Information Systems 46 (2014) 45–60

Contents lists available at ScienceDirect

Information Systems

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

Identification of ontologies to support information systems development

Ghassan Beydoun [a,*], Graham Low [b], Francisco García-Sánchez [c], Rafael Valencia-García [c], Rodrigo Martínez-Béjar [c]

[a] *School of Information Systems and Technology, University of Wollongong, Australia*
[b] *School of Information Systems, Technology and Management, University of New South Wales, Australia*
[c] *Faculty of Computer Science, University of Murcia, Spain*

ARTICLE INFO

Article history:
Received 6 February 2012
Received in revised form
3 November 2013
Accepted 5 May 2014
Recommended by F. Carino Jr.
Available online 28 May 2014

Keywords:
Information systems development
Ontologies
Early requirements
Ontology retrieval
i* Models

ABSTRACT

Ontologies can provide many benefits during information systems development. They can provide domain knowledge to requirement engineers, are reusable software components for web applications or intelligent agent developers, and can facilitate semi-automatic model verification and validation. They also assist in software extensibility, interoperability and reuse. All these benefits critically depend on the provision of a suitable ontology (ies). This paper introduces a semantically-based three stage-approach to assist developers in checking the consistency of the requirements models and choose the most suitable and relevant ontology (ies) for their development project from a given repository. The early requirements models, documented using the i* language, are converted to a retrieval ontology. The consistency of this retrieval ontology is then checked before being used to identify a set of reusable ontologies that are relevant for the development project. The paper also provides an initial validation of each of the stages.

1. Introduction

Ontologies provide a mechanism for representing domain knowledge to a varying degree of formalism [1]. Guarino (1998) defines an *"ontology driven information system"* as a system in which the ontology is an integral component of the system and is used at run time to ensure that the system achieves its goals and functionality. Ontologies can also be used at design time by software developers [2]. In fact, *ontology*-driven information system development is another recently coined term referring to the use of ontologies as a central artifact in information systems development [3]. For example, ontologies can be read by future users of a system, and as a joint development element with the user, they can be used to validate and improve the quality of software work products during various phases of the development process [4]. They can improve the outcome of various requirement engineering activities. For example, they can improve elicitation by bridging common communication gaps between users and developers (e.g. [5]). They can also be used to improve the requirements models by supporting a dedicated verification and validation requirement engineering activity [6].

Ontologies can also play a prominent role in expediting software development knowledge reuse. This includes reuse of artifacts as well as development knowledge and work products [7]. As attention is increasingly paid to higher level reuse issues (beyond code reuse), (e.g. reuse of models and reuse of project development and management knowledge [8]), ontologies are emerging as a promising vehicle in delivering much touted promises of runtime and design time flexibility in new paradigms such as software-as-a service and Service Oriented

[*] Corresponding author.
E-mail address: beydoun@uow.edu.au (G. Beydoun).

http://dx.doi.org/10.1016/j.is.2014.05.002
0306-4379/© 2014 Elsevier Ltd. All rights reserved.

Architectures. They also have been proposed as a conduit to the reuse of design models and developers' skills by having them as central constructs driving the whole of the Software Development Life Cycle [8].

Benefits of ontologies, in development and in reuse, are often predicated on identifying appropriate domain ontology (ies) that are readily available to software developers. These can then guide software work product modeling and verification. Ontologies can also facilitate the interoperability of work products and the continuing operation of a correct system [9].The focus of this paper is on identifying appropriate ontologies for information systems developers based on early system requirements to enable subsequent ontology-based information systems development. The paper provides a three stage approach that validates and uses early requirement models to identify suitable ontologies to support the development of a given application. Often a single ontology from a given repository may not be semantically adequate. A subsequent integration of a number of ontologies may therefore be necessary to ensure the semantic coverage of the application domain. Our proposed approach identifies suitable ontologies which may be subsequently integrated and adapted for the software project at hand. This then acts as a filter that can be used to retrieve a relevant set of ontologies. The approach consists of the following stages:

- Stage 1: develop an intermediate retrieval ontology. A theoretical mapping converts early requirements models into an intermediate retrieval ontology.
- Stage 2: perform consistency check of the intermediate retrieval ontology and refine the early requirements models (if necessary).
- Stage 3: compare domain ontologies with the intermediate retrieval ontology to identify a set of relevant ontologies.

Our work is based on the insight that, whilst they may be ambiguous, incomplete and/or inconsistent, early requirements expressed informally can robustly be used to generate a formal ontology that can be used to recommend an ontology from a repository that “best” matches the early requirements. This ontology may also suggest modifications and/or additions to the early requirements. With subsequent processing, ontologies retrieved using early requirement models can reliably support the development of the work products of a system with the added bonus that ontology-based work products are reusable components and can themselves be stored and indexed for later information systems development activities. Thus this paper automatically integrates early requirement models into ontology-driven information systems development.

The work presented in this paper ensures that information system developers are provided with a set of supporting relevant ontologies that can be used to underpin the development of the whole system. The approach identifying the ontologies is theoretically grounded. Its effectiveness and reliability as a cornerstone for ontology-based system development is demonstrated using a case study example. The rest of this paper is organized as follows: Section 2 discusses related work with emphasis on recent efforts to integrate the use of ontologies within the Information Systems development lifecycle. Section 3 presents the conceptual underpinning of how we use requirement models to retrieve supporting ontologies for IS development. An initial validation of our proposed approach is presented in Section 4. Finally Section 5 concludes with a discussion of limitations and future extensions of the work.

2. Related work

The use of ontologies for general software development on the basis that this produces better quality models is not a new idea. Ontologies impact on information systems in both the temporal and structural dimensions [3]. The temporal dimension refers to whether the ontology is used at development time or run time while the structural dimension considers how an ontology could affect the IS components (e.g. database, user interface and/or application program). At development time, the use of ontologies typically involves reusing ontologies organized into libraries of domain, task or generic ontologies. They typically assist in ensuring system correctness. For example, ontologies have been used to decide what models should be included in the Model Driven Architecture for a system (e.g. [10]). Ontologies are generally easier to understand than most analysis and design models that require specific and in-depth methodological knowledge. They have been advocated as intermediary elements to support the development of analysis models [5]. A methodology-independent technique is presented to use ontologies in support of the validation processes in [11] to facilitate the creation of models for inexperienced modelers or to assist more experienced ones detecting and resolving errors.

The use of ontologies in our work to be detailed in Section 3 is multifaceted (see Fig. 1 in Section 3). An ontology is used as a knowledge filter (in Stage 1 and Stage 3), a consistency check platform of requirement models (in Stage 2) and a central artifact in an ontology-centric methodology (beyond Stage 3, as later elaborated in Section 5). The filtering aspect of our work in Stage 3 is similar to [12], where an ontology filters relevant mathematical models for adaptive software. Our matching in Stage 3 is syntactic rather semantically driven as in [12]. However Stage 3 is not the main contribution in this paper. Our key contribution is how the ontology is generated in Stage 1. A mapping filters through a set of concepts and relationships from the requirement model(s) to form a retrieval ontology. This automatically generated (formal) ontology is then used as a filter for relevant ontologies (in Stage 3). The (formal) retrieval ontology is first exploited to execute a consistency check of the requirement models (Stage 2). This heeds earlier proposals to specify a suitable conceptual modeling language for a domain to improve the quality of models (e.g. [13]). Our work is similar to the work in [5] which uses ontologies to validate and verify software models. However our work here is innovative in automatically providing such a conceptual model from the early requirement models. More recently [14] proposed a tool to build and automatically verify conceptual models