Development of platform-independent multi-user choreographies for virtual worlds based on ontology combination and mapping

Emanuel Silva, Nuno Silva
School of Engineering
Plytechnic of Porto
Porto, Portugal
{ecs,nps}@isep.ipp.pt

Hugo Paredes, Paulo Martins, Benjamim Fonseca,
Leonel Morgado
INESC TEC/UTAD
University of Trás-os-Montes e Alto Douro
Vila Real, Portugal
{hparedes, pmartins, benjaf, leonelm}@utad.pt

Abstract*—This paper presents two contributions: (i) a system architecture capable of staging platform-independent choreographies within different virtual worlds, and (ii) an ontology-based solution for capturing and representing multi-user choreographies with reduced time/effort. We argue that choreographies for virtual worlds should be clearly separated from the technical characteristics of their execution in virtual world technological platforms. Due to the heterogeneity of the various virtual worlds and their domain requirements, we propose exploiting the modularity, generality, and granularity dimensions of ontologies to simplify and empower the choreography modeling capabilities. Instead of a unique ontology, several ontologies with different levels of generality and granularity can be progressively combined to support the modeling requirements of a given choreography. Because these ontologies are aligned with the ontology of each specific virtual world platform, the mapping and transformation between the core ontology is simplified and automated, thus reducing the development and time-to-market.

***Keywords*— virtual worlds; choreographies; multi-user; ontology; combination; mapping**

I. Introduction

Virtual worlds have achieved significant levels of interest for supporting teaching/learning activities [1], [2], particularly to enhance the exploitation of multi-agent situations [3-6], by providing immersive environments where concepts are developed in a simulated context [7], [8]. Virtual worlds such as Second Life, OpenSimulator, Activeworlds, and others have to some extent democratized this process, by allowing non-technical users to develop immersive 3D environments which can be readily made available to other users.

Choreographies of virtual actors are a specific type of educational content employed in simulation or role-playing environments, allowing a set of users to be involved in complex, multi-actor contexts for learning. This content requires planning, coordination of actors, sequencing actions and defining constraints, so that all individual components are consistently organized as a whole. Choreographies of virtual actors can be seen not just as automated behaviors, but as actions that can be performed entirely by human users, or in concert by a combination of human users and virtual computer-controlled characters able to respond to changes in their environment or context –enabling human trainees/students to play educational roles as part of teams. For instance, we are cooperating with the Portuguese Air Force on a mechanical maintenance simulator for F-16 engines where mechanics may watch a team of virtual mechanics interact with a virtual engine, or be part of the team, to practice established maintenance procedures [3].

Designing choreographies is a resource-intensive, time-consuming and costly effort. More so in mixed scenarios combining human and computer-controlled actors, the latter's behavior having to be programmed/developed. For this reason, it would be desirable for these choreographies involving computer-controlled actors not to be hostage to a specific virtual world platform, but rather deployable on potentially any virtual world platform. Currently, since virtual platforms are very heterogeneous, e.g. in terms of functionalities, data models, execution engine, programming interfaces for integration with external systems, and programing/scripting language, deploying a platform-based choreography into another platform is difficult and time-consuming.

We believe that a further step can be taken in the wider application of these platforms to educational scenarios, if developers could share across different virtual world platforms not just ordinary 3D resources such as objects and animations, but also complete choreographies including computer-controlled actors, as those found in advanced simulations.

In this paper we propose a design/model approach to represent mixed human/automation choreographies in a way that enables their deployment on distinct virtual world platforms with minimal effort and time. The rest of the paper comprehends four more sections. Section II describes background knowledge and related work. Section III presents the proposed system architecture, and Section IV details the data dimension of the system architecture. In particular section IV describes the efforts developing the choreography ontology and its application in the architecture. Finally, Section V summarizes the proposal and advances future directions.

978-1-4673-0853-3/12/$31.00 ©2012 IEEE

II. Related Work

Normally the creation of a choreography foresees its staging in a specific virtual world, and the application and planning of concepts is tailored to this virtual world. Sharing the content produced in order to be used in different virtual worlds has not been subject to substantial research.

The MIMESIS system [9] uses the Unreal Tournment game engine to integrate control of actions based on a plan through an interactive environment. The architecture is designed to bridge the gap between game engine design and development. The actions are represented in a dual form: a procedural representation for game engines and a declarative one for AI systems. To represent individual actions, MIMESIS follows the STRIPS language approach [10], which expresses planning representing actions, states, and goals. In STRIPS, the world state is represented by a set of predicates and every action that can be used is described as a set of preconditions for its execution and a set of post-conditions. These two separate representations are still relatively linked because they depict the same information, the existing mapping only converting strings between them.

Another example is the Zócalo system [11], [12] inspired by MIMESIS, with a tool to link the declarative representation of planning domain and the procedural representation of virtual worlds through code generation techniques. Data is represented in XML, with semantics based upon Lisp-like data formats. Code generation is based on a partial class generation approach, with code modules mapped with planning info, and an XML representation created in the declarative style of STRIPS. For each virtual world a XSLT template is created to transform the XML and generate virtual-world-specific source code. A software client replaces the default game engine to control the environment and action execution.

III. System Architecture

We argue that the representation and design of choreographies should be clearly separated from the technical characteristics of deployment in virtual world platforms. The core ideas behind our approach are illustrated in Figure 1.

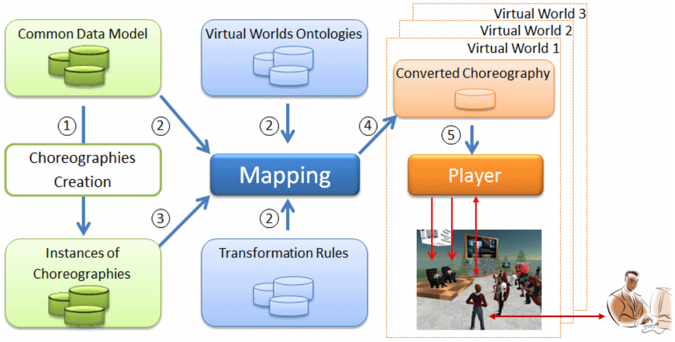

Figure 1. Process of creation, transformation and staging of choreographies for any virtual world.

The proposed architecture has three components: (1) data models (a generic high-level and independent data model, and specific data models each virtual world platform); (2) a mapping component that establishes links between the generic data model and the specific data model of each virtual world, and performs all the necessary data transformations; (3) a choreography player component that has the ability to schedule actions according to the choreography data and control its execution by virtual characters. Further, it must monitor the performance of avatars controlled by human users, by comparing the performed actions with those described in the choreography, and react accordingly. As a data model we decided to adopt ontologies as an adequate compromise between representation and expressiveness of information and operate close to the human conceptual level. Also, there is long-established ontology mapping and data transformation technology and literature [13].

Firstly, for the common data model an ontology will serve as a generic reference for any virtual world, capturing the common conceptualization of a particular domain of application: relevant (i) domain concepts, their (ii) properties, (iii) relationships, and (iv) the characters that carry out the (v) actions in the choreography.

Secondly, the content authoring process by choreographers consists of instantiating the action concepts defined in the common data model and their planning (1). This process of authorship results in the instantiation of choreographies. We envisage that this process can be achieved by specific-purpose applications, adequate for end-users. These can be varied, from simple schemes (e.g., [7]) to programming by example or demonstration of sample choreographies from human users.

Thirdly, in order to know exactly how a domain of application is represented within a particular virtual world, one must set up a specific ontology to describe that virtual world's concepts and properties. In our approach, it is a necessary condition that for each virtual world where an ontology will be staged, an associated ontology exists containing the information required for the domain of application. Their development may be time-consuming, but they need only be done once for each platform.

Fourthly, correspondences and transformation rules between ontologies should be created, containing matches and adjustments between entities of both ontologies: the generic one and the specific one. I.e., acting as a bridge between the ontology of the common model and the ontology specific to a virtual world platform.

Finally, the mapping process is a fundamental point, due to the separation between the generic information and the specific information of a staging environment. This process is based on all conceptual information defined by the ontology of the common model, the ontology of a virtual world platform for staging and the transformation rules between these ontologies (2). A choreography that is instantiated according to concepts of the ontology of common model is loaded (3) and converted using the transformation rules. The result will be a choreography whose instances correspond to the target virtual world concepts (4). Customizations according to the destination virtual world can be applied to the choreography during this conversion phase, and specify its storage format.

For staging a choreography within a virtual world, it is necessary that its contents are interpreted and executed through a choreography Player compatible with this virtual world (5). The Player is a computer program capable of interpreting the high-level language of the choreography and transforms it into low-level commands specific to each virtual world platform. It carries out the choreography staging, monitors its execution, and responds to context and environment changes. It must have the ability to schedule actions and distribute them by virtual characters. When a role is played by a human user, the Player must monitor his/her behavior, comparing performed actions with expected ones in the choreography, and react accordingly.

IV. The Data Dimension

Here we describe the choreography abstract data model, the domain choreography data model and the mapping process. The choreography ontology is a semantic description of the entities in the representation of choreographies. This ontology is then further applied (extended) semantically describing the choreography entities specific to an application domain, giving rise to the so called domain choreography ontology. Accordingly, while the choreography ontology is applied to a variety of domain applications, the domain choreography ontology tends to be specific to an application domain.

A. The Choreography Ontology

STRIPS has been used to describe character actions in virtual worlds. It has two fundamental limitations. First, it is not able to capture the conceptualizations of the domain (objects, states, characters, scenario). Serializing STRIPS choreographies in XML does not overcome this limitation as XML has insufficient expressivity to explicitly and formally specify a domain conceptualization [14]. Second, it does not support choreographies based multiple characters: the concept of "character" is not present in STRIPS. But it can represent goal-oriented plans, their actions and pre/post conditions in an isolated representation of entities, following a logical structure that allows their sequencing. We aim to join the goal-driven approach with restrictions, so actions can be assigned to avatars with specific roles. To use the same information in different environments, a common understanding about it is necessary. This happens when participants of the various involved fields of knowledge agree and share the meaning of concepts. In this sense, we propose using ontologies as explicit specifications of conceptualizations [15] to model domain knowledge, since "ontologies represent the best answer to the demand for intelligent systems that operate closer to the human conceptual level" [16]. In a simple and pragmatic way, an ontology is a data structure composed of a set of concepts and properties, with concepts organized according to a taxonomy, interrelated through properties and restrictions/constraints. Further, the semantics associated with the ontology's constraints ensure a common understanding about the data. Particularly relevant is their combination with inference engines, endowing ontologies with the capacity to obtain new implicit facts from those declared explicitly. Thus, we adopted an ontology-based representation of STRIPS and extended it with missing features. The resulting choreography ontology is in Figure 2.

In particular, notice the concepts of Choreography, Scenario, Object and Character, and several relations between these and the STRIPS-proposed concepts. The choreography ontology is sufficiently generic for avoiding particular specifications of any particular virtual world. This is an abstract ontology, in sense that it is not intended to be instantiated. Instead, it is intended to be extended with the specific domain concepts and constraints, in a way that the included entities inherit the semantics of the initial entities.

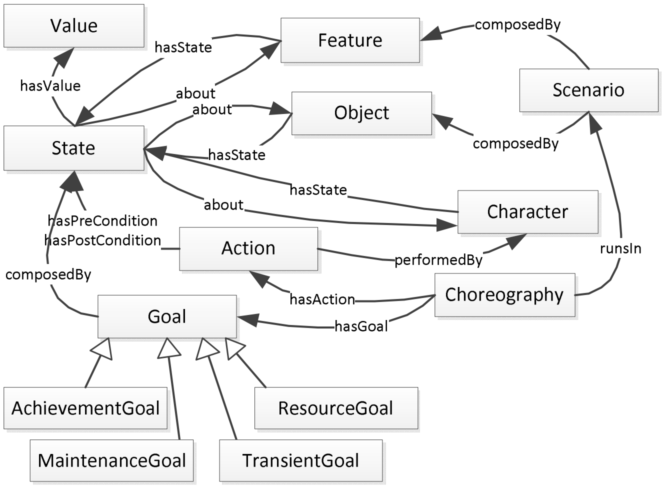

Figure 2. The Choreography Ontology

B. Domain Choreography Ontology

The domain choreography ontology is therefore the model that semantically describes the choreography in a specific domain. A high-level representation can have different interpretations by different stakeholders. Similarly, in different virtual worlds or even in the same virtual world, the implementation of a high-level action can be performed differently, with more or less detail depending on the desired accuracy. In light of this it is desirable to allow the representation of actions with different levels of granularity and detail to make possible its future execution with the degree of accuracy better suited to the requirements of the virtual world of implementation or staging requirements. The data model should remain equidistant to virtual worlds but, dependent on the scope of the application (e.g. children's story, training of assembly procedures).

For that we propose adopting a modularization approach and exploit the generality and granularity dimension of ontologies, simplifying and empowering the modeling capabilities. Instead of reinventing the wheel, the idea is to import already existent ontologies that represent relevant knowledge. Several ontologies with different generality and granularity are progressively combined (imported) to support the modeling requirements of the domain/choreography. Four types of ontologies are considered: domain specific ontologies (both domain and actions/tasks ontologies) and generic ontologies that are independent of domain (either about concepts or action/tasks). The center of Figure 3 depicts the taxonomy of an example of a domain choreography ontology. This is a concrete ontology, in the sense that will be used to create instances of the choreographies.

C. The Transformation of Choreographies

While the domain choreography ontology contains the concepts and properties of a specific domain, it can be substantially different from the ontology of any virtual world. The transformation rules specify which concepts of the domain choreography ontology match concepts of a particular virtual world ontology, according to the most appropriate level of detail.

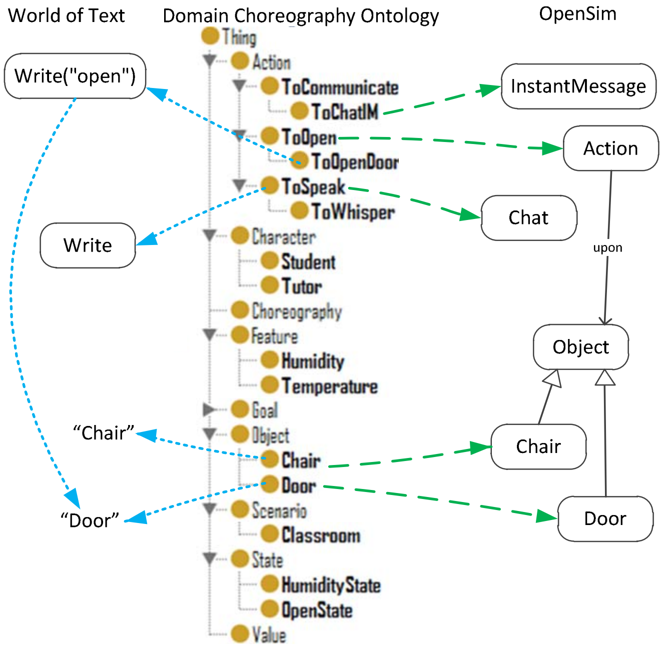

Figure 3. A graphical example of ontology mapping between ontologies.

This is done by establishing a correspondence and a relation between two concepts of the two ontologies. By connecting two concepts it becomes explicit as they are recognized in each of the ontologies thus eliminating any possible difference of terminology and semantics. Transformations are performed to convert the terminology, the representation format and the desired level of detail through an appropriate set of mapping rules to each of these virtual worlds.

V. Conclusions

The proposed system architecture allows the development of platform-independent choreographies and their staging in various virtual world platforms seamlessly. It's based on an ontology-based representation of choreographies, and its transformation to various platforms through ontology mapping. We support the idea that the use of ontologies is the best way to represent the conceptual information to approximate the intelligent systems to the human conceptual level and propose a definition of a generic high-level data model to represent concepts, properties, restrictions and actions, and the use of different levels of granularity and detail to enable the representation of the different features and functionalities of the various virtual worlds. Further, we advocate the exploitation of modularization, granularity and generality dimensions of ontologies in order to reuse knowledge about actions, domain, choreographies, etc. Ontologies aligned with specific virtual world platforms, simplify and automate the mapping and transformation from the core ontology, reducing development time and resources.

While the proposed ideas are conceptually validated through an in-house developed prototype, more suitable tools are necessary to aid creating and validating choreographies, and thorough experiments must be carried out to evaluate the pragmatic validity of the approach.

Acknowledgment

This work is partially supported by the Portuguese projects: World Search (QREN11495) and OOBIAN – Living Knowledge (QREN 12677), both funded by FCT/MCTES and FEDER through the COMPETE program for operational factors of competitivity.

References

[1] R. Macredie, S. J. . Taylor, X. Yu, and R. Keeble, 'Virtual reality and simulation: an overview', in Proceedings of the 28th conference on Winter simulation, 1996, pp. 669–674.

[2] L. Morgado, J. Varajão, D. Coelho, C. Rodrigues, C. Sancin, and V. Castello, 'The Attributes and Advantages of Virtual Worlds for Real World Training', The Journal of Virtual Worlds and Education, vol. 1, no. 1, pp. 15-35, 2010.

[3] B. Fonseca, H. Paredes, J. Rafael, L. Morgado, and P. Martins, 'A Software Architecture for Collaborative Training in Virtual Worlds: F-16 airplane engine maintenance', in Collaboration and Technology: 17th International Conference, CRIWG 2011, Proceedings, pp. 102-109, Berlin: Springer, 2011..

[4] T. E. Wright and G. Madey, 'A survey of technologies for building collaborative virtual environments', The International Journal of Virtual Reality, vol. 8, no. 1, pp. 53–66, 2009.

[5] K. Getchell, I. Oliver, A. Miller, and C. Allison, 'Metaverses as a Platform for Game Based Learning', in Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on, 2010, pp. 1195–1202.

[6] A. Schmeil, M. Steinbusch, A. Jost, M. Henn, M. Jacobi, M. Schwitalla, and B. Hasler, 'A Workflow for Designing Virtual Worlds for Collaborative Learning', in Games and Virtual Worlds for Serious Applications (VS-GAMES), 2010 Second International Conference on, 2010, pp. 151–158.

[7] A. Lopes, B. Pires, M. Cardoso, A. Santos, F. Peixinho, P. Sequeira, and L. Morgado, 'System for Defining and Reproducing Handball Strategies in Second Life On-Demand for Handball Coaches' Education', in Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications - ED-MEDIA 2009, Honolulu, Hawaii, USA, pp. 3954-3961. Chesapeake, VA: AACE, 2009.

[8] P. Kapahnke, P. Liedtke, S. Nesbigall, S. Warwas, and M. Klusch, 'ISReal: An Open Platform for Semantic-Based 3D Simulations in the 3D Internet', in The Semantic Web – ISWC 2010, vol. 6497, P. Patel-Schneider, Y. Pan, P. Hitzler, P. Mika, L. Zhang, J. Pan, I. Horrocks, and B. Glimm, Eds. Springer Berlin / Heidelberg, 2010, pp. 161–176.

[9] R. M. Young, M. O. Riedl, M. Branly, A. Jhala, R. J. Martin, and C. J. Saretto, 'An architecture for integrating plan-based behavior generation with interactive game environments', Journal of Game Development, vol. 1, no. 1, pp. 1–29, 2004.

[10] R. E. Fikes and N. J. Nilsson, 'STRIPS: a new approach to the application of theorem proving to problem solving', in Proceedings of the 2nd international joint conference on Artificial intelligence, San Francisco, CA, USA, 1971, pp. 608–620.

[11] T. M. Vernieri, 'A web services approach to generating and using plans in configurable execution environments', 2006.

[12] R. M. Young, J. Thomas, C. Bevan, and B. A. Cassell, 'Zócalo: A Service-Oriented Architecture Facilitating Sharing of Computational Resources in Interactive Narrative Research', 2011.

[13] J. Euzenat and P. Shvaiko, Ontology Matching, Softcover reprint of hardcover 1st ed. 2007. Springer, 2010.

[14] M. Klein, D. Fensel, F. Van Harmelen, and I. Horrocks, 'The relation between ontologies and XML schemas', Electronic Trans. on Artificial Intelligence, 2001.

[15] T. R. Gruber and others, 'A translation approach to portable ontology specifications', Knowledge acquisition, vol. 5, no. 2, pp. 199–220, 1993.

[16] L. Obrst, H. Liu, and R. Wray, 'Ontologies for Corporate Web Applications', AI Magazine, vol. 24, no. 3, p. 49, Sep. 2003.