An Ontology For Supporting The Evolution of Virtual Reality Scenarios

Mauro Dragoni[1], Chiara Ghidini[1], Paolo Busetta[2], Mauro Fruet[2], and Matteo Pedrotti[2]

[1] FBK–IRST, Trento, Italy
[2] Delta Informatica, Trento, Italy dragoni|ghidini@fbk.eu
paolo.busetta|mauro.fruet|matteo.pedrotti@deltainformatica.eu

Abstract. Serious games with 3D interfaces are Virtual Reality (VR) systems that are becoming common for the training of military and emergency teams. A platform for the development of serious games should allow the addition of semantics to the virtual environment and the modularization of the artificial intelligence controlling the behaviors of non-playing characters in order to support a productive end-user development environment. In this paper, we report the ontology design activity performed in the context of the PRESTO project aiming to realize a conceptual model able to abstract the developers from the graphical and geometrical properties of the entities in the virtual reality, as well as the behavioral models associated to the non-playing characters.

1 Introduction

Serious games with 3D interfaces are a branch of VR systems and are often used for the training of military personnel (in individual as well as team coordination danger situations) and, more recently, for the training of civilian professionals (firefighters, medical personnel, etc.) in emergency situations using tools such as VBS3[3] and XVR[4].

A crucial step towards the adoption of VR for training is the ability to configure scenarios for a specific training session at reduced costs and complexity. By looking at state of the art technologies, it is already possible to do so for physical landscapes, physical phenomena, and crowds (including their behaviors), and trainers and system integrators can assemble and customize serious game products for a specific scenario using commercial products and libraries that need to be (easily) adapted to the specific landscapes and needs of the clients.

Current attempts to the programming of non playing characters rely on ad hoc specifications/implementations of their behaviors done by VR developers. Thus, a specific behavior (e.g., a function emulating a panicking reaction) is hardwired to a specific item (e.g., the element "Caucasian_boy_17" of a VR such as XVR) directly in the code. This generates a number of problems typical of ad hoc, low level solutions: the solution is scarcely reusable, it often depends on the specific knowledge of the code of a specific developer, and is cumbersome to modify, since every change required by the trainer has to be communicated to the developers and directly implemented in the code in a case by

[3] https://www.bisimulations.com/
[4] http://futureshield.com/xvr-esemble.shtml

case manner. The existence of high level specifications of non playing characters and modular behaviors, described in a manner that is independent from the specific VR, and available for both trainers and developers, would be an important step towards the definition of reusable, flexible, and therefore cheaper, scenarios that include non playing characters.

In this paper, we focus on the experience of using Semantic Web techniques, and in particular lightweight ontologies, for the high level description of the artificial entities (including characters) and their behaviors in gaming in order to uncouple the description of scenarios performed by the trainers from their physical implementation in charge to the developers. Differently from a number of works in literature that often uses ontologies for a detailed description of the geometrical properties of space and objects, the focus of our work is on the description of the entities of a VR scenario from the cognitive point of views of the trainers and the developers alike, in a way that is semantically well founded and independent of a specific game or scenario [1], and with the goal of fostering clarity, reuse, and mutual understanding [2].

To the best of our knowledge, the construction of the ontology presented in this paper provides a first experience towards the description of a virtual world from a cognitive level that can highlight the potential and criticality of using Semantic Web techniques, and existing ontologies, to describe a VR from a cognitive point of view and can provide the basis for further developments.

2 The PRESTO Project

The objective of PRESTO (Plausible Representation of Emergency Scenarios for Training Operations) research project is the creation of a system for the customization of serious games scenarios based on virtual reality. The advantage of this system, compared to the state of the art, resides in the richness and the ease of defining the behavior of artificial characters in simulated scenarios, and on the execution engines able to manage cognitive behaviors, actions, and perceptions within a virtual reality environment. One of the main outcome of the project is the possibility of specifying procedures, psychological profiles, and other factors that influence the behavior of individuals and/or small groups in any role (emergency teams, victims, observers, terrorists, criminals, etc.) and to build scenarios, for instance a car accident, in which part or all of the people involved are simulated by artificial characters. To this end, the system has to include an environment for building the training scenarios by the VR trainer, tools for the specification of cognitive and perceptual models used for augmenting psychological profiles of non-player characters, and execution engines able to manage cognitive behaviors, actions, and perceptions within a virtual reality environment.

The system can be used, for example, for training safety personnel, for the verification and the optimization of operational procedures, and for the analysis of work environments. The system has been tested in a pilot use case selected in a specific application domain of large interest in both commercial and research fields: training for emergency management within close environments (such as fires, evacuations, overload of users due to external factors such great disasters scale, etc.). The pilot has been be conducted in collaboration with the Health Services of the Trentino local government (APSS).

The open problems addressed by this project may be summarized as follows:

1. the perception of the virtual environment by an artificial character and the execution of its models and procedures must be able to adapt to the context, to its history and status (fatigue, emotions, intake of stimulants such as caffeine or depressants such as alcohol) and must maintain a level of variability (i.e. in the accuracy of the vision, the rate of reaction, in the choices among alternatives) such that the behavior is plausible but not trivially predictable;
2. the representation of procedures and patterns of behavior must be independent of one specific usage scenario and accessible to training specialists (i.e. industrial safety or civil protection) rather than just a computer, in an environment facilitating the definition and configuration of training scenarios by such specialists.

The first open problem relates to aspects such as the usage of a BDI (Beliefs-Desire-Intention) multi-agent system with cognitive extensions, CoJACK [3], as the artificial intelligent engine for the generation/selection of behaviors in serious games [4], that go beyond the scope of this paper.

What we present in this work, instead, is the experience of using Semantic Web techniques, and in particular lightweight ontologies, to contribute to the second open problem, that is the development of a programming environment for serious game platforms thanks to end-user development tools [5] and the ability to mix and match scenario components (including behavioral components) taken off-the-shelf from a market place.

3 PRESTO Ontology Design

The development of programming environment for the high level description of artificial entities (including characters) and their behaviors in scenarios of serious games requires the ability to represent a wide range of entities that *exist* in the (artificial) world. The approach taken in PRESTO is to use ontologies to represent this knowledge, in a way that is semantically well specified and independent of a specific game or scenario [1].

The construction of the PRESTO ontology therefore is driven by typical questions that arise when building ontological representations of a domain, that is:

– "What are the entities that exist, or can be said to exist, in a Virtual Reality scenario?"
– "How can such entities be grouped, related within a hierarchy, and subdivided according to similarities and differences?"

Differently from Ontology in philosophy, where these questions are motivated from the need to investigate the nature and essence of being, we have looked at these questions from the pragmatic point of view of computer science, where ontologies and taxonomic representations have been widely proposed and used to provide important conceptual modeling tools for a range of technologies, such as database schemas, knowledge-based systems, and semantic lexicons [2] with the aim of fostering clarity, reuse, and mutual understanding.

A serious problem we had to face in PRESTO was the lack-of/limited-availability of training experts and software developers, and the broad scope of items and behaviors that can occur in an arbitrary scenario of VR, that can range from terrorist attacks in a war zone, to a road accidents in a motorway, to a fire alarm in a nuclear plant or hospital and so on. Because of that reason, building everything from the ground up by relying on domain experts and using one of the state of the art ontology engineering methodologies such as METHONTOLOGY [6] was deemed unfeasible. Thus the process followed in PRESTO has been driven by an attempt to: (1) maximize the reuse of already existing knowledge and (2) revise and select this knowledge with the help of experts by means of more traditional ontology engineering approaches such as the one mentioned above. The choice of already existing knowledge has lead us to consider the following two sources:

- state of the art foundational ontologies which provide a first ontological characterization of the entities that exist in the (VR) world; and
- the concrete items (such as people, tools, vehicles, and so on) that come with virtual reality environments and can be used to populate scenarios.

Our choices for the PRESTO project were the upper level ontology DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering) [7], and the classification of elements provided by XVR. DOLCE was chosen as this ontology not only provides one of the most known upper level ontologies in literature but it is also built with a strong cognitive bias, as it takes into account the ontological categories that underlie natural language and human common sense. This cognitive perspective was considered appropriate for the description of an artificial world that needs to be plausible from a human perspective. The decision to use the classification of elements provided by XVR was due to the extensive range of item available in their libraries (approximatively one thousand elements describing mainly human characters, vehicles, road related elements, and artifacts like parts of buildings) and the popularity of XVR as virtual reality platform.

The construction of the first version of the ontology of PRESTO was therefore performed by following a middle-out approach, which combined the reuse and adaptation of the conceptual characterization of top-level entities provided by DOLCE and the description of extremely concrete entities provided by the XVR environment. More in detail,

- we performed an analysis and review of the conceptual entities contained in DOLCE-lite [7] together with the Virtual Reality experts (both trainers and developers) and selected the ones referring to concepts than needed to be described in a VR scenario; this analysis has originated the top part of the PRESTO ontology described in Section 4.1.
- we performed a similar analysis and review of the XVR items, together with their classifications, in order to select general concepts (e.g, vehicle, building, and so on) that refer to general VR scenarios; this analysis has originated the middle part of the PRESTO ontology described in Section 4.2.

- as a third step we have injected (mapped) the specific XVR items into the ontology, thus linking the domain independent, virtual reality platform independent ontology to the specific libraries of a specific platform, as described in Section 4.3.

A reader could ask now why we didn't simply/mainly rely on the XVR classification in order to produce the, so called, PRESTO ontology. The reason is twofold: first of all, the XVR classification mainly concerns with objects. It provides therefore a good source of knowledge for entities "that are" (in DOLCE called Endurants), but a more limited source of knowledge on entities "that happen" (in DOLCE called Perdurants). Second, the XVR libraries contain objects described at an extremely detailed level whose encoding and classification resembles more to a Directory structures built to facilitate the selection of libraries rather than a well thought is-a hierarchy and therefore presents a number of problems that prevent its usage 'as such'. In the following, we review the most common problems we found in the categorization of the XVR items:

- Concepts names are used to encode different types of information. For instance the concept name "Caucasian_male_in_suit_34" is used to identify a person of Caucasian race, dressed in suit and of 34 years of age. Encoding the information on race, age, and so on via e.g., appropriate roles enables the definition of classes such as e.g., "Caucasian_person", "young adult", "male" and so on and the automatic classification (and retrieval) of XVR item via reasoning.
- The terminology used to describe concepts is not always informative enough: for instance, it is difficult to understand the meaning of the entity "HLO_assistant" from its label and description and to understand whether this item may suggest a type of "assistant" that may be useful in several scenarios and could therefore be worth adding to the ontology.
- The level of abstraction at which elements are described varies greatly. For instance the library containing police personnel items classifies, an the same hierarchical level the general concept of "Police_Officer" and the rather specific concept of "Sniper_green_camouflage".
- the criteria for the classification is not always clear: for instance, the "BTP_officer" (British Transport Police) concept is not a subclass of "Police_Officer".
- Certain general criteria of classification are not present in all the libraries. As an example, the general concept "Adult_Male" should be a general concept used for the classification of male characters. Nonetheless, it is present in e.g, the library of "Environment_humans" (that is, the library that describes generic characters) and is not present in e.g., the libraries of "Rescue_humans" and "Victims" (that is, the libraries of characters impersonating rescuers and victims, respectively).
- Unclear classification: for instance, in the XVR original classification a "sign" is a "road_object", and a "danger_sign" is an "incident_object". By considering that no relations are defined between the entities "sign" and "danger_sign", it is not possible to infer any relation between "danger_sign" and "road_object".
- Duplication of concept names: for instance, the label "police_services" is used to describe both human police characters in the library "environment_human", and police vehicles, in the library "rescue_vehicle".

In the next section we provide an overview of the PRESTO ontology and of its top-level, middle level and XVR specific components in detail.

4 The PRESTO ontology

As introduced in Section 3, the PRESTO ontology [5] is composed of three parts: (i) a top level part constructed with the help of DOLCE; (ii) a middle level describing general entities that can occur in a VR scenario, and (iii) a specific set of entities representing objects and "behaviors" available in a concrete VR.

4.1 The Top-level Ontology: DOLCE Entities

Figure 1 shows the taxonomy of DOLCE entities taken from [7] revised and customised to the needs of PRESTO.

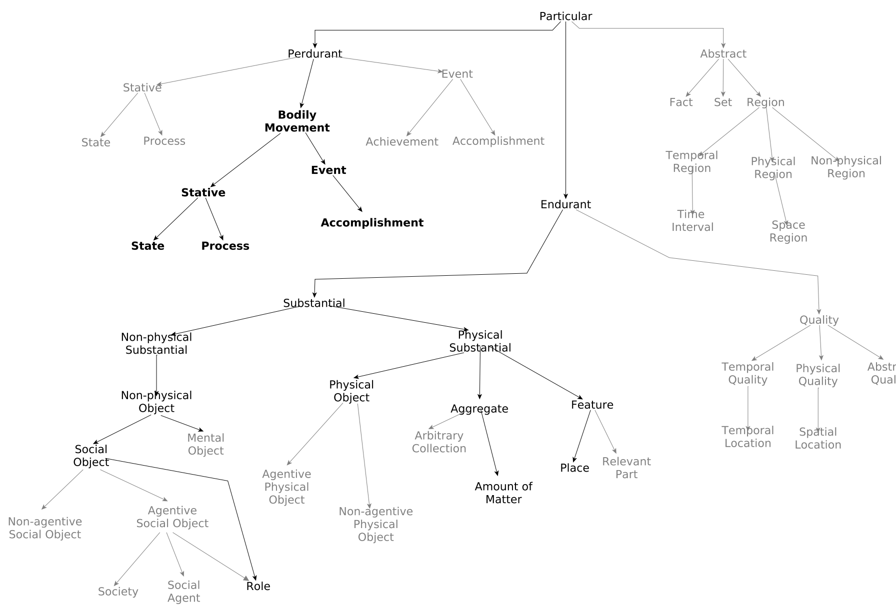

Fig. 1: The top-level PRESTO ontology.

Entities in gray where not included in the PRESTO ontology, while entities in boldface where added specifically for PRESTO.

Among the first level of entities we selected **Endurants** and **Perdurants**: endurants are indeed useful to describe the big number of physical and non-physical objects that can occur in a serious game, including avatars, vehicles, tools, animals, roles and so on; perdurants are instead useful to describe what happens in a scenario. Concerning endurants the diagram in Figure 1 shows the ones we selected to be included in PRESTO;

[5] The current version of the PRESTO ontology cannot be published due to copyrights constraints. A preliminary version, from which it is possible to observe the rational used for modeling it, may be found here: https://shell-static.fbk.eu/resources/ontologies/CorePresto.owl

note that we did not include the distinction between agentive and non-agentive physical objects because of an explicit requirement by the PRESTO developers. In fact, they require the possibility to treat every object in a VR as an agentive one for the sake of simplicity[6]. While perdurants can be useful in a VR to describe a broad set of "things that happen", in the current version of the ontology they were mainly used to describe animations (that is, "bodily_movements") of avatars. From an ontological point of view we felt it was appropriate classify them according to the categories of stative and eventful perdurants included in DOLCE. In fact, we can have *state* bodily movements (e.g., being sitting), *process* bodily movements (e.g., running), and *accomplishment* bodily movement (e.g., open a door). The investigation of animations did not show examples of *achievement* bodily movements, which were therefore not included in the ontology.

The current version of the ontology does not contain **Qualities**, but current work (not described in this paper) is devoted to investigate how to include them in a further revision. Instead **Abstracts** do not seem to play a role in the PRESTO ontology.

4.2 The Middle-level Domain Ontology

This part augments the top level ontology described above with concrete, but still abstract, entities that may appear in a broad range of virtual reality scenarios for serious games. The current version of the ontology is composed of 311 concepts, 5 object properties and 3 annotations properties. Concerning the Endurant part the main entities modeled in the middle-level ontology pertain classifications of persons (avatars), buildings, locations, tools / devices, vehicles, and roles. Concerning perdurants the ontology contains concepts describing *state*, *process* and *accomplishment* bodily movement.

4.3 Injecting The Bottom-level Ontology

The linking of the bottom-level ontology, representing the classification scheme used for organizing the items contained in the 3D-library, is not a trivial task. Indeed, the correct alignment of these levels enables the transparency of the system with respect to the actual content of the 3D-library.

While the creation of the top and middle-level of the PRESTO ontology is meant to create a stable knowledge source, the definition of the alignments with the bottom-level elements is an activity that has to be done every time a new 3D-library is plugged into the system.

To ease this injection we decided to accomplish it in two separate steps: (i) an automatic definition of alignments by using an ontology alignment tool and (ii) a manual refinement of the alignments before using the complete ontology in the production stage.

The output of the alignment task is the linking between the abstract concepts contained in the middle PRESTO ontology and the concrete items contained in the underlying 3D-library implemented in the system. Indeed, such alignments allow the access

[6] A typical example is vehicle, which the developers prefer to treat as an agentive objective, rather than a non agentive object driven by an agent, for the sake of simplicity of the code.

to the entire set of items defined in the 3D-library and that are physically used for building the virtual reality scenario.

For sake of clarification about the alignment process works, let's consider the following example. In the middle-level of the ontology we have defined the concept "Tent" representing a general tent that may be used for building a virtual reality scenario. By plugging, for example, the XVR library, we need to find an alignment between the entity "Tent" and the specific tent items contained in XVR, such as "Decontamination_Tent_Zone_1", "Family_tent_blue", "Treatment_Area", and so on. To do that, as first step, we execute the Alignment API library [8]: for the entity "Tent", the XVR item identified in the 3D-library and aligned with it is "Tents". Such an alignment, classifies the bottom-level ontology "Decontamination_Tent_Zone_1", "Decontamination_Tent_Zone_2", "Decontamination_Tent_Zone_3", "Family_tent_blue", "Family_tent_orange", "Festival_tent", and "Treatment_Area" as children of the concept "Tents". As a consequence, all these elements can be retrieved and used at run time to produce a specific scenario which requires the presence of a tent, while the scenario can still be described using the abstract term "tent". Also, the same high level scenario may be easily adapted to the usage of other 3D-libraries, simply by exploiting the (different) mappings of such libraries with the middle level "Tents" concept.

In some cases the automatic alignment we used fails: for example, the middle-level entity "Weapon" is automatically aligned with the bottom-level entity "Baton" instead of being aligned with the bottom-level entity "Service-weapon". In these cases, a manual refinement of the generated alignments was done afterwards for pruning wrong axioms.

By considering the XVR use case, the automatic alignment procedure allowed a time-effort reduction, with respect of doing everything manually, of around 65% in the definition of the alignment between the middle-level and the bottom-level ontologies, thus showing the potential of using ontology mapping technologies in the concrete scenario of virtual reality libraries.

5 Enriching the VR for decision-making and coordination

There are a number of aspects required for decision-making and coordination of activities that cannot be fully captured via static taxonomies and aggregations but are worth describing in an ontology not only for its inherent representational and deductive power, which helps in structuring abstract reasoning, but for the ability built into PRESTO of dynamically and arbitrarily add and remove tags to any item within the VR. These tags are generically called "qualities" since they are mostly described as **Qualities** entities in the PRESTO ontology. They form a layer of knowledge shared by all PRESTO components (including configurator systems, DICE (an agent framework) agents, monitor and control GUIs, and end-user development tools) without the need of modifying the game engine or hard-coding relationships among categories and properties into software. Note that this layer could have been built into the ontology itself (technically, by representing all items in the VR as individuals stored in a triple store) but this would have created issues with distribution, deployment and performance, so it is managed differently. Further, DICE supports the tagging of BDI (Belief-Desire-Intention) plans and intentions by software developers; these tags can be used for introspection and monitoring of the activity of an agent.

Qualities are still work in progress, since they reflect the progressive development of behavioral models. At the moment, they are used for two main reasons: to represent an item's characteristics and dynamic state; and, to enable recognition (of activities and intentions) and coordination.

Examples of characteristics and states represented as qualities include:

- the characteristic of being a "gate", which indicates something that can be crossed but only after performing some enabling actions if required and coordinating with others, thus it is relevant to the models of navigation. A gate may be the revolving door at the entrance of a room, the sliding door of a lift, a driveway gate, a railroad crossing, and so on, all of which may have been classified very differently in the VR. Note that a permanently sealed door is not a gate in this definition;
- the dynamic state of being "open", which may be associated to gates (as above) as well as to entities not relevant to navigation (e.g. windows). Stative qualities are represented as a is-a hierarchy, whose root is a generic name (such as "openness") and whose children are the possible values of the quality (in this example, open, close, semi-open, semi-close, etc.). Items are tagged with the leaves (e.g., open or close) but the PRESTO API allows querying the current state by using the root, thus implicitly checking if the item does have that quality in the first place. Other examples of wide applicability include "liveliness" (which includes "alive", "dead", "impaired") and "functioning" (specialized in "running" and "stopped");
- dynamic states such as "body posture" and "facial expression", also organized in hierarchies as mentioned above. While posture and expression apparently are properties of humans only, they can be also applied to animals and even to non-living entities; for instance, in shooting ranges (and their VR reconstructions), puppets used as targets may have different postures;
- dynamically changing values of various nature. PRESTO allows the association of an arbitrary content together with a tag to an item, thus this mechanism is essentially a way to add data fields to an object without impacting the general PRESTO API. For instance, the reward mechanism in a Unity game built for instructional purposes has been implemented as a "money"-tagged accumulator on a specific item.

As mentioned earlier, the PRESTO ontology classifies also the animations that can be applied by a game engine to entities. While this classification is used at the moment as a configuration tool, essentially to make DICE models agnostic with respect to the underlying technology, it is the first step towards a solution to the problem of intention recognition, which in turn is the base for the simulation of coordinated behaviour (no matter whether amicable, e.g. teamwork as fire fighters in the fire example presented earlier, hostile, e.g. opposition in a security scenario, or simply observation to anticipate future moves and take decisions, e.g. avoiding a safety exit door when too many people are engaging it during an alarm). Intention recognition is something that is innate in humans and cognitively complex animals (e.g. dogs) but computationally very hard if taken by principle; machine learning may come to the rescue in certain situations, but in a VR scenario where nuances of body and expressions are hard to capture and represent, let alone the limited number of training cases, this is not an option. In PRESTO, qualities are exploited to allow entities to make their recognizable activities publicly visible; thus, intention and action recognition is reduced to reading certain qualities automatically set by DICE when starting animations or appropriately tagged plans.

To do a further step ahead, work is in progress on game-theoretical descriptions of coordinated behavior, including queuing and other crowding behaviors, accessing shared resources, and so on, in order to enable the definition of policies at a very abstract (meta-) level. This work exploits, in addition to PRESTO's tagging of items, the equivalent in DICE for goals and plans as well as its support for introspection of intentions and motivations. In a nutshell, DICE agents tag themselves and any involved object with qualities that indicate the move they want to play in a coordination game, while their meta-level, cognitive models would try to achieve or stop pursuing aptly tagged goals and plans according to the agent's own moves in the game as well as of those entities perceived in the environment. The specification of policies is expected to substantially reduce the coding required by models and to allow the reuse of the same coordination patterns in many different situations, e.g. a single policy for queuing to pass through a gate (which will be part of the navigation models) as well as for queuing at the entrance of an office or at the cashier in a supermarket (which are decision-making behaviors not related to navigation goals).

A simplistic (but already available and of great practical use) coordinated behavior exploiting qualities is goal delegation from an agent to another agent. By means of the PRESTO API, any entity in a game can submit a goal to be pursued by any other entity; when the goal is enriched with a few predefined parameters, the destination DICE agent publishes the fact that it has accepted a goal or that has achieved it (or failed to achieve or refused), allowing the submitter (or any other observer, including PRESTO's session script engine) to monitor and coordinate behaviors without the use of any additional agent protocol.

In the PRESTO ontology, qualities are represented as endurant or perdurant, depending on their lifetime – static characteristics are endurant while stative, behavioral and coordination qualities are perdurant.

As a final note, it is worth mentioning that PRESTO uses ontologies, in addition to classifications and qualities as discussed above, for other purposes such as:

- to represent individual, rather than objective, perspectives on the world. Currently, an ontology is used to capture the possible values used by DICE models to appraise entities that may have an influence on behaviours. These values range from positive to negative at different levels, from "friendly" to "dangerous, to stay distant from". For reasons similar to those that led to the management of qualities in PRESTO, the relationships between ontological classifications and appraisal values are captured by configuration files at various level of granularity (shared by all NPCs of a certain type rather than specific for an individual) rather than within the ontology;
- software engineering practice, e.g. to allow the definition of certain APIs in a language-independent format, with the automatic generation of software in some cases, and similarly for independence from the game engine when accessing com-

monly available resource types (e.g. animations, as mentioned above) by means of an engine-neutral syntax.

6 Related Work And Conclusion

In this paper, we focused on the experience of using Semantic Web techniques, and in particular lightweight ontologies, for the description of the artificial entities and their behaviors in gaming with the aim of uncoupling the description of virtual reality scenarios from their physical implementation in charge to the developers.

With respect to the literature, where ontologies are often used for a detailed description of the geometrical properties of space and objects [9], we focused more on how the description of the entities of a VR scenario can be easily represented and managed from the practical point of view. Indeed, the literature addressed such problems only marginally by focusing mainly on the use of ontologies for managing the representation of virtual reality scenarios themselves [10,11], even if in some cases a clear target domain, like the management of information related to disasters [12], is took into account. Also the description of character behaviors have been supported by using ontologies for different purposes like as support for UML-based descriptions [13] or as a "core" set of structural behavioral concepts for describing BDI-MAS architectures [14].

However, all these works do not take into account issues concerning the practical implementations of flexible systems for building virtual reality scenarios. The proposed solution demonstrated the viability of using Semantic Web technologies for abstracting the development of virtual reality scenarios either from the point of view of the 3D-design and from the modeling of character behaviors.

References

1. Gruber, T.R.: Toward principles for the design of ontologies used for knowledge sharing? International Journal of Human-Computer Studies **43**(5–6) (1995) 907 – 928
2. Guarino, N., Welty, C.: Identity and subsumption. In: The Semantics of Relationships: an Interdisciplinary Perspective, Kluwer (2001)
3. Ritter, F.E., Bittner, J.L., Kase, S.E., Evertsz, R., Pedrotti, M., Busetta, P.: CoJACK: A high-level cognitive architecture with demonstrations of moderators, variability, and implications for situation awareness. Biologically Inspired Cognitive Architectures **1** (July 2012) 2–13
4. Evertsz, R., Pedrotti, M., Busetta, P., Acar, H., Ritter, F.: Populating VBS2 with Realistic Virtual Actors. In: Conference on Behavior Representation in Modeling & Simulation (BRIMS), Sundance Resort, Utah (March 30 – April 2 2009)
5. Paternò, F.: End User Development: Survey of an Emerging Field for Empowering People. ISRN Software Engineering **2013** (2013) 1–11
6. Fernández-López, M., Gómez-Pérez, A., Juristo, N.: Methontology: from ontological art towards ontological engineering. In: Proc. Symposium on Ontological Engineering of AAAI. (1997)
7. Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., Schneider, L.: Sweetening ontologies with dolce. In: Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web, Springer-Verlag (2002) 166–181

8. David, J., Euzenat, J., Scharffe, F., dos Santos, C.T.: The alignment API 4.0. Semantic Web **2**(1) (2011) 3–10
9. Chu, Y.L., Li, T.Y.: Realizing semantic virtual environments with ontology and pluggable procedures, applications of virtual reality. InTech (2012)
10. Bille, W., Pellens, B., Kleinermann, F., Troyer, O.D.: Intelligent modelling of virtual worlds using domain ontologies. In Delgado-Mata, C., Ibáñez, J., eds.: Intelligent Virtual Environments and Virtual Agents, Proceedings of the IVEVA 2004 Workshop, ITESM Campus Ciudad de Mexico, Mexico City, D.F , Mexico, April 27th 2004. Volume 97 of CEUR Workshop Proceedings., CEUR-WS.org (2004)
11. Xuesong, W., Mingquan, Z., Yachun, F.: Building vr learning environment: An ontology based approach. In: Education Technology and Computer Science, 2009. ETCS '09. First International Workshop on. Volume 3. (March 2009) 160–165
12. Babitski, G., Probst, F., Hoffmann, J., Oberle, D.: Ontology design for information integration in disaster management. In Fischer, S., Maehle, E., Reischuk, R., eds.: Informatik 2009: Im Focus das Leben, Beiträge der 39. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 28.9.-2.10.2009, Lübeck, Proceedings. Volume 154 of LNI., GI (2009) 3120–3134
13. Bock, C., Odell, J.: Ontological behavior modeling. Journal of Object Technology **10** (2011) 3: 1–36
14. Faulkner, S., Kolp, M.: Ontological basis for agent ADL. In Eder, J., Welzer, T., eds.: The 15th Conference on Advanced Information Systems Engineering (CAiSE '03), Klagenfurt/Velden, Austria, 16-20 June, 2003, CAiSE Forum, Short Paper Proceedings, Information Systems for a Connected Society. Volume 74 of CEUR Workshop Proceedings., CEUR-WS.org (2003)