Validating Business Requirements Using MAS Analysis Models

[Extended Abstract]

Nektarios Mitakides
Technical University of Crete
University Campus
Chania, Greece
+30 69 45 12 72 37
nmitakidis@isc.tuc.gr

Nikolaos Spanoudakis
Technical University of Crete
University Campus
Chania, Greece
+30 28 21 03 77 44
nikos@amcl.tuc.gr

Pavlos Delias
Eastern Macedonia and Thrace
Institute of Technology
Agios Loukas, Kavala, Greece
+30 25 10 46 23 68
pdelias@teikav.edu.gr

ABSTRACT

This paper presents a method aimed to assist an engineer in transforming agent roles models to a process model, compliant with the XML Process Definition Language (XPDL) portable standard. This method bridges the gap between software engineers and the business world by allowing a Multi-Agent System (MAS) analysis model, such as the Gaia or ASEME Methodologies role models, to be represented as a business process model. Thus, on one hand, the software engineer can employ available tools to validate specific properties of the modeled system even before its final implementation, and, on the other hand, a business partner has greater potential to comprehend the system being modeled. The method includes a tool for aiding the engineer in the transformation process. This tool uses a recursive algorithm for automating the transformation process and guides the user to dynamically integrate two or more agent roles in a process model with multiple pools. The tool usage is demonstrated through a running example, based on a real world project. Simulations of the defined agent roles can be used to a) validate the system requirements and b) determine how it could scale. This way, developers and managers can configure processes' parameters and identify and resolve risks early in their project.

Categories and Subject Descriptors

D.2.1 [**Software Engineering**]: Requirements/Specifications – *Methodologies, Tools.*
D.2.4 [**Software Engineering**]: Software/Program Verification – *Validation.*
I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence – *Multiagent systems.*

Keywords

Verification of agent-based systems; Methodologies for agent-based systems

1. INTRODUCTION

This paper aims to bring agent technology close to the world of business modeling. It bridges the gap between software engineers and the business world by allowing a Multi-Agent System (MAS) analysis model to be represented as a business process model. Thus, on one hand, the software engineer can employ available tools to validate specific properties of the modeled system even before its final implementation, and, on the other hand, a business partner has greater potential to comprehend the system being modeled through intuitive process visualization.

In this paper we show how liveness formulas, an important property of agent role models, introduced by the Gaia methodology [8] can be transformed to process models. Moreover, we show how these models can be integrated to produce a process model of a multi-agent system using the XML Process Definition Language (XPDL) portable standard. Having transformed the MAS analysis model to a process model we can use simulation to verify several properties of the modeled system, and also determine its ability to scale. Therefore, this work is expected to have a high impact on a) Agent Oriented Software Engineering (AOSE) researchers, who will be informed about the properties of process models and the possibilities that they offer to software engineering, b) AOSE practitioners using the Gaia methodology and its successors, who can immediately take advantage of this work to evaluate their models, and, c) those who use business process models for agent-based simulations [17] or for communicating them to business people [3], who can now use an AOSE methodology to aid them in their modeling tasks.

2. THE Liveness2XPDL ALGORITHM

MAS, according to Gaia, are viewed as being composed of a number of autonomous interactive agents forming an organized society in which each agent plays one or more specific roles. In the analysis phase, Gaia defines the structure of the MAS using the *role model*. This model identifies the roles that agents have to play and the interaction protocols between the different roles. Roles consist of four attributes: *responsibilities*, *permissions*, *activities* and *protocols*. Responsibilities are the key attribute related to a role since they determine the functionality. Responsibilities are of two types: *liveness properties*–the role has to add something good to the system, and *safety properties*–the role must prevent something bad from happening to the system.

Liveness describes the tasks that an agent must fulfill given certain environmental conditions. The activities are tasks that an agent performs without interacting with other agents. Finally, protocols are specific patterns of interaction with other roles. Gaia originally proposed some schemas that could be used for the representation of interactions between the various roles in a system. However, this approach was too abstract to support complex protocols [2]. ASEME [5] moved one step further by

Appears in: *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015), Bordini, Elkind, Weiss, Yolum (eds.), May, 4–8, 2015, Istanbul, Turkey.*

allowing protocols to define the involved roles processes as liveness formulas that would later be included in the liveness of the system role model (a model inspired by the Gaia roles model). This is one assumption of this work, i.e. that the protocols are a send message action, a receive message action or a combination of message send and receive actions and activities for each participating role.

We chose the XML Process Definition Language (XPDL version 2.1), a standard supported by the Workflow Management Coalition (www.wfmc.org) as the target process model language.

The transformation algorithm uses elements from the liveness formulas grammar and the XPDL metamodel. It is a recursive algorithm that takes the liveness formula expression elements from left to right and applies specific predefined templates, gradually building the XPDL process (see Figure 1).

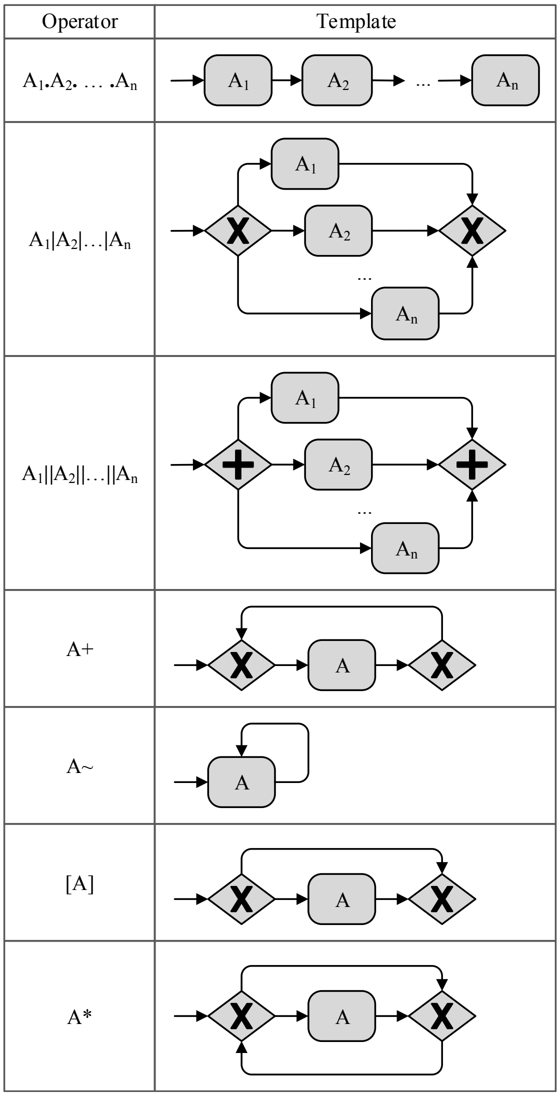

Figure 1. Gaia operator templates for XPDL generation.

We create one XPDL *Pool* instance (pools are lanes in which parallel roles' processes are modeled in XPDL/BPMN) for each transformed role. Then the user defines the associations for message sending and receiving activities aided by the Liveness2XPDL tool that we have developed and offer as free-open source software, which can be browsed at github: https://github.com/ASEMEtransformation/Liveness2XPDL.

3. DISCUSSION AND CONCLUSION

Herein, we showed how the output of the analysis phase model of the Gaia methodology, or its derivatives, e.g. ASEME [6], can be transformed to a process model. Process models are useful paradigms as they allow the usage of a wide range of tools for simulation, thus providing the means to explore non-functional properties of the system under construction. On the other hand, process models are commonly used by business stakeholders, who can now understand and appreciate a MAS analysis model.

In our previous work [1], we provided transformation templates targeting the BPMN v1 metamodel. This work extends that one by targeting the XPDL metamodel and by catering for integrating multiple roles in a single process model.

A very promising path for future work lies in developing a code generation tool based on the process model and targeting the WADE toolkit of the popular JADE platform, which provides support for automatically creating agents that execute tasks defined in XPDL (jade.tilab.com).

4. REFERENCES

[1] Delias, P. and Spanoudakis, N. 2010. Simulating Multi-agent System Designs Using Business Process Modeling. *8th European Workshop on Multi-Agent Systems (EUMAS 2010)* (Paris, France, 2010).

[2] Moraitis, P. and Spanoudakis, N. 2006. The GAIA2JADE Process for Multi-Agent Systems Development. *Applied Artificial Intelligence*. 20, 2-4 (Feb. 2006), 251–273.

[3] Onggo, B.S.S. 2012. BPMN pattern for agent-based simulation model representation. *Proceedings Title: Proceedings of the 2012 Winter Simulation Conference (WSC)* (Dec. 2012), 1–10.

[4] Pascalau, E., Giurca, A. and Wagner, G. 2009. Validating Auction Business Processes using Agent-based Simulations. *Proceedings of 2nd International Conference on Business Process and Services Computing (BPSC2009), March 23-24, Leipzig, Germany* (2009).

[5] Spanoudakis, N. and Moraitis, P. 2008. An Agent Modeling Language Implementing Protocols through Capabilities. *2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology* (Dec. 2008), 578–582.

[6] Spanoudakis, N. and Moraitis, P. 2011. Using ASEME Methodology for Model-Driven Agent Systems Development. *Agent-Oriented Software Engineering XI*. D. Weyns and M.-P. Gleizes, eds. Springer-Verlag. 106–127.

[7] Szimanski, F., Ralha, C.G., Wagner, G. and Ferreira, D.R. 2013. Improving Business Process Models with Agent-Based Simulation and Process Mining. *Enterprise, Business-Process and Information Systems Modeling* (Berlin, Heidelberg, 2013), 124–138.

[8] Wooldridge, M., Jennings, N.R. and Kinny, D. 2000. The Gaia Methodology for Agent-Oriented Analysis and Design. *Autonomous Agents and Multi-Agent Systems*. 3, 3 (Sep. 2000), 285–312.