Developing an Ontology for Autonomous Entities in a Virtual Reality: The PRESTO Experience

Paolo Busetta[1](✉), Mauro Fruet[1], Piero Consolati[1],
Mauro Dragoni[2], and Chiara Ghidini[2]

[1] Delta Informatica, Trento, Italy
{paolo.busetta,mauro.fruet,piero.consolati}@deltainformatica.eu
[2] FBK–IRST, Trento, Italy
{dragoni,ghidini}@fbk.eu

Abstract. The PRESTO project focuses on modeling the behaviour of humans and rational entities in general (e.g. animals, vehicles with a human driver), specifically to represent decision making driven by norms or doctrine as well as culture and emotional factors. PRESTO's models are used to drive NPCs (Non-Player Characters) in serious games, currently applied to emergency management and training in health environments (hospitals and such) even if the technology is not domain- nor game-specific.

A number of requirements have led to the adoption of ontologies as the main classification and annotation mechanism of both the external world and the internal states of an NPC. Structuring and building these ontologies have been done by adopting a mixed top-down and bottom-up approach. The main results are (1) a top-level ontology inspired by the well-known DOLCE; (2) the structuring of the ontologies in composable domain-specific and individual sections; (3) a tool for the semi-automatic extraction of categories from the available virtual reality assets.

This paper focuses on the ontology design, illustrated with a few practical examples.

Keywords: Virtual reality · Serious games · Ontologies

1 Introduction

Serious games with 3D interfaces are a branch of VR systems often used for the training of military personnel and, more recently, for the training of civilian professionals (firefighters, medical personnel, etc.) in emergency situations. Commonly used systems include VBS3 by Bohemia Interactive[1] and XVR by E-semble[2].

[1] https://www.bisimulations.com/.

[2] http://www.xvrsim.com.

J. Hodicky (Ed.): MESAS 2015, LNCS 9055, pp. 3–16, 2015.
DOI: 10.1007/978-3-319-22383-4_1

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. State of the art technologies typically enable the user to specify physical landscapes, physical phenomena, and crowds (including their behaviours). 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.

Not so advanced is the technology for enriching the scenarios with non-player characters (NPCs), that is, those characters (people, animals, vehicles, small teams, and so on) directly involved in game playing in collaboration with (or in opposition to) human players, but whose behaviour is entirely animated in an artificial manner. Here the problem is (at least) twofold. A first problem is the lack of configuration environments for trainers and system integrators to complement the "physical landscape" with descriptions of the non-player characters at a high-level, ideally in a way suitable to a non-programmer. As an example, such an environment would allow the configuration of a scenario for fire emergency training in a hospital ward that contains, in addition to the physical reconstruction of the ward building and of the fire, a set of non-player characters composed of three nurses, of which one expert and two novices, one doctor from another ward who is not familiar with the safety procedures of the ward, and eight patients among which a child and a blind patient; the configuration should specify the physiological and psychological characteristics of these NPCs and what is expected from them while a game session unfolds. A second problem is the lack of algorithms for the generation / selection of realistic and plausible behaviours for non-player characters, able to adapt themselves to the evolution of the game. Currently, the programming of NPCs mostly relies on ad hoc specifications / implementations of their behaviours done by game developers. Thus, a specific behaviour (e.g., a function emulating a panicking reaction) is hardwired to a specific item (e.g., the element "Caucasian_boy_17" in 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 case manner. While this is not perceived as a major issue in entertainment games (but economics and a push for better game experiences are changing this, too), in serious gaming the cost and complexity of ad-hoc development is only partially covered by available budgets. Thus, typical solutions to this problem include, in multi-player games, the recruitment of experts to impersonate characters (such as team mates, enemies, victims, injured people, and so on) or, as in XVR, letting the trainer changing the scenario in real-time by hand.

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 behaviours in gaming in order to uncouple the description of scenarios performed by training specialists by means of end-user tools from the implementation of behavioural models, in charge to software 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 [13], and with the goal of fostering clarity, reuse, and mutual understanding [14].

PRESTO is briefly introduced in the next section. Section 3 sketches the approach adopted to build its main ontology. Its result is the shared vocabulary presented in Sect. 4, grounded in the foundational ontology DOLCE, that helps in identifying the basic entities of a VR scenario, together with their mappings to items of a specific VR implementation (such as XVR). Section 5 briefly discusses a part of the ontology that is strictly related to decision-making and coordination, independently of the VR.

2 Overview of PRESTO

PRESTO (Plausible Representation of Emergency Scenarios for Training Operations) [5,6] aims at adding semantics to a virtual environment and modularising the artificial intelligence controlling the behaviours of NPCs. Its main goal is to support a productive end-user development environment directed to trainers building scenarios for serious games (in particular to simulate emergency situations such as road and industrial accidents, fires and so on) and in general to game masters wanting to customize and enrich the human player's experience. The framework for behavioural modeling in PRESTO, called DICE, was inspired by a BDI (Belief-Desire-Intention) [4,15] multi-agent system with cognitive extensions, CoJACK [9,16]. PRESTO offers powerful end-user development tools for defining the parts played by virtual actors (as end user-written behaviours) and the overall session script of a game. PRESTO supports a specific virtual reality, XVR from E-Semble, a well known tool in use for Emergency Management and Training (EMT) in a number of schools and organisations around the world, as well as Unity 3D and, at least in principle, is agnostic with respect to the game engine in use.

PRESTO provides three main mechanisms that enable the reuse and adaptation of behavioural models to different scenarios, games or even game engines: semantic facilities, an interpreter of scripts in DICE, and facilities for game session control.

The semantization of the game environment and of part of the cognitive states of an NPC supports decision-making based on game- and scenario-independent properties. To this end, ontologies are used for the classification of objects and locations, for annotating them with properties and states (called "qualities") that allow abstract reasoning and for the (agent-specific) appraisals of perceptions, in particular to deal with potentially dangerous situations. The use of ontologies in PRESTO has been partially discussed in [8] and will be the focus of the following sections.

The flow of perceptions, the properties of entities, the appraisal values of DICE behavioural models are classified by means of the PRESTO ontologies, thus enabling the development of generic BDI logic (goals, plans and beliefs) independent of the scenario of use. Additionally, DICE provides an interpreter for high-level scripts, called "DICE Parts", written by means of a graphical editor by the end-user (typically a trainer during the preparation of a specific scenario). A DICE Part can invoke multiple goals concurrently, terminate them when specific events happen (including timeouts and perceptions), define reactions to perceptions or to modifications of the internal state of the agent (including appraisals and moderators), change the state of the agents itself, and so on. While the DICE Part language is limited in its expressivity, the cost of producing a part is minuscule with respect to directly programming the underlying BDI logic. An effort is required on developers of behavioural models in BDI logic to provide goal-directed behaviours that are suitable for composition within user-written parts and adaptable to different scenarios thanks to semantic-based reasoning; the PRESTO pilot project and other demonstrators are helping in accumulating experience and defining guidelines.

Finally, PRESTO has an end-user facility to edit and control session-level scripts inspired by interactive books. A session script is composed of a set of scenes connected as a graph. At each scene, goals can be given to NPCs (which may trigger user-written parts), their internal state changed (including emotions) and objects manipulated. The trainer starts a script at the beginning of a training session and advances it by manually navigating the graph of scenes or letting PRESTO choose the next one e.g. when certain events happen or when a timer expires. This allows a large, potentially unlimited number of different sessions to unfold from a single script with no need to reprogram the NPCs once equipped with all required behavioural models and DICE Parts. In the hospital ward example presented earlier, the initial scene would command visitors, patients and nurses to accomplish their routine goals; the script may continue with alternative scenes such as "fire breaking in a patient room" or "fire breaking in a surgical facility", each with different people involved, and then with sequences that may lead e.g. to smoke filling the area and visitors fleeing or an orderly managed situation with the intervention of fire fighters, chosen according to the decisions of the trainer and the events occurring during a session.

3 PRESTO Ontology Design

The development of programming environment for the high level description of artificial entities (including characters) and their behaviours 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 [13].

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 [14] 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 behaviours that can occur in an arbitrary scenario of VR, that can range from terrorist attacks in a war zone, to 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 [11] 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 led 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) [12], and the classification of elements provided by XVR. DOLCE was chosen as this ontology since, in addition to providing one of the most known upper level ontologies in literature, it is 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 items available in their libraries (approximately 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. In more detail,

- We performed an analysis and review of the conceptual entities contained in DOLCE-lite [12] 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 Sect. 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 Sect. 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 Sect. 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 structure 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 items 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 the general concept of "Police_Officer" and the rather specific concept of "Sniper_green_camouflage" at the same hierarchical level.
- The criteria for the classification are 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 Sect. 3, the PRESTO ontology[3] 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 "behaviours" available in a concrete VR.

4.1 The Top-Level Ontology: DOLCE Entities

Figure 1 shows the taxonomy of DOLCE entities taken from [12] revised and customised to the needs of PRESTO.

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 Fig. 1 shows the ones we selected to be included in PRESTO; 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[4]. While perdurants can be useful in a VR to describe a broad set of "things that

[3] 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.

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

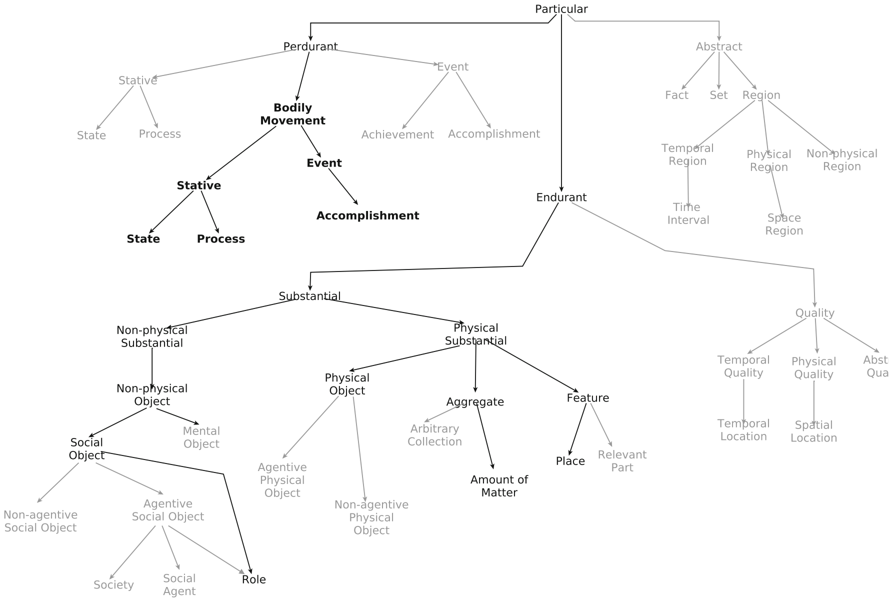

Fig. 1. The top-level PRESTO ontology

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 to 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.

We will discuss **Qualities** in Sect. 5. 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. An excerpt of the middle-level ontology can be seen in Fig. 2.

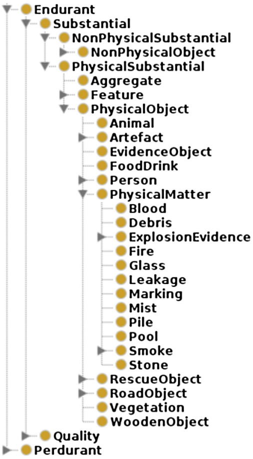

Fig. 2. The middle-level PRESTO ontology

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

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. More details on the alignment procedure, including cases in which the automatic step fails, can be found in [8].

5 Enriching the VR for Decision-Making and Coordination

The previous section illustrated a classification of the VR world in which NPCs act. However, 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 agents, monitor and control GUIs, and end-user development tools) without the need of modifying the game engine or hardcoding 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 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 behavioural 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 behaviour, including queuing and other crowding behaviours, 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 behaviours not related to navigation goals).

A simplistic (but already available and of great practical use) coordinated behaviour 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 behaviours 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, behavioural 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 commonly 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 behaviours 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 [7], 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 [2,17], even if in some cases a clear target domain, like the management of information related to disasters [1], is taken into account. The description of character behaviours has been supported by using ontologies for different purposes like as support for UML-based descriptions [3] or as a "core" set of structural behavioural concepts for describing BDI-MAS architectures [10].

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 behaviours.

References

1. 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. LNI, vol. 154, pp. 3120–3134. GI (2009). http://subs.emis.de/LNI/Proceedings/Proceedings154/article2822.html
2. 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, 27th April 2004. CEUR Workshop Proceedings, vol. 97. CEUR-WS.org (2004). http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-97/IVEVAFinal_S1_01.pdf
3. Bock, C., Odell, J.: Ontological behavior modeling. J. Object Technol. **10**(3), 1–36 (2011). http://dx.doi.org/10.5381/jot.2011.10.1.a3
4. Bratman, M.E.: Intention, Plans, and Practical Reason. Harvard University Press, Cambridge (1987)
5. Busetta, P., Ghidini, C., Pedrotti, M., Angeli, A.D., Menestrina, Z.: Briefing virtual actors: a first report on the presto project. In: Romano, D. (ed.) Proceedings of the AI and Games Symposium at AISB 2014, April 2014
6. Calanca, P., Busetta, P.: Cognitive navigation in presto. In: Proceedings of the AI and Games Symposium at AISB (2015)
7. Chu, Y.L., Li, T.Y.: Realizing semantic virtual environments with ontology and pluggable procedures. In: Applications of Virtual Reality. InTech (2012)
8. Dragoni, M., Ghidini, C., Busetta, P., Fruet, M., Pedrotti, M.: Using ontologies for modeling virtual reality scenarios. In: Proceedings of ESWC (2015)
9. 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)
10. Faulkner, S., Kolp, M.: Ontological basis for agent ADL. In: Eder, J., Welzer, T. (eds.) The 15th Conference on Advanced Information Systems Engineering (CAiSE 2003), Klagenfurt/Velden, Austria, 16–20 June, 2003, CAiSE Forum, Short Paper Proceedings, Information Systems for a Connected Society. CEUR Workshop Proceedings, vol. 74. CEUR-WS.org (2003). http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-74/files/FORUM_44.pdf
11. Fernández-López, M., Gómez-Pérez, A., Juristo, N.: Methontology: from ontological art towards ontological engineering. In: Proceedings of Symposium on Ontological Engineering of AAAI (1997)

12. Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., Schneider, L.: Sweetening ontologies with DOLCE. In: Gómez-Pérez, A., Benjamins, V.R. (eds.) EKAW 2002. LNCS (LNAI), vol. 2473, pp. 166–181. Springer, Heidelberg (2002). http://dl.acm.org/citation.cfm?id=645362.650863
13. Gruber, T.R.: Toward principles for the design of ontologies used for knowledge sharing? Int. J. Hum. Comput. Stud. **43**(5–6), 907–928 (1995). http://www.sciencedirect.com/science/article/pii/S1071581985710816
14. Guarino, N., Welty, C.: Identity and subsumption. In: Green, R., Bean, C.A., Myaeng, S.H. (eds.) The Semantics of Relationships: An Interdisciplinary Perspective. Information Science and Knowledge Management, vol. 3, pp. 111–126. Kluwer, The Netherlands (2001)
15. Rao, A.S., Georgeff, M.P.: Bdi agents: from theory to practice. In: Proceedings of the First International Conference On Multi-agent Systems, ICMAS 1995, pp. 312–319 (1995)
16. 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. Biol. Inspir. Cogn. Architect. **1**, 2–13 (2012)
17. Xuesong, W., Mingquan, Z., Yachun, F.: Building vr learning environment: an ontology based approach. In: First International Workshop on Education Technology and Computer Science, ETCS 2009, vol. 3, pp. 160–165, March 2009