Spring 3-2018

ONTOLOGICAL SEMANTICS AND ENTERPRISE CONTENT MANAGEMENT IN SYSTEMS DEVELOPMENT

Osama Bassam Rabie
Virginia Commonwealth University, rabieob@vcu.edu

Heinz Roland Weistroffer
Virginia Commonwealth University, hrweistr@vcu.edu

ONTOLOGICAL SEMANTICS AND ENTERPRISE CONTENT MANAGEMENT IN SYSTEMS DEVELOPMENT

Osama Bassam Rabie
Virginia Commonwealth University
rabieob@vcu.edu

H. Roland Weistroffer
Virginia Commonwealth University
hrweistr@vcu.edu

ABSTRACT

This paper proposes a dynamic method to develop information systems. Our premise is that using ontology supports automation and objectivity in processing the stakeholders' requirements, and that following the enterprise content management (ECM) lifecycle helps in making information systems development more systematic.

Keywords

Ontology, ECM, systems development

INTRODUCTION

Distributed information, big data, and mounting security challenges make information systems development increasingly complex. Capturing, organizing, processing, and maintaining systems requirements that satisfy the stakeholders are arduous. De Cesare et al. (de Cesare et al. 2007) stated that the adoption of semantic-based systems development increases adaptability and flexibility of software systems. Their premise is that information systems model the real world, and thus, information systems development should be based on ontological semantics, as semantics increase the precision of mapping and the integration of complex real-world phenomena with systems components.

This paper proposes a dynamic method to develop information systems. Our premise is that using ontology supports automation and objectivity in processing the stakeholders' requirements, and that following the enterprise content management (ECM) lifecycle helps in making information systems development more systematic.

The organizational systems development process is getting increasingly complex, due in part to scale of projects, conflicts in requirements, and heterogeneity of distributed data. Nowadays, systems development typically uses multiple sources to acquire information. "There is an increasing awareness of the value of data that can be mined to harvest information, and more and more organizations are trying to maximize the benefits of the data they own" (Rabie and Weistroffer 2015). Many organizations have their data distributed among multiple sources in a multitude of configurations. Ontology can be used to integrate data to overcome data syntax and semantic heterogeneity. Ontology has scalability to adjust for changes in projects scale, e.g., changes in the requirements, changes of data sources, or changes in organizational policy. It can support reasoning to resolve conflicts in requirements. While the use of ontologies in systems development is rarely discussed in the published literature, there is an increasing need for research on ontology use (Ashraf et al. 2015).

Berners-Lee stated "I have a dream for the Web [in which computers] become capable of analyzing all the data on the Web – the content, links, and transactions between people and computers. A 'Semantic Web,' which should make this possible, has yet to emerge, but when it does, machines talking to machines will handle the day-to-day mechanisms of trade, bureaucracy, and our daily lives. The 'intelligent agents' people have touted for ages will finally materialize" (Berners-Lee 1999). Ontologies can offer the mechanism for automated and systemized data integration to support systems development.

SEMANTIC-BASED INFORMATION SYSTEMS DEVELOPMENT BACKGROUND

Rabie and Weistroffer (2015) discussed four categories of literature explaining the use of ontologies in information systems development. Category one is using domain ontology to capture information systems requirements. The authors referenced two papers in this category, Farfeleder et al. 2011 (Farfeleder et al. 2011) and Kaiya and Saeki (Kaiya and Saeki 2006). Farfeleder et al. proposed a prototypic semantic guidance system that uses domain ontology to aid engineers in capturing requirements. Kaiya and Saeki proposed an ontology based requirements elicitation method, which suggests requirements addition or deletion to enhance consistency by using domain ontology with inference rules and quality metrics.

Category two is using ontology to detect conflicts between maintenance requirements and actual specifications. In this category, Rabie and Weistroffer referenced a paper by Liu and Yang (Liu and Yang 2012), experimenting with an ontology-based blog in which stakeholders can request system changes. Possibly resulting conflicts among these changes are detected via rule inferences.

Category three uses ontology to check the consistency of system's requirements. In this category, Rabie and Weistroffer referenced a paper by Beydoun et al. (Beydoun et al. 2014), which proposed creating ontologies based on the requirements list and rule inference. Rule inference is used to check for inconsistency, concept satisfaction, and subclasses relations.

Category four is using ontology for managing requirements. In this category, Rabie and Weistroffer referenced a paper by Kumar and Kumar (Kumar and Kumar 2011), which proposed an ontology-based knowledge management system. The proposed system uses four requirements management metrics to assess the development of information systems. The metrics include size and status of requirements, requirements specifications, perceived work efficiency, and changes to requirements.

Albani and Dietz (Albani and Dietz 2011) proposed a methodology to develop enterprise information systems that is based on the notions of enterprise ontology and business components. The methodology explicates information systems development within a conceptual framework called generic system development process. The proposed methodology focuses on the essential features of an enterprise, thereby reducing the complexity of domain models and finding stable business components suitable for business process re-design and reengineering. Their premise is that the implementation of the business domain changes more often than the actual business domain, thus, finding stable business components is more challenging than describing the business domain. Their proposed methodology can offer stable business components because it applies ontological models to find components.

We are not aware of any published paper that proposes following the enterprise content management (ECM) lifecycle and combining it with ontology. Our proposed ontological semantics and enterprise content management lifecycle to develop information systems seems to be the first such proposal, combining the benefits of the ECM lifecycle and ontology.

THE ONTOLOGICAL SEMANTIC ECM SYSTEMS DEVELOPMENT LIFECYCLE

The effective use of data can give organizations a substantial competitive advantage. Effective use of data may also support their systems development efforts. However, the majority of organizational data and information are aggregated from multiple sources in a multitude of configurations (Dhillon 2015). Moreover, many organizations use data available on the web as information sources (Fisher and Sheth 2004), which can result in challenges due to scale, heterogeneity, and distribution of content. These challenges can be handled using ontology-based ECM to support systems development (Fisher and Sheth 2004). Organizations use ECM systems to manage and generate their content assets (Alalwan et al. 2014).

This paper proposes using ontology-based concepts of ECM systems in organizational systems development. The framework is presented in Figure 1. The proposed framework uses concepts of each activity of the ECM lifecycle proposed by Smith and McKeen (Smith and McKeen 2003) to process organizational systems requirements. The ECM lifecycle, presented in figure 1A, includes four activities: capture, organize, process, and maintain. In capture, organizations gather content that helps operational activities and business intelligence. Organize is to unify the captured content to make it more navigable. Process is to develop and analyze the content to help in decision-making. Maintain is to ensure that the content is up-to-date and starts a new cycle of content when necessary.

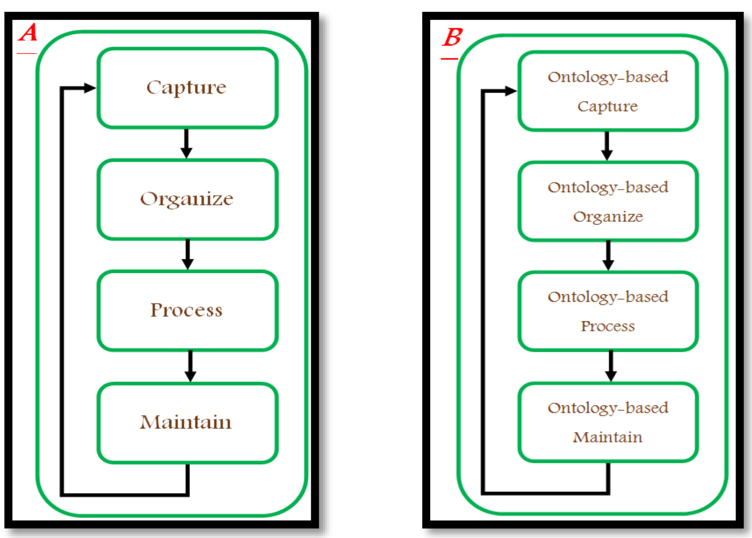

Figure 1A. Enterprise Content Management (ECM) Lifecycle
Figure 1B. Proposed Semantic ECM Lifecycle

The proposed framework, shown in Figure 1B, uses the concepts of the ECM lifecycle to support organizational systems development. Ontology-based capture makes use of an itemized requirements list. Ontology-based organize maps the requirements to atomic concepts of the domain ontology. Ontology-based process evaluates the requirements list by assessing the quality and structural requirements characteristics in the ontology. Ontology-based maintain updates and maintains the ontology of the requirements list. If the stakeholders provide new or modified requirements, a new cycle may start for the new requirements, i.e. move back from ontology-based maintain to ontology-based capture. The cycle will incrementally and iteratively capture the requirements, detect conflicts in change requests, check consistency, and manage the requirements. The iterations will continue until the quality and structural characteristics in the ontology are deemed satisfactory.

ONTOLOGY MEASURES FOR QUALITY AND STRUCTURAL CHARACTERISTICS

There are several ways to evaluate the quality and structure of the ontology. Table 1 explains the symbols used for various measures.

Symbol	**Definition**
Oc	ontology set of concepts
F	set of frame references
Fr	frame references set of relationships
Fi	frame references set of instances
Fc	frame references set of concepts
F*	set of references that ontology does not need to for a given implementation
||	absolute value
/	divided by
^	logical AND
∩	intersection of sets (i.e. the common elements of these sets)
p(c)	number of parents for a concept c
{}	indicates a set (e.g. {c belongs to Oc: P(c) > 1}is the set of concepts that have more than one parent)
< >	indicates any instance of a collection of sets (e.g. <Fc, Fi, Fr> refers to any instance that belongs to the frame references set of concepts, instances, or relationships)

Table 1. Definition of Symbols

1. Precision

Precision measures the amount of overlap between what was modeled in the ontology and the intended domain relative to the size of the ontology (Guarino 2004): $|Oc \cap Fc| / |Oc|$. A small value may indicate that the requirements list is incomplete and should be added to.

2. Recall

Recall measures coverage, defined as the amount of overlap between the ontology and the domain relative to the size of the domain (Guarino 2004): $|Oc \cap Fc| / |Fc|$. Similarly to precision, a small value may indicate that the requirements list is incomplete and should be added to.

3. Data Consistency

Data consistency is measured as the proportion of meta-consistent concepts in the ontology set of concepts: $|\text{meta-consistent}(Oc)| / |Oc|$. Data-consistency relates to the accuracy and integration levels between data. If the requirements list is found to be inconsistent, items causing the inconsistency should be removed, in order of least priority.

4. Tangledness

Tangledness measures the proportion of concepts with multiple parents: $|\{c: c \in Oc \wedge P(c) > 1\}| / |Oc|$

5. Incompleteness

Incomplete concept classification (Gómez-Pérez 2001): |{c: for all <Fc, Fi, Fr> belongs to F*, for some (c belongs to Fc) AND (c does not belong to Oc)}|. If the requirements list is incomplete, more items should be added by focusing on concepts like function, object, and environment and by tracing relationships like apply, perform, is-a, has-a, and requires.

EXAMPLE

This section shows an example on using the proposed ontology-based framework for organizational systems development. Domain ontology of library was created for demonstration.

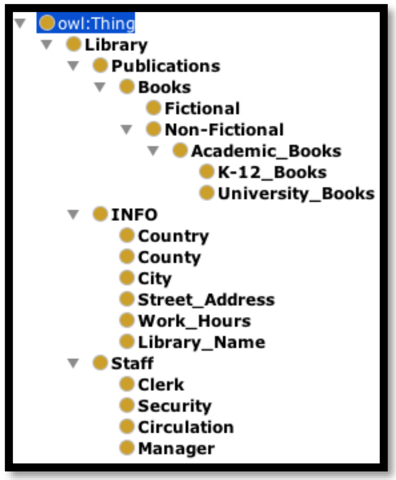

Figure 2. The Classes of Library Domain Ontology

Figure 2 shows the classes of library domain ontology that is used in this example. The class Library is the super class and has three subclasses Publications, INFO, and Staff.

First Cycle Ontology-Based Capture

The development cycle starts by eliciting the requirements from stakeholders. In this example, the requirements list consists of the following items:

- First requirement: Create Library System
- Second requirement: All books should be able to be checked out
    - Second Requirement A: Seventh grade math books should not be checked out for more than a week
    - Second Requirement B: Managers can extend the check-out period
- Third requirement: If the library opens until mid-night in King's County, Washington state, they should have the term wAy by the end of the library's name, e.g., University District Public Library wAy

First Cycle Ontology-Based Organize

The second step is to map each requirement item to its atomic concepts of the domain ontology. In this example, Library System will be mapped to the class library. The second requirement is mapped to the classes Circulation and Books. The rule of the seventh-grade math books is mapped to the classes Circulation and k-12_Books. The second requirement B is mapped to the classes Manager and Circulation. The third requirement is mapped to the concepts Country, County, Work_Hours, and Library_Name.

First Cycle Ontology-Based Process

The third step is to evaluate the requirements list by evaluating the quality and structural characteristics in ontology, thus, performing ontology-based process. For this example, several measures of the measures presented earlier are used to evaluate the ontology created in the second step.

First Cycle Ontology-Based Maintain

The fourth step needs the requirements engineer to take two actions. First, (s)he should declare that the requirements list has incomplete requirements. Second, (s)he should get back to the stakeholders for requirements elicitation. Knowing the concepts to ask the stakeholders about is the major benefit of using the proposed ontology-based framework for organizational systems development. Based on the ontology-based requirements evaluation, the requirements engineer will try to maintain the ontology of requirements list, thus, performing ontology-based maintain. The requirements engineer will not ask the stakeholders for requirements that discuss the concepts Publications and Non_Fictional since they should be dropped to improve the conciseness of the ontology. Since there are concepts that need to be discussed, the requirements engineer should start a new cycle.

The next cycle starts after substituting the concepts Publications with Books and Non_Fictional with Acdemic_Books, then remove the concepts Publications and Non_Fictional from the ontology. In this case, it does not affect the mapping of the concepts since the concepts Non-Fictional and Publications were not mapped at the second step of the first cycle.

Second Cycle Ontology-Based Capture

In this cycle, new and modified requirements are elicited. The stakeholders gave the following requirements in addition of the requirements of the first cycle:

- Fourth requirement: clerks work as circulation staff
- Fifth requirement: the library system should have a list of security guards for each library on Walker Avenue, Baltimore, Maryland
- Sixth requirement: college level books should be listed in a separate list
- Seventh requirement: fictional books should have lists for young adults and for kids

Second Cycle Ontology-Based Organize

The fourth requirement places Circulation under Clerk since clerks work as circulation staff. The fifth requirement maps the concepts to Security, City, and Street_Address. It marks discussing all the concepts under INFO. The sixth requirement is mapped to University_Books. It marks discussing all the concepts under Academic_Books. The seventh requirement is mapped to Fictional and needs the addition of two concepts under it: Kids and Young_Adults.

Second Cycle Ontology-Based Process

The third step is to evaluate the requirements list by evaluating the quality and structural characteristics of the ontology, thus, performing ontology-based process. For this example, several measures of the measures presented in section 3.1 are used to evaluate the ontology created in the second step.

Second Cycle Ontology-Based Maintain

The fourth step needs the requirements engineer to take five actions. First, (s)he should substitute the concept Circulation with Clerk. Second, (s)he should map the concepts previously mapped to Circulation to Clerk. Third, (s)he should remove the concept Circulation. Fourth, mark the concept Staff as discussed. Fifth, she should add the concepts Kids and Young_Adults under Fictional. Based on the ontology-based requirements evaluation, the requirements engineer will try to maintain the ontology of requirements list, thus, performing ontology-based maintain. Since the structure of the ontology is altered, the requirements engineer should start a new cycle to assess the quality and structure of the ontology. Figure 3 shows the list of classes after the second cycle.

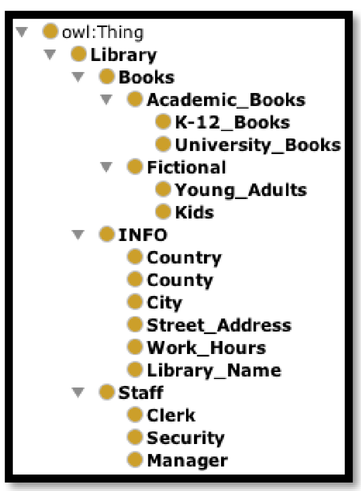

Figure 3. List of Classes after the third Cycle

Third Cycle Ontology-Based Capture

- Eighth requirement: assess the quality and structural characteristics of the ontology

Third Cycle Ontology-Based Organize

In this case, no new concepts are introduced so the requirement engineer can start evaluating the ontology.

Third Phase Ontology-Based Process

The third step is to evaluate the requirements list by evaluating the quality and structural characteristics in ontology, thus, performing ontology-based process. For this example, several measures of the measures presented in section 3.1 are used to evaluate the ontology created in the second step.

Third Phase Ontology-Based Maintain

No specific action is required. The requirements engineer can end the requirements elicitation and assessment.

CONCLUDING REMARKS

This paper presents an ontology-based framework for organizational systems development, using the enterprise content management (ECM) system lifecycle as guideline. It supports four areas where ontology can support systems development (Rabie & Weistroffer, 2015), viz. capturing requirements, detecting conflicts in change requests, consistency checking, and requirements management. The presented example shows the precision of requirements assessment. It also shows objectivity in assessment that can lead to fast improvement of the requirement elicitation process.

A major limitation of the proposed framework is that it does not handle non-functional requirements. Another limitation is the large number of measures the requirements engineer should process to evaluate the quality and structural characteristics in ontology.

REFERENCES

1. Alalwan, J.A., Thomas, M.A., and Weistroffer, H.R. 2014. "Decision Support Capabilities of Enterprise Content Management Systems: An Empirical Investigation," *Decision Support Systems* (68), pp. 39-48.
2. Albani, A., and Dietz, J.L. 2011. "Enterprise Ontology Based Development of Information Systems," *International Journal of Internet and Enterprise Management* (7:1), pp. 41-63.
3. Ashraf, J., Chang, E., Hussain, O.K., and Hussain, F.K. 2015. "Ontology Usage Analysis in the Ontology Lifecycle: A State-of-the-Art Review," *Knowledge-Based Systems* (80), pp. 34-47.
4. Berners-Lee, T. 1999. *Weaving the Web*, (First Edition ed.). HarperBusiness.
5. Beydoun, G., Low, G., García-Sánchez, F., Valencia-García, R., and Martínez-Béjar, R. 2014. "Identification of Ontologies to Support Information Systems Development," *Information Systems* (46), pp. 45-60.
6. de Cesare, S., Holland, G., Holtmann, C., and Lycett, M. 2007. "Semantic-Based Systems Development," *OOPSLA Companion*, p. 760.
7. Dhillon, G. 2015. *Enterprise Cyber Security*, (2nd edition ed.). CreateSpace Independent Publishing Platform.
8. Farfeleder, S., Moser, T., Krall, A., Stålhane, T., Omoronyia, I., and Zojer, H. 2011. "Ontology-Driven Guidance for Requirements Elicitation," in *The Semanic Web: Research and Applications*. Springer, pp. 212-226.
9. Fisher, M., and Sheth, A. 2004. "Semantic Enterprise Content Management," in *Practical Handbook of Internet Computing*. pp. 1-25.
10. Guarino, N. 2004. "Toward a Formal Evaluation of Ontology Quality," *IEEE intelligent Systems* (19:4), pp. 78-79.
11. Kaiya, H., and Saeki, M. 2006. "Using Domain Ontology as Domain Knowledge for Requirements Elicitation," *Requirements Engineering, 14th IEEE International Conference*: IEEE, pp. 189-198.
12. Kumar, S.A., and Kumar, T.A. 2011. "Study the Impact of Requirements Management Characteristics in Global Software Development Project: An Ontology Based Approach," *International Journal of Software Engineering and Application* (2:4), pp. 107-125.
13. Liu, C.-L., and Yang, H.-L. 2012. "Applying Ontology-Based Blog to Detect Information System Post-Development Change Requests Conflicts," *Information Systems Frontiers* (14:5), pp. 1019-1032.
14. Rabie, O.B.J., and Weistroffer, H.R. 2015. "Use of Ontologies in Information Systems Development," *Workshop on E-Business*: Springer, pp. 202-206.
15. Smith, H.A., and McKeen, J.D. 2003. "Developments in Practice Viii: Enterprise Content Management," *Communications of the Association for Information Systems* (11:1), p. 41.