Ontology-based Validation of Agent Oriented Modelling

A. Lopez-Lorca
G. Beydoun
University of Wollongong
Wollongong NSW 2522 Australia
{aall645, beydoun}@uow.edu.au

L. Sterling
Swinburne University of Technology
Hawthorn VIC 3122 Australia
University of Melbourne
Parkville VIC 3010 Australia
lsterling@groupwise.swin.edu.au
leon@cs.mu.oz.au

T. Miller
University of Melbourne
Parkville VIC 3010 Australia
tmiller@unimelb.edu.au

ABSTRACT
Despite the potential of Multi-Agent Systems (MAS), this technology has not been widely adopted by industry yet. Due to its complexity, errors in modelling activities can be costly. Early validation of MAS models can prevent rework or building a system non-compliant with client's specification. We propose a general ontology-based process to validate any kind of software models that can be adapted in a broad range of software development projects. We illustrate this for MAS development as its complexity justifies additional costs associated with applying our add-on validation process. This work provides early evidence of the soundness of our approach. We successfully validate and improve the quality of MAS models for a real-life development project showing that our MAS models validation process can contribute to harnessing the commercial potential of MAS technology.

Categories and Subject Descriptors
I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence – *Multi Agent Systems*; I.6.4 [**Simulation and Modeling**]: Model Validation and Analysis.

General Terms
Verification.

Keywords
Ontology-based, validation, software model, multiagent system model.

1. INTRODUCTION
Ontologies provide a mechanism for representing domain knowledge to a varying degree of formalism [1]. We advocate the use of ontologies to validate and improve the quality of software workproducts during development processes. As an element of joint development with the user, ontologies can bridge common communication gaps between users and developers. We illustrate using an ontology to check consistency, correctness and completeness of models against initial system requirements. We expect that as intermediary modelling elements, ontologies can facilitate and improve the development of software workproducts, potentially reducing the development and maintenance costs of software systems. We provide a methodology-independent and ontology-based add-on to facilitate the creation of models especially in certain scenarios: In developments of inexperienced modelers, to guide their work and avoid errors; in initial MAS developments of experienced modelers in any other technology, as agents have many important particularities which cannot be found in other paradigms; in projects where the domain is complex or unknown, for experienced and inexperienced modelers alike; in projects dealing with the same domain, to enable reuse of the generated domain knowledge (i.e. ontologies). Whilst the focus of our illustrations is on applying ontologies to improve the development of MAS models, we expect that our approach is easily adaptable to other development paradigms such as agile methods.

Many existing works focus on the use of ontologies to MAS. Of these many focus on the process itself. For example, in [3], a method is given to adapt extreme programming methods to develop a lightweight ontology to help agile development of MAS. It is refined further in [4]. Our focus in this paper is the quality of the MAS workproducts through a domain enriched process rather than the software process itself. Other works use ontologies to assist in the development of workproducts in particular in the detailed design phase. Tran et al. [5] present an ontology-based MAS for the domain of a peer-to-peer (P2P) information sharing community where ontologies are built and used in development-time to create the models and in run-time to exchange information between agents. Although they use domain ontologies during development and run-time, they do not provide detailed support for the validation of MAS, which is the focus of our proposal.

Our approach shares similar goals with the work developed by Brandão et al. [2]. They propose the use of ontologies as a method for the verification of MAS designs. They use an ontology to model the MAS modelling language. These model-diagram mappings enable the automatic validation of the models to check that there are neither intra-model nor inter-model inconsistencies. The main difference with our proposal is that they can validate the models against their theoretical structure and dynamics, but use no information about the specification or application domain. Furthermore, they do not generalise their efforts to outside MAS development and have not validated their proposal properly.

2. AN ONTOLOGY BASED SOFTWARE MODELS VALIDATION ADD-ON
Our proposed ontology-based MAS software models validation (Figure 1) consists of five activities. Our proposal is an add-on to the development process and it is completely independent of the underlying software models or their development methodology.

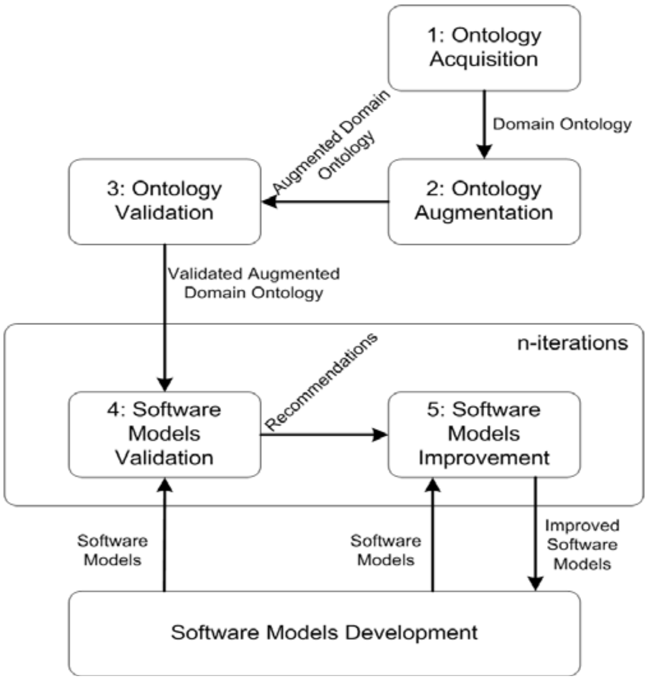

Figure 1. Ontology-mediated software models validation add-on process overview.

In the *Ontology Acquisition* activity a suitable ontology is retrieved from an existing repository, otherwise one is built using the most suitable ontology engineering techniques. In the *Ontology Augmentation* activity, the ontology is augmented to represent features related to the chosen development paradigm. Domain concepts are annotated to link them to paradigm concepts and relations between them are created according to existing relations defined for the paradigm. In our case study we identify the MAS terms *Goal*, *Role*, *Activity*, *Environment* and *Agent* and relations between them such as *Role responsible for Goal*, *Agent plays Role* or *Activity follows Activity*. In the *Ontology Validation* activity, members of the development team validate the ontology with the client to reach a common understanding and compliance to client's conceptualisation. In the *Software Models Validation* activity, the models are validated against the augmented ontology. This activity provides the control element for new iterations. A new iteration will be necessary as long as any non-trivial recommendation is made to improve the quality of the models. In the *Software Models Improvement* activity, the recommendations are analysed by the developers to choose which to apply and which to ignore. After improving the quality of the models according to chosen recommendations, the new set of models will be validated in the next iteration. In our case study we validate and improve goal, agent, interaction, scenario, organisation, role and environment models for MAS development.

Development proceeds with each iteration further along the sequence of workproducts required by the chosen methodology. The development and validation of the software models are intertwined and done concurrently. Problems of reviewed models are fixed before their full development. Any models yet to be commenced in that iteration will take advantage of the recommendations avoiding compounded errors. This has been proven in our case study, as recommendations made in the first iteration to the interaction model were used to prevent errors in the development of the scenario during the second iteration.

3. CONCLUSION AND FUTURE WORK

We apply ontologies to validate and improve the quality of software models. We take into account the domain as specified by the client's requirements, bridging any communication gap between clients and developers. Models are validated as soon as they are available, fixing errors as they arise and avoiding compounding and propagating errors to later phases of the development. To integrate our validation add-on seamlessly into the development process, we use an iterative, incremental and concurrent development process. The process iterates over intermediate versions of the model to achieve high quality. It is incremental in nature, not all the models are considered during each iteration. It is concurrent as development and validation activities overlap.

Applying our process can incur additional development cost and requires a cost justification. It is particularly appealing in critical software application where errors can be very costly and disastrous. This cost overhead may also be justified in the scenarios described in Section 1. That said, the cost of the validation can be greatly reduced by more effective reuse of existing ontologies. With advent of the Semantic Web, more ontologies are made available. More importantly, there is a great scope for generating the amendment proposals automatically, harnessing automatic reasoning capabilities of ontologies. Indeed, we are now studying this possibility with the expectation to develop a tool that can significantly alleviate the burden of the details of the ontology-mediated validation process. In the future, we also intend to apply the ontology-mediated software model validation process to further cases studies to fine-tune it and to test our forthcoming tool.

4. REFERENCES

[1] Beydoun, G., A. Hoffmann, J.T. Fernández-Breis, R. Martínez-Béjar, R. Valencia-García and A. Aurum. *Cooperative Modeling Evaluated.* International Journal of Cooperative Information Systems. 2005. **14**(1): p. 45-71.

[2] Brandão, A.A.F., V.T.d. Silva and C.J.P.d. Lucena, *Observed-MAS: An Ontology-Based Method for Analyzing Multi-Agent Systems Design Models*, in *Agent-Oriented Software Engineering VII*. 2007. Springer Berlin / Heidelberg. p. 122-139.

[3] Hristozova, M. and L. Sterling. *An eXtreme Method for Developing Lightweight Ontologies*, in *Workshop on Ontologies in Agent Systems, at the International Join Conference on Autonomous Agents and Multi-Agent Systems*. 2002. Bologna, Italy: ACM.

[4] Lister, K., M. Hristozova and L. Sterling. *Reconciling Implicit and Evolving Ontologies for Semantic Interoperability*, in *Ontologies for Agents: Theory and Experiences*, V. Tamma, Cranefield, S., Finin, T., Willmott, S, Editor. 2005. Birkhäuser Basel. p. 121-144.

[5] Tran, N., G. Beydoun and G. Low. *Design of a Peer-to-Peer Information Sharing MAS Using MOBMAS (Ontology-Centric Agent Oriented Methodology)*, in *International Conference on Information Systems Development*. 2006. Budapest, Hungary: Springer.