shown that 3D Virtual Worlds that involve complex interactions of participants and may include both humans and autonomous agents should be treated as Normative Multiagent Systems. For design and deployment of such Virtual Worlds we developed the Virtual Institutions Methodology. This methodology is supplied with a set of tools that facilitate the design, development and execution of such environments. We would like to stress that, to our knowledge, Virtual Institutions is the first methodology that is specifically concerned with the developments of Virtual Worlds with normative regulations of interactions. To evaluate the methodology we have applied it to the development of the Uruk prototype visualizing the behavior of citizens of ancient Mesopotamia, 3000 B.C.

References

1. Linden Lab: Second Life. http://secondlife.com
2. Gartner, Inc.: Gartner Says 80 Percent of Active Internet Users Will Have A "Second Life" in the Virtual World by the End of 2011, visited 23.07.2007. http://www.gartner.com/it/page.jsp?id=503861 (2007)
3. Hunter, D., Lastowka, F.G.: To kill an avatar. Legal Affairs (July 2003)
4. Weinstein, J., Myers, J.: Same principles apply to virtual world expansion as to China and other new markets. Media Village (November 28th) (2006. Available at http://www.mediavillage.com/jmr/2006/11/28/jmr-11-28-06/)
5. Mam, J., Haegler, S., Yersin, B., Mller, P., Thalmann, Daniel,, Van Gool, L.: Populating Ancient Pompeii with Crowds of Virtual Romans. In: 8th International Symposium on Virtual Reality, Archeology and Cultural Heritage - VAST. (2007)
6. Bogdanovych, A.: Virtual Institutions. PhD thesis, UTS, Australia (2007)
7. Russel, S., Norvig, P.: Artificial Intelligence a Modern Approach. Prentice Hall. Pearson Education International, Upper Saddle River, New Jersey, USA (2003)
8. Dennett, D.C.: Cognitive wheels: The frame problem of AI. In: Minds, Machines, and Evolution: Philosophical Studies. Cambridge University Press (1984)
9. Esteva, M.: Electronic Institutions: From Specification to Development. PhD thesis, Institut d'Investigació en Intelligència Artificial (IIIA), Spain (2003)
10. Bogdanovych, A., Esteva, M., Simoff, S., Sierra, C., Berger, H.: A methodology for 3D Electronic Institutions. In: Proceedings of AAMAS'07 Conference, Honolulu, Hawaii, USA, ACM (2007) 346–348
11. Esteva, M., de la Cruz, D., Sierra, C.: ISLANDER: an Electronic Institutions editor. In: First International Conference on Autonomous Agents and Multiagent systems, Bologna, ACM Press (July 2002) 1045–1052
12. Drago, S., Bogdanovych, A., Ancona, M., Simoff, S., Berger, H., Sierra, C.: From Graphs to Euclidean Virtual Worlds: Visualization of 3D Electronic Institutions. In Dobbie, G., ed.: Australasian Computer Science Conference (ACSC2007). Volume 62 of CRPIT., Ballarat Australia, ACS (2007) 25–33
13. Bogdanovych, A., Simoff, S., Esteva, M.: Virtual institutions: Normative environments facilitating imitation learning in virtual agents. In: proceedings of the 8-th International Conference on Intelligent Virtual Agents (IVA 2008), Lecture Notes for Computer Science, Berlin/Heidelberg, Springer (2008) 456–464
14. Esteva, M., Rosell, B., Rodriguez-Aguilar, J.A., Arcos, J.L.: AMELI: An Agent-Based Middleware for Electronic Institutions. In: Proceedings of AAMAS'04 Conference, Los Alamitos, CA, USA, IEEE Computer Society (2004) 236–243