Computers in Industry 65 (2014) 1301–1315

Contents lists available at ScienceDirect

Computers in Industry

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

An ontology change management approach for facility management

Perrine Pittet *, Christophe Cruz, Christophe Nicolle

Le2i Laboratory, UMR CNRS 6306, University of Burgundy, Dijon, France

ARTICLE INFO

Article history:
Received 1 July 2013
Received in revised form 30 May 2014
Accepted 11 July 2014
Available online 18 September 2014

Keywords:
Ontology change management
Ontology evolution
Building information modeling
Facility management
$\mathcal{SHOIN}(\mathcal{D})$
OWL DL

ABSTRACT

Facility management (FM) or technical property management is an approach to operate, maintain, improve and adapt buildings and infrastructures of organizations. A FM project requires the cooperation of many actors from different domains so it has to be automated in a constrained collaborative environment. This paper proposes a new approach for ontology change management applied on facility management of such projects. The industrial challenge is, firstly, to ensure consistency of a FM project knowledge from the construction phase to the technical property management phase (after delivery). Secondly, it has to provide to each actor of the project a personal up-to-date "view" of the building knowledge related to its business profile and allow its evolution. The scientific approach, called OntoVersionGraph, is a change management methodology for managing ontology life cycle including ontology evolution and versioning features, in conjunction with contextual view modeling. Its contribution is the impact management of changes between the ontology and its different views.

1. Introduction

Facility management (FM) or technical property management is an approach to operate, maintain, improve and adapt buildings and infrastructures of organizations [24]. In practice, it can cover a wide range of services including property management, financial management of the building, management of human resources related to the building, health and associated risks of the building, maintenance (domestic services such as cleaning, security. . .) and supplies management. A FM project requires the cooperation of many actors from different domains such as architects, civil engineer, plumbers. . . During such projects, teams set up many business processes independently from other teams in other domains.

To achieve facility management of such projects, the facility manager needs powerful tools to organize the knowledge produced by each of the actors during the building life cycle. The creation of an information system dedicated to the use of facility management requires enormous phases of knowledge modeling. In addition, the knowledge of the building changes during its life cycle in response of the different types of information generated by the actors of the project linked to the MOP Law ("loi de Maîtrise d'Ouvrage Publique" defined by the French government, stating authority control and project management relationships for the public market since 1985). Consequently modeling the facility management of a building during its life cycle addresses the problem of heterogeneity of information exchanged between the actors. It demonstrates the need to homogenize the representation of these exchanges with the building knowledge all along its lifecycle.

Modeling building information during its life cycle is the aim of the BIM (Building Information Model) technology. It aims at facilitating integration, interoperability and collaboration in the building industry. BIM can be defined as the process of generating, storing, managing, exchanging, and sharing building information in an interoperable and reusable way. To enable building information sharing, Industry Foundation Classes (IFCs) are the standard that specifies object representations for FM projects (Vanlande et al., 2003); they include object specifications, or classes, and provide a useful structure for data sharing among applications. A BIM system is a tool that enables users to integrate and reuse IFC building information and domain knowledge throughout the lifecycle of a building. However, the main feature of a BIM system described by an IFC file is the 3D modeling of a building with data management, data sharing and data exchange during the building life cycle. To resolve the heterogeneity issues of facility management and ensure a complete interoperability between actors of the project, a semantic layer is missing.

The semantic heterogeneity issue has been addressed by the Semantic Web, which marks a shift from publishing data in human

* Corresponding author. Tel.: +33 380396857.
E-mail addresses: perrine.pittet@u-bourgogne.fr, perrine.pittet@checksem.fr (P. Pittet), christophe.cruz@u-bourgogne.fr (C. Cruz), cnicolle@u-bourgogne.fr (C. Nicolle).

http://dx.doi.org/10.1016/j.compind.2014.07.006
0166-3615/

readable HTML documents to machine-readable documents. This Web 3.0 technology has provided a powerful tool to represent the knowledge of a domain with the aim of making heterogeneous information understandable and easy to process by both human and machine: ontology. Nevertheless, ontology, to be useful at the industrial level must not only be used as a meta-model. It has to focus on the project content to help exploiting its business processes through semantic representation. Besides mechanisms of ontology evolution can help maintaining the consistency of this knowledge throughout the project life cycle.

The problem is then the following for facility management: how to build a long-term vision of knowledge taking into account the life cycle and the heterogeneity of the actors of the building?

This paper proposes a new approach for ontology change management applied on the facility management domain. The industrial challenge is, firstly, to ensure consistency of a FM project knowledge from the construction phase to the technical property management phase (after delivery). Secondly, it has to provide to each actor of the FM project a personal "view" of the building knowledge related to its business profile. The scientific approach is based on a methodology for managing ontology life cycle including ontology evolution and versioning features, in conjunction with contextual view modeling. The major contribution is the realization of a versioning system exploiting evolution logs as storage model, for evolving and maintaining the ontology consistency, and extracting and managing ontology views. The main results are the improvement of an innovative collaborative platform dedicated to facility management and the foundation of a scientific approach for ontology change management based on evolution logs. This is a real revolution on the market of uses. The methodology has been implemented in the OntoVersioning API, extending the Java Jena API dedicated to OWL ontology management with the change management features cited above.

This paper is articulated in two main sections. Section 2 presents a state of art on ontology change management and a discussion on the related deadlocks of the existing proposals regarding contextual and user-centered systems like facility management systems. A state of art on ontology views gives the further objectives change management should try to reach to bridge these gaps. Section 3 describes our proposal called OntoVersionGraph, which is an ontology change management methodology formalized for change management of $\mathcal{SHOIN}(\mathcal{D})$ ontologies in contextual and user-centered systems. The $\mathcal{SHOIN}(\mathcal{D})$ description logic was chosen to formalize OntoVersionGraph because description logics are logics dedicated to knowledge representation and this particular logic has the same semantics as the OWL DL ontology language, which is massively used to represent formal ontologies. However, the methodology can be extended to other description logics by undertaking more or less language constructors for change modeling. The methodology process is applied to resolve a facility management common issue implying evolution of a BIM ontology impacting different views and actors of an FM project.

2. Change management state of art and discussion

This section aims at identifying the deadlocks of ontology change management regarding collaborative and user-centered systems based on ontologies such as FM projects and introducing the solution we chose for our methodology. The first part presents a state of art on ontology life cycle and its management. The second part lists the common issues identified by the literature concerning change management of ontologies in collaborative systems, like the need of specialization or profiling such ontologies. The third part deals with the ontology view solution to bridge this gap in the context of FM projects multi-user ontology management.

2.1. From ontology life cycle to change management: a state of the art

In recent years, building ontologies are gaining ground to provide to the Semantic Web clear semantics in an agreed, consistent and shared encodings. Actually, ontologies make possible to application, enterprise, and community boundaries of any domain to bridge the gap of semantic heterogeneity. Ontology development, to be correctly achieved, requires a dynamic and incremental process [1]. It starts with a rigorous ontological analysis [2] that provides a conceptualization of the domain to model agreed by the community. The ontology, specified in a formal language, approximates the intended models of the conceptualization [3]: the closer it is the better it is. The ontology needs to be revised and refined until an ontological commitment is found. Ulterior updates of the ontology, addressed by ontology evolution, aim at responding to changes in the domain and/or the conceptualization [4]. Changes are consequently inherent in the ontology life cycle. Ref. [5] defines an ontology change as an action on an ontology resulting in an ontology that is different from the original version. Changes help in incorporating new features by improving the ontology conceptualization, however their application can generate inconsistencies.

An ontology becomes inconsistent when its conceptualization and specification does not respect the constraints of the model and axioms of this ontology (Haase and Stojanovic [6]). Two main inconsistency types can affect the ontology: structural and logical. Structural consistency maintenance implies a change modeling respecting the language constructors, which are structural constraints. Logical consistency maintenance deals with the respect of logical constraints in order to protect the ontology semantics. In the literature, ensuring structural and logical consistent updates of ontologies during their lifecycle is the activity studied by the Ontology Evolution research domain.

A couple of ontology evolution methodologies have been proposed like [7–9]. Among them, the AIFB methodology [9], which is one of the most popular, identifies 6 phases to ensure the quality of the ontology evolution process: detection, representation, semantics, implementation, propagation and validation. The validation of one evolution process iteration gives birth to an evolved version of the ontology. However, the AIFB process as such does not take into account the management of the different versions of the ontology generated all along its lifecycle. Accessing different versions of an ontology, rollback to a previous one or comparing two of them are features that are provided by the ontology versioning studies. Ontology versioning refers to the capacity of managing ontology evolution by creating and managing its different versions (Plessers and De Troyer [10]). A versioning system is often based on the use of evolution or versioning logs to trace changes applied at each evolution iteration. Such logs can usually be queried to retrieve and compare ontology versions in a transparent way.

Ontology evolution and versioning are so linked that Ref. [5] introduces the notion of change management to define the interactions between these two activities. Change management combines ontology evolution and versioning features to manage ontology changes and their impacts. Following these guidelines, numerous change management methodologies are proposed in the literature.

2.2. Change management related works deadlocks

Ontology change management systems (OCMS) are direct implementation of these change management methodologies. Since 2007, many works have combined ontology evolution and versioning into OCMS [4,11–17]. The evolution subject has been deeply studied in these works. They especially addressed the