Model Driven Game Quest Scenario Development for Massively Multi-Player Role-Playing Games: A Case Study*

Bogdan Okreša Đurić, Igor Tomičić, Sven Vukelić
University of Zagreb
Faculty of Organization and Informatics
Pavlinska 2, Varaždin, Croatia
{dokresa, itomicic, svukelic}@foi.hr

Abstract. *In this paper, a new application case of the novel organizational metamodel for Large-Scale Multiagent Systems (LSMASs) being developed is going to be presented. The mentioned metamodel is being tested on a new computer game of Role-Playing Game (RPG) genre by developing a game quest scenario through the model-driven approach. This new application of the said metamodel will further enhance the feature of general applicability of the metamodel to various LSMAS domain examples.*

Keywords. mmorpg, agent based modeling, agent organizations

1 Introduction

The graphical modeling tool for modeling and building a system comprising agents, groups of agents, roles (special types of grouped constraints), actions (inherently bound to specific roles), sets of actions designated as processes, and quests (composed of a series of sub-goals or tasks), used in this paper, was presented in (Okreša Đurić, 2017; Okreša Đurić, 2016), and the initial application case on The Mana World (TMW) Massively Multi-player Online Role-Playing Game (MMORPG) was introduced in (Schatten, Tomičić, et al., 2017; Schatten, Okreša Đurić, et al., 2017b; Schatten, Okreša Đurić, et al., 2017a). Although the existing body of research has considerably contributed in bringing the model driven approaches into the game development domain (e.g. in modeling narratives, spatial and temporal relationships, gameflow, game modes and similar), the novelty of our approach is given via the organizational aspects of in-game quests that are inherently bound to organizational behaviours of players and Non-Player Characters (NPCs). By applying the organizational metamodel for LSMASs on the second test-bed and developing the game quest scenario with organizational aspects in mind, the general applicability of the metamodel would be further argued.

*This work has been fully supported by the Croatian Science Foundation under the project number 8537.

2 Related Work

According to (Dormans, 2012), model-driven engineering "is a suitable approach to deal with the dynamic process of designing games." A state-of-the-art in model-driven game development, summarizing the developments in "game design languages, game software modelling languages, game models, game software models, model-driven game frameworks, game software frameworks, model-driven engineering tools and assistive user interfaces", is presented by (Tang and Hanneghan, 2011). In their work, authors provide an extensive overview of given subjects, supported by illustrative examples, advantages and shortcomings of reviewed languages, tools and frameworks.

Adams proposed the so called "flowboard" for the purposes of documenting game structure (Adams, 2004). The word "flowboard" is derived from a combination of the words flowchart and storyboard, using both approaches in modelling by using a number of pictures (rectangles) in a non-linear series, presenting a mockup of the screen in one of the gameplay modes (for example, menu screen, high score table, splash screen, etc.). The arrows in the flowboard denote constraints of progressing from one gameplay mode to another, with text labels describing events' triggers that are required for the transition to take place. The flowboard is a simple tool for modelling various gameplay modes, that might be adapted for a certain level game scenario dynamics, but lacks the expressiveness for describing more details about game quests, organizational structure and game role constraints.

Onder proposed the story beat diagram for modelling the progression of a story (Onder, 2002). The elements of the diagram include ovals (representing activities/tasks) and arrows (representing progressions), and are able able to describe available routes for the game player, which are possible to explore within the game story. While providing an ordered overview of the game-play scenario flow, the story beat diagram does not provide additional details on players, organizational context, spatial data, interactions of players/NPCs and similar.

Unified Modelling Language (UML) has also been

used in the context of game design. In their work (Taylor et al., 2006), authors have adapted the use-case diagram language defined in UML in order to create a custom modelling language for the description of the game flow, intended for the design of individual game levels, and have demonstrated its application in practice. Their "Computer Game Flow Design Diagram" uses arrows do denote directions of a game-flow, rectangles for objects that are fixed within the game spatial context, and ovals for objects that can move throughout the game space. Although their model additionally uses pseudo-code for the description of various details of the game (responses, interactions, etc.), the graphical modelling language itself lacks the expressiveness for describing details of characters, roles, quests, organizational aspects, etc.

Finite State Machines (FSMs) have also been utilized in modelling games, but mostly in modelling game AI (Kienzle et al., 2007). Although FSM diagrams have considerable potential in modelling games, considering that games often include various states that can change by certain triggers (rules, interactions, events), they lack the ability to describe the quests in an organizational context (hierarchy, roles, etc.).

Natkin et. al (Natkin et al., 2004) proposed a "new formal approach for the design process of computer games", involving the modelling of combined spatial and temporal relationships in games. Authors used hypergraphs for spatial (topological) relationships' modelling, and logical and temporal relationships are modeled by using Petri nets. By using this approach, game developers could model game flow and game events, but couldn't tackle characters, their roles/behaviours, quest details or organizational aspects of the game quests.

Although the existing body of research within the model-driven game development field showed promising results on various aspects of game development (narratives, spatial and temporal relationships, game flow, game modes), there is a lack of research involving organizational aspects of more complex games that include quests and foster organizational behaviour, such as MMORPGs.

3 Organizational Metamodel for Large-Scale Multiagent Systems

The novel organizational metamodel for LSMASs that will, in the end, include working with organizational dynamics as a feature of human organizations translated to LSMAS domain, was initially presented in (Okreša Đurić, 2017) and is being developed as an integral part of the ModelMMORPG project[1]. The main idea of the mentioned metamodel is organizational modeling of complex multiagent systems comprising large numbers of agents, thus referring to the idea of LSMASs. The *meta* prefix is based on the fact that the metamodel can be used for various purposes, i.e. for organizational modeling of models pertaining to various application domains, all of which are based in the general concept of LSMAS. This multi-application statement is an intention to show various application domains that can be modeled using the said metamodel through a number of case-studies. An example of a model is shown in Sec. 4.1 of this paper.

The general idea of the metamodel is to provide a tool for organizational modeling of LSMAS concepts in a manner similar to that of organizing human organizations, but with a special emphasis on seven perspectives which are argued to be invaluable for the ongoing research of LSMASs (Schatten, 2014):

- organizational structure,
- organizational culture,
- strategy,
- processes,
- individual agents,
- organizational dynamics,
- context and inter-organizational aspects.

The stated is set to be achieved through a basically two-step process initially presented in (Okreša Đurić, 2016), and further discussed in (Okreša Đurić, 2018): an established ontology comprising concepts of organizational modeling, and based upon it a metamodel containing all the necessary concepts of the mentioned ontology that are deemed needed and necessary for the main goal of the metamodel - organizational modeling of LSMAS.

As stated before, the metamodel is being developed as a part of the ModelMMORPG project, and is therefore mostly related to its main purpose - modeling of MMORPGs. The metamodel was envisioned with generality and wide spectre of possible use though, and is therefore created to be applicable to a wider array of LSMAS application domains. Being originally built for the application in MMORPG domain, but with general use in mind, the metamodel prominently features several key concepts:

- organizational unit - this concept that represents a class of agent that can play various roles contains some key actions (such as ChangeRole action used for changing roles the agent plays at the moment), and is knowledgeable of some norms of the system;
- role - a normative concept of the system, every norm is a set of constraints that can be posed on

[1]Large-Scale Multi-Agent Modelling of Massively On-Line Role-Playing Games; Artificial Intelligence Laboratory at the Faculty of Organization and Informatics, University of Zagreb; for further information visit http://ai.foi.hr/modelmmorpg.php

an agent, i.e. an agent can play a norm and thus apply a set of conditions of the system upon itself, such as a set of available actions;

- process - actions that can be performed by a role are grouped into sets called processes - the set of actions is not constrained to a single role though, and can consist of actions from various roles, all of which can be coordinated to achieving a common goal;

- objective - the concept of objective represents a goal that is to be reached by the agents of the system using available actions of the roles individual agents can play.

The mentioned core elements of the metamodel are applicable to the MMORPG domain, although for a more detailed description of a specific application example further concepts would be welcome (e.g. various types of items). This omitting of concepts that would foster detailed modeling is motivated by generalization of the use of the metamodel. Relation of MMORPG concepts and generalized LSMAS concepts can be noted in various examples not related to the domain of MMORPG, since the metamodel elements are general enough to be used in application domains other than MMORPG, e.g. smart cities, the Internet of Everything (IoE), etc.

Another interesting feature of the metamodel is its application template generator that is described in (Schatten, Okreša Đurić, et al., 2017a). This feature of the metamodel generates application template based on Smart Python Agent Development Environment (SPADE), thus providing the users with agent templates and their possible actions (or behaviours in SPADE language) as defined using the modeling tool. A sample listing of the generator is shown in Sec. 4.2, based on the example shown in this paper. SPADE was chosen as a tool that is based on Python programming language, and is rather simple to use, along with the fact that it is the basic Multiagent Systems (MASs) development environment used in the ModelMMORPG project.

4 Game-based Example

Minion Wars is an open source, browser-based alternate reality game based on usage of Google Maps and Google Places Application Programming Interface (API). User interacts with virtual world through map interface and is given a wide variety of actions that it can perform in immediate vicinity of its location which is tracked and updated in real time. Goal of the user is to gather experience points and resources by visiting specific locations called nodes. Nodes are either randomly generated or retrieved from Google Places API and are defended by NPC agents called minions. In order to access nodes, user is supposed to defeat enemy minions through use of its own minions. Some of the actions available to the user are:

- spend experience points to upgrade abilities - abilities are personal characteristics that define aspects such as amount and strength of users minions, number and range of minions it can send away on missions to remotely access nodes and speed of camp upgrades.

- build and upgrade camps - camps can host minions that then automatically gather resources from nodes in certain radius. Upgrading camps allows it to host more minions and access nodes in larger area.

- manage minions - organize minions in groups, decide which minions remain with user, which are stationed in camps and which are sent away on missions.

4.1 Example Quest Description

Landmark Quest is a special type of quest available to a user. User can earn rewards by completing sets of specific tasks that are geofenced to areas in certain radius of locations assigned to Landmark Quests. These quests can be started by simply entering the designated quest area. Some of the tasks that user can complete are as follows:

- defeat certain amount of enemies,

- defeat specific enemy with help of other users (e.g. a “boss” enemy),

- solve puzzles,

- visit specific locations within the area.

Users can perform any of the tasks available at the moment in any order. For each task completed, the user is awarded participation points. Upon getting certain predetermined amount of participation points, the quest is considered completed.

As the user progresses with tasks, smaller rewards are continuously awarded. The rewards can come in form of experience points, in-game currency, or resources and information. Upon completion of designated set of tasks, the user receives a much more substantial reward.

A particular example of such a quest within the context of Minion Wars would require for user to visit the Tower of London.

To start the quest, the user needs to enter the area in direct vicinity of the Tower of London. They are immediately presented with initial backstory and how the area is filled with hostile minions led by a minion chief. The user is tasked with visiting specific areas within the Tower (Inner Ward, Outer Ward, Innermost Ward). Between each visit, the user is also required to defeat several hostile minions that roam the area.

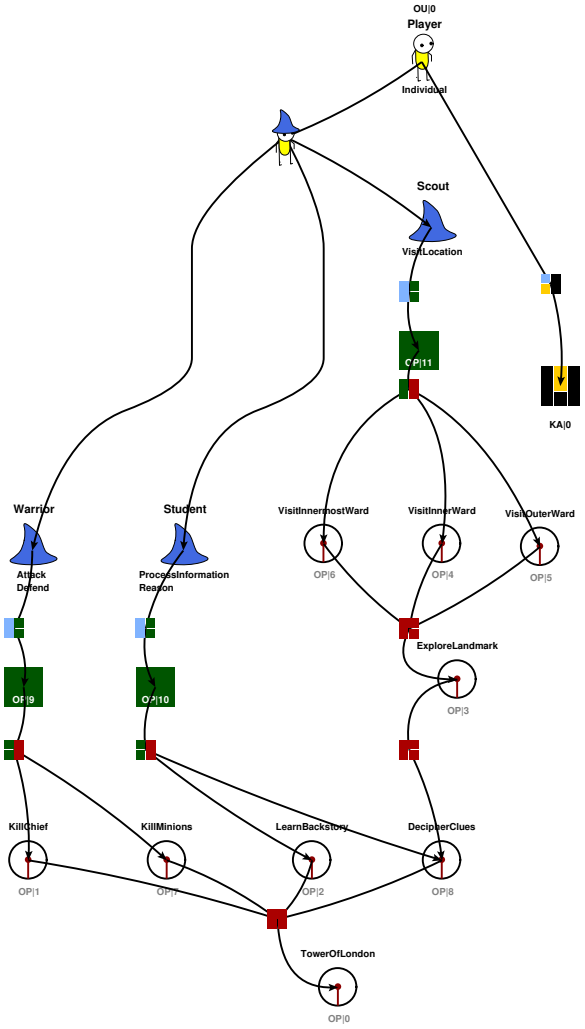

Figure 1. Model of the example quest using the described metamodel

Upon visiting each of the areas, the user is presented with information about the visited location and bits of information regarding the local minion chief. Upon visiting all the key locations, the user is tasked with finding the definite final location (the White Tower) by deciphering the clues from previous tasks. Upon entering the final location, the user is able to confront the minion chief in combat. After defeating the minion chief, the user is awarded further pieces of information about the Tower of London and significant in-game monetary reward.

4.2 Example Quest Modeling

The described example quest derived from the Minion Wars game can be modeled using the described metamodel as is shown in Fig. 1. It stems from the quest description, given in Sec. 4.1, that the player (yellow stickman in Fig. 1) has to play several different roles in order to be able to fulfill the given quest, and complete it successfully. These roles (blue hats in Fig. 1) can be, in the manner of MMORPGs, named as follows: Scout, Warrior, and the unusual one - Student. Furthermore, each agent has some individual knowledge modeled as an individual knowledge artifact seen as a yellow-black square in Fig. 1.

Since the game is envisioned as not only a fun-related experience, but an educational one as well, the Student role contains actions necessary for the agent (the human player in this case) to learn the knowledge the game provides them with, and use it when the game demands it. Therefore, the Student role is useful when completing knowledge-related tasks of the given quest example - The Tower of London (quests and tasks are shown circular in Fig. 1). The second important role emerging from the given quest is the Scout role, which enables the player agent (either artificial or human) to conduct actions (actions are grouped in processes shown as green squares in Fig. 1) pertaining to scouting the given area and moving through space. Such actions are necessary for the exploring tasks of the quest. Finally, the Warrior role is self-explainable, since the quest demands the player to best a number of minions, and their chief.

Only through the combination of the mentioned three roles, and their actions, the player can successfully complete the given quest. Upon finishing the quest, the player is presented with an award of some kind, depending on the quest, and their performance while solving the given quest.

Using the provided model, and the modeling tool being developed, which is described earlier in this paper, a developer can generate application template of the modeled system, along with the logical dependencies built based on the set relations of the defined roles, their actions, processes, and the tasks of the given quest. Such an initial programming code can boost the rapidity of modeled system development. An excerpt of such a template is shown in Lst. 1, which shows the SPADE agent template along with its single action for changing roles, and an excerpt from its knowledge base regarding the Scout role, its actions and tasks that can be achieved by playing it.

```
import spade
from RoleBehaviours import *

class Player(spade.Agent.Agent):

  class ChangeRole(spade.Behaviour.
      OneShotBehaviour):
    def _process(self):
      [...]

  def _setup(self):
    self.addBehaviour(self.ChangeRole(), None)

    self.configureKB('SWI', None, 'swipl')
    self.addBelieve('canReachGoal(Scout,
        VisitInnerWard)')
    self.addBelieve('canReachGoal(Scout,
        VisitOuterWad)')
    self.addBelieve('canReachGoal(Scout,
        VisitInnermostWard)')
    [...]
    self.addBelieve('hasAction(Scout,
        VisitLocation)')
```

Listing 1. Part of the generated code template for SPADE Player agent

It should be noted here that the application template generator part of the metamodeling tool provides the user (the developer of the modeled system) with basic programming code pertaining to the idea of implementing agents that follow the modeled organizational features (including roles and coalitions), and the basic knowledge base necessary for utilizing the said organizational features. Therefore, the extent to which the code is generated is limited to core elements needed for SPADE agent definition, along with their behaviours (actions), role playing features, and individual agent knowledge bases that allow the agents to enact various roles based on reasoning about quests, tasks, and behaviors. Reasonably, the amount of generated code depends on the complexity of the modeled system.

5 Discussion and Conclusions

Using the quest decomposition on several lower-level tasks that have to be fulfilled in order for the main quest to be achieved, the shown model gives a clear picture of the actions and roles, i.e. constraints of the system and norms, that are needed for a player to successfully finish the given quest. The addition of the application template generator is useful inasmuch as the defined model can be easily transferred to an implementation-ready code that can be used for testing the defined quest.

In this paper we have presented a new application case of the novel organizational metamodel for LSMASs that is being developed. The metamodel was tested on a new RPG computer game that was chosen as a test-bed primarily because of its quest-oriented design, by developing a game quest scenario through the model driven approach. The new application of the metamodel is an argument in favor of its feature of general applicability to various LSMAS domain examples.

As argued in Section 2, the existing body of research, dealing with game quest development questions, lacks organizational aspects of more complex games that deal with quests and utilize organizational behaviours. The research approach proposed within this paper encompasses expressiveness used to describe more details about game quests, quest hierarchies, organizational structure, game role constraints and interactions of players and NPCs.

The future work will orient towards improving the general readability of the metamodel in more complex game/organizational designs. Indeed, the metamodel has to be made more legible and easier to be read, which is an idea planned to be achieved through the use of multi-model modelling method.

References

Adams, E. (2004). Designing with gameplay modes and flowboards. *San Francisco: Gamasutra*.

Dormans, J. (2012). The effectiveness and efficiency of model driven game design. *Entertainment Computing-ICEC 2012*, 542–548.

Kienzle, J., Denault, A., & Vangheluwe, H. (2007). Model-based design of computer-controlled game character behavior. In *International conference on model driven engineering languages and systems* (pp. 650–665). Springer.

Natkin, S., Vega, L., & Grünvogel, S. (2004). A new methodology for spatiotemporal game design. In *Mehdi, q. and gough, n.,(eds.). proceedings of cgaide 2004, 5th game-on international conference on computer games: Artificial intelligence, design and education* (pp. 109–113).

Okreša Đurić, B. (2016). Organizational Metamodel for Large-Scale Multi-Agent Systems. In F. de la Prieta, M. J. Escalona, R. Corchuelo, P. Mathieu, Z. Vale, A. T. Campbell, ... M. N. Moreno (Eds.), *Trends in practical applications of scalable multi- agent systems, the paams collection* (Chap. 8, pp. 387–390). Advances in Intelligent Systems and Computing 473. Seville, ES: Springer International Publishing. doi:10.1007/978-3-319-40159-1_36

Okreša Đurić, B. (2017). A Novel Approach to Modelling Distributed Systems: Using Large-Scale Multi-agent Systems. In Z. Mahmood (Ed.), *Software project management for distributed computing* (1st ed., Chap. 10, pp. 229–254). Springer International Publishing AG. doi:10.1007/978-3-319-54325-3_10

Okreša Đurić, B. (2018). Towards Modelling Organisational Dynamics for Large-Scale Multiagent Systems. In F. De la Prieta, Z. Vale, L. Antunes, T. Pinto, A. T. Campbell, V. Julian, ... M. N. Moreno (Eds.), *Trends in cyber-physical multi-agent systems. the paams collection - 15th international conference, paams 2017* (1st ed., pp. 245–248). Advances in Intelligent Systems and Computing 619. Cham: Springer International Publishing. doi:10.1007/978-3-319-61578-3_28

Onder, B. (2002). Writing the adventure game. *Game design perspectives*, 28–43.

Schatten, M. (2014). Organizational Architectures for Large-Scale Multi-Agent Systems' Development: An Initial Ontology. *Advances in Intelligent Systems and Computing*, *290*, 261–268. doi:10.1007/978-3-319-07593-8_31

Schatten, M., Okreša Đurić, B., Tomičić, I., & Ivković, N. (2017a). Agents as Bots – An Initial Attempt Towards Model-Driven MMORPG Gameplay. In Y. Demazeau, P. Davidsson, J.

Bajo, & Z. Vale (Eds.), *Advances in practical applications of cyber-physical multi-agent systems: The paams collection* (pp. 246–258). Lecture Notes in Artificial Intelligence 10349. Cham, Switzerland: Springer International Publishing. doi:10.1007/978-3-319-59930-4_20

Schatten, M., Okreša Đurić, B., Tomičić, I., & Ivković, N. (2017b). Automated MMORPG Testing – An Agent-Based Approach. In Y. Demazeau, P. Davidsson, J. Bajo, & Z. Vale (Eds.), *Advances in practical applications of cyber-physical multi-agent systems: The paams collection* (pp. 359–363). Lecture Notes in Artificial Intelligence 10349. Cham, Switzerland: Springer International Publishing. doi:10.1007/978-3-319-59930-4_38

Schatten, M., Tomičić, I., Okreša Đurić, B., & Ivković, N. (2017). Towards an Agent-Based Automated Testing Environment for Massively Multi-Player Role Playing Games. *MIPRO 2017 40th Jubilee International Convention Proceedings*, 1361–1366. Retrieved from http://bib.irb.hr/prikazi-rad?rad=881115

Tang, S. & Hanneghan, M. (2011). State-of-the-art model driven game development: A survey of technological solutions for game-based learning. *Journal of Interactive Learning Research*, *22*(4), 551.

Taylor, M. J., Gresty, D., & Baskett, M. (2006). Computer game-flow design. *Computers in Entertainment (CIE)*, *4*(1), 5.