Int. J. Human-Computer Studies 87 (2016) 20–37

Contents lists available at ScienceDirect

Int. J. Human-Computer Studies

journal homepage: www.elsevier.com/locate/ijhcs

Supporting agent oriented requirement analysis with ontologies ☆

Antonio A. Lopez-Lorca [a], Ghassan Beydoun [a,*], Rafael Valencia-Garcia [b], Rodrigo Martinez-Bejar [b]

[a] University of Wollongong, Australia
[b] University of Murcia, Spain

ARTICLE INFO

Article history:
Received 29 June 2014
Received in revised form
30 October 2015
Accepted 30 October 2015
Communicated by S. Staab
Available online 10 November 2015

Keywords:
Requirements
Validation
Ontology
Ontology modelling
Multi agent systems

ABSTRACT

Requirements analysis activities underpin the success of the software development lifecycle. Subsequent errors in the requirements models can propagate to models in later phases and become much costlier to fix. Errors in requirement analysis are more likely in developing complex systems. Particularly, errors due to miscommunication and misinterpretation of a client's intentions are common. Ontologies relying on formal descriptions of semantics have often been used in multi agent systems (MAS) development to support various activities and generally improve the complex systems produced. However, their use during requirements analysis to validate match with the client's conceptualisation is largely unexplored. This article presents an ontology driven validation process to support requirement analysis of MAS models. This process is underpinned by an agent-based metamodel that describes commonly used informal agent requirement models. The process concurrently and incrementally validates the informal MAS requirement models produced. The synthesis of the process is first justified and illustrated in a manual tracing of the process. The paper then describes an interactive support tool to harness the formal semantics of ontologies and by pass the costly manual effort. The validation process is evaluated and illustrated using three case studies.

1. Introduction

Requirement engineering activities leading into the design phase are typically sequenced as follows: elicitation from users, analysis, negotiation between stakeholders, verification and validation of resultant models [Sadraei and Aurum, 2007]. Throughout these activities, cognitive limitations of modellers, communication gaps with the users, together with the interdependencies of models, can lead to errors. These errors can snowball in the course of the development process and become more costly to fix [Westland, 2002]. As such, capturing errors in the requirement models during the verification and validation activities before they propagate into the design and later phases is often pursued. Formal technical reviews, involving developers and users, are commonly undertaken to validate requirements [Pressman, 2005]. This involves jointly examining the specification looking for missing information, inconsistencies, conflicting or unrealistic requirements. This can be effective but it is highly time-consuming and can therefore be expensive.

This research focuses on semi-automating the verification and validation activities of requirement models in agent oriented software engineering (AOSE). We do not advocate the use of formal languages to express requirement models. Their formality can support automation but also tends to hinder the communication with the client. This could ultimately have a negative impact on the quality of the models [Miller et al., 2011]. We instead advocate an innovative combination of reusable domain ontologies and an agent oriented metamodel. Combining these two will enable a tool to automatically validate agent informal requirement models. We present and evaluate a requirements validation process for MAS models. The process assumes an existing AOSE methodology. The process gets coupled with the chosen methodology. Many authors have proposed AOSE methodologies targeting various types of applications [Low et al., 2009]. Each methodology typically requires a different set of models to define the system [Tran and Low, 2005]. Our process can support many of the extant AOSE methodologies at a requirements engineering level. The models supported by the process will cover most extant methodologies. We identify the most prominent requirements modelling constructs across AOSE methodologies and we support them accordingly in our process.

☆This paper has been recommended for acceptance by Henrik Iskov Christensen.

* Corresponding author. Tel.: +61 2 4221 4037; fax: +61 2 4221 4045.

http://dx.doi.org/10.1016/j.ijhcs.2015.10.007
1071-5819/

The process is semi-automated. It exploits automated reasoning using two ontologies: a domain ontology and a MAS structure ontology (aka AOSE metamodel). However, the process still requires additional effort by the analysts. This effort is justified in complex modelling applications in uncertain environments where distributed adaptive systems such as MAS are sought. As we shall see, the process is also supportive in cases where the clients conceptualisation of the domain is also evolving. The process enables the validation of informal MAS models, as typically used within AOSE methodologies, using domain ontologies which can formally represent a domain [Guarino, 1998]. The bridge between the formal ontologies and the informal models consists of agent based annotations that extend the ontology content to enable cross checking against the components of informal model. That is, the domain ontology is annotated with AOSE constructs. Annotated domain ontologies are then used to check the semantics of the models produced during analysis. The annotations enable an automatic reasoner to identify the informal modelling components that can be checked against the annotated ontology components. Discrepancies between the ontology and the models highlight potential errors that need to be addressed before they propagate further in the modelling process. The validation process intertwines with the modelling process, validating and verifying models as they spiral toward acceptance and completion. The domain ontology and the clients' conceptualisations are also expected to evolve in the process. The ontology annotation is supported by an AOSE metamodel that describes the most common features of MAS. It is based on the FAML metamodel [Beydoun et al., 2009] that is proved to adequately cover over 20 existing AOSE methodologies.

This research extends earlier work in Lopez-Lorca et al. [2011]. We illustrate the manual execution of the process and we then automate harnessing the reasoning opportunities offered by the reuse of formal domain ontologies. The validation process and its automatic support are evaluated using three different development case studies of significant size highlighting the impact and the gains of the automation. In the next section, we review related work.

2. Related work

As earlier outlined, the following four activities in requirements engineering that lead into the design phase of a system: *Elicitation* of high level goals of the target system, requirements for different groups of users, along with system boundaries [Girardi and Leite, 2008]; *Analysis* of the requirements to uncover conflicts, ambiguities, missing or duplicate requirements; *Negotiation* between trade-offs to achieve agreement between stakeholders; *Verification and Validation* to find any deficiencies in consistency, accuracy and adequacy of requirements. The work in this paper focusses on supporting the validation of requirement models through using additional domain ontologies. Recent work has shown that appropriate ontologies can be identified for reuse based on indexing information extracted from the requirement models themselves [Bicchierai et al., 2013; Beydoun et al., 2014]. As the focus of the research is on the activities of "verification and validation", we restrict our discussion to works related to employing reusable ontologies to support those requirement activities. We adopt the definition in Bahill and Henderson [2005], validating requirements is the act of ensuring (1) the set of requirements is correct, complete, and consistent, (2) a model can be created that satisfies the requirements, and (3) a real-world solution can be built and tested to prove that it satisfies the requirements. The work here uses OWL to represent the ontology but the models themselves can be expressed in various notations. The use of OWL without insisting on any specific notation makes our approach easier to adapt than other approaches [Guizzardi and Wagner, 2005; Shanks et al., 2003]. We use off-the-shelf OWL reasoning to enable validation of various relations between model entities. This enables checking for consistency between the models and the domain ontology, amongst the models themselves and also supports the completion of the requirement models through inferring new relations. The proposal here can be adapted to various modelling paradigms, by reformulating the validation operators. The process to be outlined in the next can thus be seen as independent of the modelling language. However, adapting the proposal to different methodologies requires significant adaptation and this paper focusses on the agent-based modelling paradigm. It brings the use of ontologies to the centre of a model driven process as also has been recently advocated for distributed service oriented systems [Shen et al., 2014].

Various works since the 80s have advocated supporting requirement analysis through additional knowledge sources. A notable early work on this is the Telos language which is designed to create knowledge bases to support requirements and systems development [Mylopoulos et al., 1990]. Telos was designed to capture knowledge about both the application domain and the software engineering process leading to the system. Indeed, the motivation of such work overlaps with ours in the use of knowledge bases and automated reasoning to support information systems development. However, we are more specific towards supporting particular activities in the requirement analysis phase. Furthermore, we demand less knowledge engineering to deploy the knowledge and the automated reasoning. For instance, in addition to the application knowledge, the use of Telos requires four knowledge engineering requirements: system world, usage world, usage and development world. Our approach harnesses the power of reuse more effectively through reuse of existing ontologies, through focussing on specific requirement analysis activities and by focussing on specific type of systems (multi agent systems). We reuse a metamodel that encompasses knowledge about the agent models involved. By reusing an off-the-shelf, well accepted knowledge representation language, OWL, and its reasoning tools, we automatically combine these two sources of knowledge automatically in the annotated ontology.

Combining the use of a metamodel reflecting the architecture of the sought system with the use of an ontology to reflect the requirements is not new. Notable examples are Aldewereld et al. (2011) and Siegemund et al. (2011). In Aldewereld et al. (2011), the motive in that work was to enable a Model Driven Engineering approach for generating a MAS. In our approach, the motive is to enhance the quality of the outcome of the requirement analysis and gathering processes. In our work, a metamodel is used to guide how the ontology can be cross-checked against the agent models. In this context, our work not only supports development of MAS in cases where the requirements are evolving or shifting, but it also supports developers who are new to agent modelling. Siegemund et al. [2011] also build an ontology defining a requirement metamodel for Goal Oriented Requirements Engineering (GORE). Their metamodel includes the elements that define the GORE process as classes and relationships. Each particular project would instantiate the domain independent metamodel. This infrastructure makes it possible to check the consistency and completeness of the requirements. Consistency rules are defined as ontology axioms and are automatically checked by the reasoner. The metamodel presented in this article defines the concepts and relationships used in the models (well-structured) of AOSE methodologies while theirs models a general requirements specification (mostly textual descriptions) for GORE. While the consistency rules are similar in both cases, the completeness rules developed as part of this research are different. Our work