MMOG based on MAS: The MMOG Layer

(Extended Abstract)

G. Aranda
Universidad Politécnica de Valencia
Departamento de Sistemas Informáticos y Computación (DSIC)
Camino de Vera s/n – 46022
Valencia – Spain
garanda@dsic.upv.es

V. Botti
Universidad Politécnica de Valencia
Departamento de Sistemas Informáticos y Computación (DSIC)
Camino de Vera s/n – 46022
Valencia – Spain
vbotti@dsic.upv.es

C. Carrascosa
Universidad Politécnica de Valencia
Departamento de Sistemas Informáticos y Computación (DSIC)
Camino de Vera s/n – 46022
Valencia – Spain
carrasco@dsic.upv.es

ABSTRACT
Massively Multiplayer Online Games present a new and exciting domain for service-oriented agent computing as the mechanics of these virtual worlds get more and more complex. Due to the eminently distributed nature of these game systems and their growing necessity of modern AI techniques, it is time to introduce design methods that take advantage of the power of Multi-Agent Systems, Agent Organizations and Electronic Institutions in order to face the challenges of designing a modern Massively Multiplayer Online Game. This article follows previous pieces of work in the line of Multi-Agent Systems and Massively Multiplayer Online Games research into a common ontology to represent the information that agents store and share and towards the use of the THOMAS service architecture and the SPADE agent platform to address these challenges. The main focus of the article is the so called MMOG Layer: a software layer which is independent of the environment simulation and the human-interface devices. An example implementation of such layer is also set out.

Categories and Subject Descriptors
I.2.11 [Distributed Artificial Intelligence]: Multiagent Systems; K.8 [Personal Computing]: General—Games

Keywords
Multi-Agent Systems, MAS, Massively Multiplayer Online Games, MMOG

1. MMOG BASED ON MAS ARCHITECTURE

This piece of work follows in the footsteps of previous research efforts like [1] and [2] in which games in general and MMOG in particular are researched as natural scenarios for agents and MAS. A MMOG (like most complex systems) can be seen as a system split into several layered subsystems, with each layer being relatively independent and taking care of one aspect of the whole MMOG experience. From the perspective of this work, a MMOG is split into three layers: The Human-Computer-Interface (HCI), which is the client-side of the system; the Intelligent Virtual Environment, which is the virtual representation and simulation of the game environment; and the MMOG layer.

The MMOG Layer is a complex subsystem where all the game logics and mechanics are implemented and must be solved at run-time. It is independent of the IVE layer. It implements the game rules / norms controlling the game development. It is the place where all the game clients connect to play, and along with the IVE layer must facilitate game server scalability. In this line of research, this subsystem is seen as the core of a MAS and requires, at least, one agent platform as its foundation.

2. THE MMOG LAYER

Like any other agentification process, one of the key ideas is to identify the agents and types of agents that will form the system. In this case, the agents are based on the concepts and entities that form the whole game experience of a MMOG, and are explained in more detail in [2].

One issue left partially unsolved in previous works, and that this paper addresses, is the need for a common ontology for all these agents to be able to share information in a consistent way (i.e. they need to speak the same content language). It is clear that every game or game genre has its own mechanics, rules and players, but, in essence, all MMOGs share some common concepts, like player avatars. These common concepts can be expressed using a common set of conventions, a common knowledge base. Then, in order to express specific concepts for a given game or genre, custom ontologies that derive from the original one can be described. This common ontology is called MMOG Ontology.


Copyright © 2009, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.
The current project and helps developer to establish work methodologies; and fourth, it allows for rapid game prototype creation using already available tools for developing ontologies.

The MMOG Ontology has been defined using the Web Ontology Language (OWL). More precisely, it is expressed using the OWL-DL (Description Logics) sublanguage for two reasons: one, some of the features of this sublanguage, like defining one-to-many-like relational properties, where needed; and two, the full set of the OWL language (called OWL-Full) is not deemed as being computable [3]. Even so, it is a quite light ontology, as it is composed of some base classes that have datatype and object properties. The ontology uses the ‘mmog:’ prefix and it presents the following classes: mmog: Avatar, mmog: Clan, mmog: GameBeacon, mmog: GameItem, mmog: GameZone, mmog: Goal, mmog: Profile, mmog: Requirement and mmog: RewardPolicy.

mmog:Avatar: This is the generic class to represent an avatar within the game. It does not matter if its a player-controlled avatar (i.e. played by an AvatarAgent) or an AI-controlled avatar (played by a NPCAvatarAgent). Avatars are uniquely identified by their hasName property which ties each of them to a unique name string identifier within the game system.

mmog:Clan: This class represents a Player Clan that the system implements by means of an Agent Organization. It has properties referring its members, lords, goals, reward policies and constraints.

mmog:GameBeacon: A beacon is a sort of ‘named point’ of the virtual world. A marked position inside a mmog: GameZone which is referred by name. For example, the starting point or one of the exits.

mmog:GameItem: An inanimate object within the game world uniquely identified by a code string. For example, in an adventure game, the items that the players virtually carry in their inventories are instances of a subclass of mmog:GameItem.

mmog:GameZone: A class to represent a game zone. As stated earlier, a game zone is a segment of the virtual game world with its own name, theme and uniqueness. It holds one or more requirements to be entered.

mmog:Goal: A collection of conditions expressed within valid semantic terms of the game. When all the conditions are met, then the goal is reached. For example, a condition could be the equivalent expression of ‘Place object A in beacon XYZ’ or ‘Avatar B is a member of Clan C’.

mmog:Profile: The profile of a player within the game community: The player’s screen name along with all the information the system wants to keep from the player, the avatars the player controls, which role the player plays within the game system and the contact list for this player (i.e. the ‘friends list’).

mmog:Requirement: A requirement to enter a given GameZone. It can specify terms regarding player clans attributes, goals or player role.

mmog:RewardPolicy: A reward policy to apply once a goal has been achieved. It holds a collection of actions defined within game terms. Actions will normally be specified using terms from derived ontologies of specific games.

And so, this ontology serves as a foundation to build other, more specific ontologies to express the knowledge of a particular game or type of game. In the next section, an example of these kind of ontologies can be found.

2.1 Prototyping the MMOG layer

Analysing the feasibility of the proposal by means of a prototype based on a concrete game, and making use of different agent platforms to allow to manage the different services, virtual organizations and agents involved in this kind of layer. Right now, a prototype is being prepared for an online racing game where players can assume the roles of drivers of different types of vehicles (Cars, Motorbikes and Trucks) that compete in online races.

3. CONCLUSIONS AND FUTURE WORK

This work has continued the research line of MMOG Multi-Agent Systems by providing a starting point for building and exchanging semantic content between agents of these kind of systems. The existence of a basic ontology (a common starting point) to represent game-related knowledge is needed in order to achieve greater long-term goals like standardization, inter-systems communication or define a methodology to build these kind of system from the ground up. In this work, such common ontology has been defined using the standard OWL-DL semantic language, as well as a small example of a more refined ontology for a specific game type. Both of them can be downloaded from the web: http://gti-ia.dsic.upv.es/ontologias/ and are coming together for deployment and use.

An interoperability between two previously unrelated agents and services platforms, THOMAS and SPADE, has been created in order to allow agents of the MMOG Layer to provide and consume services specified in the OWL-S semantic language, which is derived from the OWL language used in the common ontology.

Moreover, the path has been open to properly define and specify the agent interactions and protocols that conform the whole communication scheme of the system, as now there is an ontology for these agents to use and to express their semantic knowledge and an organizational platform to support and enforce services, norms and agent organizations.

4. ACKNOWLEDGEMENTS

This work has been partially funded by TIN2006-14630-C03-01, PROMETEO /2008/051 and GVPRE /2008/070 projects and CONSOLIDER-INGENIO 2010 under grant CSD2007-00022.

5. REFERENCES

