International Journal of Information Management 30 (2010) 559–566

Contents lists available at ScienceDirect

International Journal of Information Management

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

Ontology management and evolution for business intelligence

Alexander Mikroyannidis*, Babis Theodoulidis

Manchester Business School, University of Manchester, Manchester M13 9SS, UK

ARTICLE INFO

Keywords:
Information management
Business intelligence
Ontology
Ontology management
Ontology evolution

ABSTRACT

The amount of heterogeneous data that is available to organizations nowadays has made information management a seriously complicated task, yet crucial since this data can be a valuable asset for business intelligence. Ontologies can act as a semantically rich knowledge base in systems that specialize in information management. The present work investigates the potential of ontologies in supporting the information lifecycle within a corporate environment for business intelligence. The paper demonstrates the use of Heraclitus II, a framework that employs ontology management and evolution in the context of information management systems. The capabilities of the framework in facilitating information management and business intelligence are evaluated through a real-life case study from the life sciences industry.

1. Introduction

The rate of growth in the amount of information available nowadays within a corporate environment poses major difficulties as well as challenges in decision making. Business intelligence (BI) consists of a collection of techniques and tools, aiming at providing businesses with the necessary support for decision making. Examples of simple BI services that already exist are various search and filtering services, as well as various content providers and aggregators that deliver semi-custom information bundles to particular users. On a more sophisticated level, information management (IM) can assist a manager in monitoring specific organizations, technologies, or areas of research, as well as being able to analyze primary data in order to draw conclusions at the level of the company's competition, sector or industry.

Ontologies are a key enabling technology for IM, as they offer information a common representation and semantics. They constitute "a shared and common understanding of a domain that can be communicated between people and application systems" (Davies, Fensel, & Harmelen, 2003). An ontology comprises a formal description of a certain domain, by defining the ontology objects (or entities) that characterise the domain, namely concepts (or classes), as well as their instances and relations. Ontologies express information in a machine-processable form, thus allowing for its efficient manipulation by software agents. They are commonly represented with the use of XML-based languages, such as RDFS (www.w3.org/TR/rdf-schema) and OWL (www.w3.org/TR/owl-features).

Ontologies provide to IM systems a semantically rich knowledge base for interpretation of unstructured content. Based on the semantics encoded within ontologies, information can be extracted from natural language texts and, on a further level of processing, knowledge can be discovered that will assist BI. Nevertheless, the way ontologies are usually managed within IM systems is unsophisticated and disregard important factors. Ontology layering or integration is rarely used and the dynamic aspect of ontologies, which requires appropriate evolution mechanisms, is often neglected. Overall, the potential of ontologies in IM and BI has yet to be fully realized and put to practical use.

The Heraclitus II framework (Mikroyannidis & Theodoulidis, 2006) considers ontologies as a semantically rich knowledge base for information management and proposes a methodology for the management and evolution of this knowledge base. In this paper, we demonstrate the application of Heraclitus II on a real-life case study related to BI in the life sciences industry. We evaluate the framework by examining the challenges posed by the case study and how these are met by Heraclitus II.

The remainder of this paper is organized as follows: Section 2 examines existing approaches in IM, as well as ontology management and evolution. Section 3 provides a brief overview of the Heraclitus II framework. In Section 4, the application of the framework on the case study is presented and in Section 5 the framework is evaluated. Finally, the paper is concluded and some plans for future work are provided.

* Corresponding author. Current address: Knowledge Media Institute, The Open University, Milton Keynes MK7 6AA, UK.
E-mail addresses: A.Mikroyannidis@alumni.manchester.ac.uk (A. Mikroyannidis), B.Theodoulidis@manchester.ac.uk (B. Theodoulidis).

0268-4012/$ – see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ijinfomgt.2009.10.002

2. Related work

Information management (IM) can be defined as "the process of managing information as a strategic resource for improving organizational performance" (Chaffey & Wood, 2005). Modern IM systems have to deal with unstructured data of considerable volume, combining methodologies from numerous disciplines, such as Information Extraction, Natural Language Processing, Information Retrieval, and Data Mining.

GATE (general architecture for text engineering) (Cunningham, Maynard, Bontcheva, & Tablan, 2002) is an architecture for language engineering, containing a suite of tools for language processing and information extraction. The GATE data flow consists of a pipeline of Processing Resources (lemmatisers, part-of-speech taggers, etc.) that run in series and utilize Language Resources (ontologies, lexicons, corpora, etc.). GATE provides a common object-oriented model of ontologies and a unified API for their use by Processing Resources (Bontcheva, Tablan, Maynard, & Cunningham, 2004). Ontologies of different formats, such as RDF(S) and OWL, need to be converted into this model so that they can be used within the GATE pipeline.

Ellogon (Petasis, Karkaletsis, Paliouras, Androutsopoulos, & Spyropoulos, 2002) is a multi-lingual, cross-platform, general-purpose text engineering environment. Ellogon is based on a modular architecture that allows reuse of its sub-systems in order to facilitate the creation of applications targeting specific linguistic needs. APIs for creating processing components are provided for a wide range of programming languages, including C, C++, Java, Tcl, Perl and Python. Although ontologies are not part of the standard Ellogon architecture, they can be added through external components that use the Ellogon APIs.

The unstructured information management architecture (UIMA) (Ferrucci & Lally, 2004) is a software architecture for developing and deploying unstructured information management (UIM) applications. A UIM application can be generally characterized as a software system that analyzes large volumes of unstructured information in order to discover, organize, and deliver relevant knowledge to the end user. UIM applications utilize a variety of technologies including statistical and rule-based natural language processing, information retrieval, machine learning, ontologies, and automated reasoning.

Whereas the aforementioned systems can be mostly regarded as generic architectures for the deployment of IM components, the Parmenides framework (Mikroyannidis, Theodoulidis, & Persidis, 2006) aims in providing managers with specialized tools that will aid decision making, via a customized analysis of the desired market. Ontologies play a crucial role in the Parmenides architecture. The knowledge base used for document processing and knowledge discovery is a collection of ontologies. In particular, these ontologies are utilized by the information extraction engine for semantic annotation of documents, as well as by knowledge discovery tools for data pre-processing tasks.

The approaches adopted in most IM systems regarding ontology management are quite simplistic. Even when multiple ontologies are used, these are not integrated but each is managed separately. In addition, the evolution of ontologies for alignment with changing business requirements is usually overlooked. Some of these limitations are addressed by systems specializing in ontology management and evolution, such as Sesame, OMM, OntoView, PROMPT, and KAON.

Sesame (Broekstra, Kampman, & Harmelen, 2002) is a generic architecture for the storage and querying of RDF and RDFS ontologies. Sesame can be coupled with a variety of repositories for the storage of ontologies, including relational databases, RDF triple stores, or remote storage services on the web. RQL, a declarative query language, is used for querying RDF data at a semantic level. The ontology middleware module (OMM) (Kiryakov, Simov, & Ognyanov, 2002) has been built on top of Sesame to handle version control and change management. Sesame and OMM have been integrated to formulate the ontology management system of the On-To-Knowledge project (www.ontoknowledge.org).

OntoView (Klein, 2004) is a web-based environment for ontology version control. It was initially designed to compare RDF ontologies, but was later extended to include other representations as well. OntoView maintains not only the transformations between different versions of ontologies, but also the conceptual relations between concepts in these versions, thus making them interoperable.

PROMPT (Noy & Musen, 2003) is a suite of tools for ontology versioning and integration that have been implemented as extensions for the Protégé ontology editor (http://protege.stanford.edu). These include iPROMPT, an interactive ontology-merging tool, AnchorPROMPT, a graph-based tool for finding similarities between ontologies, PROMPTDiff, a tool for finding differences between two versions of the same ontology, and PROMPTFactor, a tool for extracting parts of an ontology. These tools share user interface components, heuristics, data structures, and provide information for one another. The Protégé ontology editor provides overall capabilities for managing multiple ontologies.

The KArlsruhe ONtology and Semantic Web tool suite (KAON) (Haase et al., 2004) provides a number of open source software modules for the engineering, management, and evolution of ontologies. KAON modules include Description Logic Programs that are responsible for ontology reasoning, the KAON Server, which provides a generic infrastructure to facilitate engineering of ontology-based applications, and KAONtoEdit which enables OntoEdit (www.ontoknowledge.org/tools/ontoedit.shtml) to use the KAON API in order to load, modify and store KAON ontology models.

The survey on existing ontology management and evolution methodologies has revealed certain shortcomings. First of all, ontology integration suffers significantly and layering is rarely employed. In addition, there is a lack of an ontology model that will capture and enhance the temporal information contained in ontologies. Current approaches in ontology modelling do not offer sufficient temporal semantics for the evolution history of ontologies. Finally, key issues for ontology evolution, such as consistency preservation and change propagation, are often neglected. The Heraclitus II framework attempts to address these shortcomings in the context of IM and BI.

3. The Heraclitus II framework

Fig. 1 shows the ontology layering architecture adopted in Heraclitus II. The lower layers represent more generic and all-purpose ontologies, while the upper layers are customized for certain uses within an IM system. When traversing the layers from bottom to top, each layer reuses and extends the previous ones. In addition, whenever a layer extends the ones below it (e.g. with the insertion of new concepts), these extensions are propagated to the lower layers. Each layer is maintained by a different group of ontology authors, depending on the expertise that each layer requires. The integration of the ontology pyramid layers is achieved with the use of ontology mapping between ontologies belonging to the same layer (intra-layer), or different ones (inter-layer).

The *Lexical Ontology* layer contains domain-independent ontologies of a purely lexicographical nature. This layer handles lexicographical issues, such as multilingualism. An example of such an ontology is the widely adopted WordNet (Fellbaum, 1998). Modelling of a certain domain is the main characteristic of the *Domain Ontology* layer. The ontologies of the gene ontology (GO) project