

Continuous Evaluation in the process of Ontology Development

Dejan Lavbič, Marjan Kriper, and Marko Bajec

Laboratory for Data Technologies

ULJ, Faculty of Computer & Information Science

Ljubljana, Slovenia

{dejan.lavbic, marjan.krisper, marko.bajec}@fri.uni-lj.si

Abstract—Due to complexity of existing methodologies for ontology development we propose facilitating ontology development with continuous evaluation of steps in the process of ontology development. The approach is called Rapid Ontology Development (ROD) and is based on completeness indicator that helps guiding developer by constant evaluation of ontology and producing recommendations to progress to next step and improve the quality of ontology. The applicability of the approach is demonstrated on Financial Instruments and Trading Strategies (FITS) ontology. The main contribution of the paper is the suggested approach for rapid development of ontologies which brings ontology modeling closer to business users as it does not require from users to know any formal syntax.

Keywords- *Rapid ontology development, Business oriented approach, Ontology evaluation.*

I. INTRODUCTION

The simplicity of using approaches for ontology construction and accompanying tool support is an important issue which needs a lot of attention and further work. Current approaches in ontology development are technically very demanding and require long learning curve and are therefore inappropriate for business users. In majority of existing approaches an additional role of knowledge engineer is required for mediation between actual knowledge that business users possess and ontology engineers who encode knowledge in one of selected formalisms. The use of business rules management approach seems like an appropriate way to simplification of development and use of ontologies in business applications. Besides simplifying the process of ontology creation we also have to focus on very important aspect of ontology completeness. The problem of error free ontologies has been discussed, e.g., in [15] and [7]. One of the goals of Rapid Ontology Development (ROD) approach that authors propose is constant evaluation of ontologies during the development process for major types of errors. User can therefore, based on recommendations, improve the ontology and eliminate the error. It is also a very important aspect that before the usage the ontology itself is error free. Thus we define ROD model that introduces detail steps in ontology manipulation. The starting point was to improve existing approaches in a way of simplifying the process and give user tool support throughout the lifecycle and not to conclude with developed ontology but enable the use of ontology in various scenarios.

The paper is structured as follows: after describing related works in Section II, we continue with the description of the Rapid Ontology Development model (Section 3). The evaluation of the model is provided in Section IV. Finally, in Section V, we give concluding remarks and ideas for future work.

II. RELATED WORKS

Ontology is a vocabulary that is used for describing and presentation of a domain and also the meaning of that vocabulary. The definition of ontology can be highlighted from several aspects, e.g., from taxonomy ([3], [17], [22]) as knowledge with minimal hierarchical structure, vocabulary ([1], [11]) with words and synonyms, topic maps ([6], [14]) with the support of traversing through large amount of data, conceptual model ([10], [12]) that emphasizes more complex knowledge and logic theory ([3], [23]) with very complex and consistent knowledge.

Ontologies are used for various purposes ([2], [4], [5], [9], [16]) such as natural language processing, knowledge management, information extraction, intelligent search engines, digital libraries, business process modeling, etc. While the use of ontologies was primarily in the domain of academia, situation now improves with the advent of several methodologies for ontology manipulation. Existing methodologies for ontology development in general try to define the activities for ontology management, activities for ontology development and support activities. More detailed insight into wide spectrum of methodologies can be found, e.g., in [3], [5], and [19], whilst here only the most representative are depicted. CommonKADS [18] is focused towards knowledge management in information systems with analysis, design and implementation of knowledge. Enterprise Ontology [21] is the groundwork for many other approaches and is also used in several ontology editors. METHONTOLOGY [8] enables the construction of ontologies at the knowledge level and the approach is very close to prototyping. Another approach is TOVE [20] where authors suggest using questionnaires, which is useful where domain experts have very little knowledge of knowledge modeling. OTK Methodology [19] defines steps in ontology development into detail and introduces two processes – Knowledge Meta Process and Knowledge Process. UPON [13] is based on Unified Software Development Process and is supported by UML language. DILIGENT [5] is focused on different approaches to distributed ontology development.

In aforementioned methodologies there is a lack of Rapid Application Development (RAD) approaches in ontology development, the use of ontologies in business applications and approaches analogous agile methodologies in software engineering. There is also an evident lack of approaches that do not require extensive technical knowledge of formal languages and techniques for capturing knowledge from domain experts. The majority of approaches require an additional role of knowledge engineer that transfers the knowledge into formal syntax within knowledge base.

This paper introduces a novel approach in ontology modeling based on good practices and existing approaches while trying to eliminate the need of knowing formal syntax required for codifying the ontology and therefore bringing ontology modeling closer to business users who are actual knowledge holders. The following section will introduce the process, required tasks and highlight the advantages of Rapid Ontology Development (ROD) approach.

III. RAPID ONTOLOGY DEVELOPMENT MODEL

A. ROD process

Ontology development following ROD approach is through 3 stages pre-development, development and post-development as depicted in Figure 1. Every stage delivers a specific output with the common goal of creating functional component based on ontology that can be used in several systems and scenarios. In pre-development stage the output is feasibility study that is used in subsequent stage development to construct essential model definition. The latter artifact represents the schema of problem domain that has to be coupled with instances from the real world. This is conducted in the last stage post-development which produces functional component for usage in various systems.

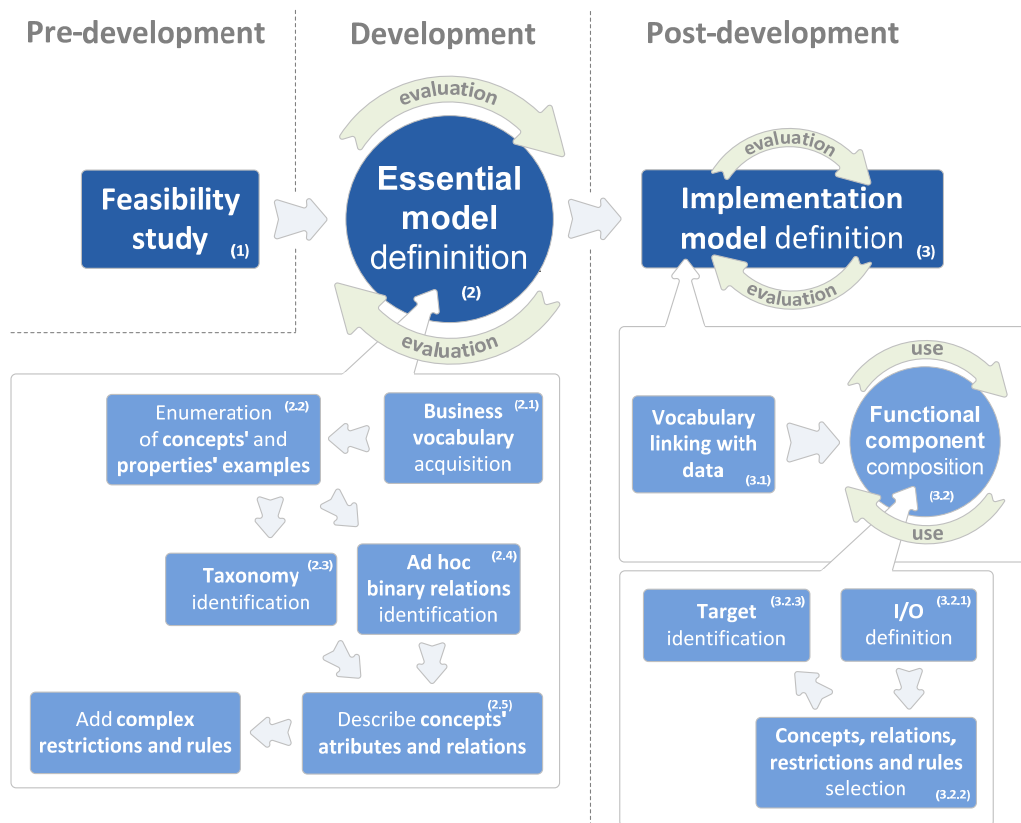


Figure 1: Process of Rapid Ontology Development

The first stage called pre-development is concerned with feasibility study (step 1) of problem domain. This step includes assessing the scope of the project with clear definition of boundaries. Next stage is development with the main goal of producing essential model definition (step 2). The development stage contains several steps: business vocabulary acquisition (step 2.1), enumeration of concepts' and properties' examples (step 2.2), taxonomy identification

(step 2.3), ad hoc binary relations identification (step 2.4), describe concepts' attributes and relations (step 2.5) and add complex restriction and rules (step 2.6). Very important aspect of this stage is constant evaluation of developed ontology using ontology completeness assessment indicator OC which is presented in Section 3.2. The last stage is post-development where implementation model definition (step 3) is constructed. The post-development stage contains 2 steps:

vocabulary linking with data (step 3.1) and functional component composition (step 3.2). The latter step is furthermore decomposed into I/O definition (step 3.2.1), concepts, relations, restrictions and rules selection (step 3.2.2) and target identification (step 3.2.3). From evaluation point of view this stage is similar to development stage, because is also constantly evaluated using ontology completeness assessment indicator OC (see Section III.B).

B. Ontology Completeness

To aid users and to simplify progressing through steps in process of Rapid Ontology Development, ontology completeness indicator is introduced. There are 2 main purposes of OC indicator:

(1.) It can be used independently of ROD process (with other ontology development methodologies or ad hoc). Based on semantic review of ontology, enhancements for ontology improvement are available to the user in a form of multiple actions of improvement sorted by their impact. Besides actions and their impacts, detail explanation of action is also available.

(2.) As a helper tool and facilitator in progressing through steps of ROD process. While the user is in a certain step of the process, the OC measurement is adapted to that step by redefinition of weights for calculation. When OC measurement reaches a threshold (e.g., 80%) user can progress to the following step. The adapted OC value for every phase is calculated on-the-fly and whenever a threshold value is crossed, a recommendation for progressing to next step is generated. This way user is aided in

progressing through steps of ROD process from business vocabulary acquisition to functional component composition. In case that ontology already exists, with OC measure we can place the completeness of ontology in ROD process and start improving ontology in suggested phase of development.

Ontology completeness (OC) is defined as $OC = f(C, P, R, I) \in [0, 1]$, where C is set of concepts, P set of properties, R set of rules and I set of instances. OC can further be defined as $OC = \sum_i^n w_i \cdot leafCondition_i$ where n is the number of leaf conditions (see Figure 2) and leafCondition is a leaf condition, where semantic check is executed. For relative weights and leaf condition calculation the following restrictions apply $\sum_i w_i = 1, \forall w_i \in [0, 1]$ and $\forall leafCondition_i \in [0, 1]$. Relative weight w_i denotes global importance of leafCondition_i and is dependent on all weights from leaf to root concept.

The tree of conditions in OC calculation is depicted in Figure 2 and contains semantic checks that are executed against the ontology. The top level is divided into TBox, RBox and ABox components. Subsequent levels are then furthermore divided based on ontology error classification [7]. Aforementioned sublevels are description, partition, redundancy, consistency and anomaly. This proposed structure can be easily adapted and altered for custom use. Leafs in the tree of OC calculation conditions are implemented as semantic checks while all preceding elements are aggregation with appropriate weights as depicted in Figure 2.

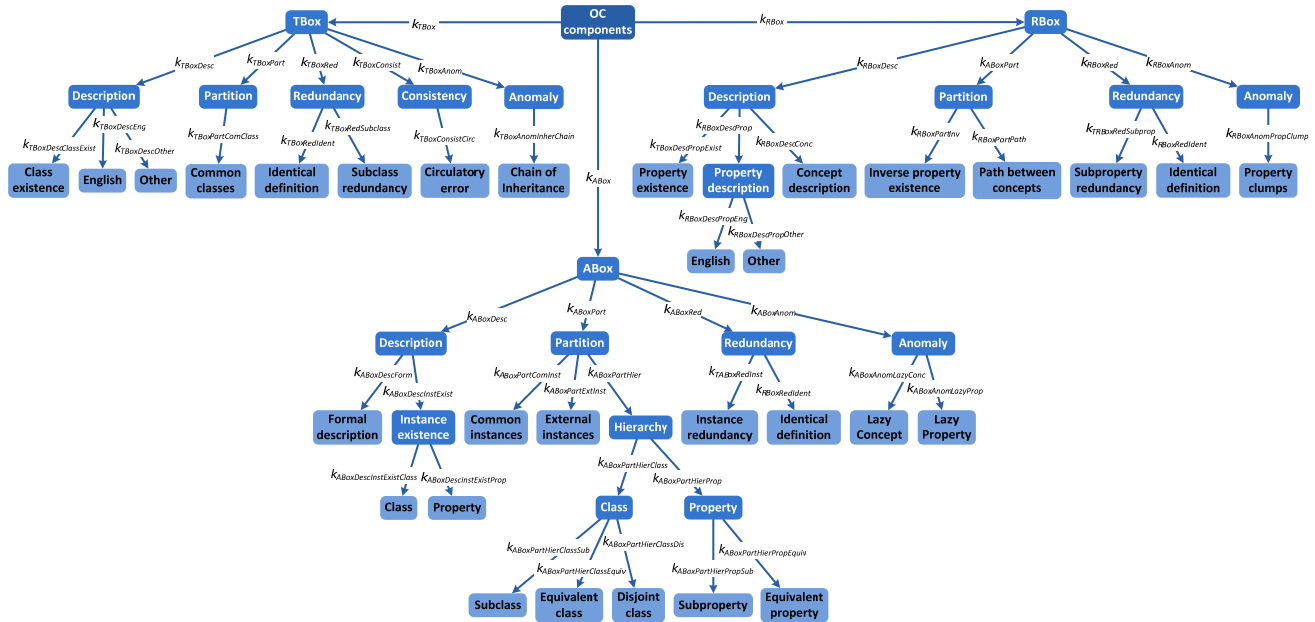


Figure 2: Ontology completeness (OC) tree of conditions, semantic checks and corresponding weights

Certain phases put emphasis on selected components. In initial stages user has to deal with description and structure of concepts, while at the end of essential model definition

restrictions and rules become more important. As post-development and implementation model definition is concerned users have to think about instances of schematic

part of ontologies, therefore attention is shifted to ABox component. There are two types of outputs from ontology completeness (OC) calculation (see Figure 3): (1) OC price and (2) Recommendations to improve OC price.

As depicted in Figure 3 OC price is presented as a value expressed in percentage (e.g., 68%) and visualized as a progress bar. Besides this basic view it turns out that radar chart view of OC price is also very informative as it highlights which areas need improvement. In radar chart view top levels for visualization become description, partition, redundancy, consistency and anomaly as they are more suitable for business users than TBox, RBox and ABox components. All the recommendations are listed in a table view and sorted by their impact. When a recommendation is selected the impact is also depicted in radar chart for better understanding of how the change will affect ontology.

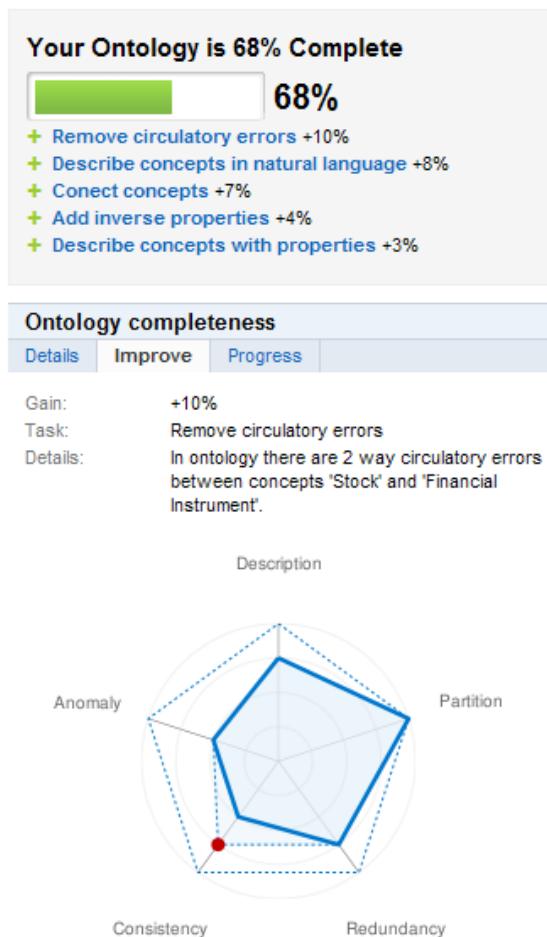


Figure 3: User interface and display of ontology completeness results and improvement recommendations

IV. EVALUATION

Rapid Ontology Development (ROD) process was verified on a case study from financial domain. The requirement was to develop an ontology that contains basic

information about financial instruments, trading and custom trading strategies using ROD. The solution enabled the user to test multiple trading strategies on real trading data that was available for selected stocks in pre-prepared CSV files and Yahoo! Finance resource for real-time and historic data was also available. The developed solution was exported as a functional component (standalone J2SE application) with input parameter of financial instrument symbol and as output trading days that have either buy or sell signals and trade reason. Business user defined the ontology containing required knowledge about financial instruments and at least 3 trading strategies that can easily be combined and tested individually or together. During the process of ontology building user actions were continuously monitored. The grain of collected data was one iteration step where ontology completeness prices and number of ontology elements (concepts, properties and axiom, including rules) were recorded.

The ontology produced is depicted in Figure 4 and is based on 2 simple facts about trading with financial instruments: (1) financial instrument is traded at a stock exchange market and (2) financial instrument is traded on a trading day. Specialization of financial instrument was introduced with Stock, ETF (Exchange Traded Fund) and K.O. certificate. The same approach was used for representing distinction between trading days with buy and sell signals. All concepts were according to OC rules described with formal properties and also natural language.

Trading strategy was implemented in separate ontology that used dynamic import of aforementioned ontology. This was utilized due to modular approach and the ability to develop and use several strategies separately or combined. Trading strategy mainly contains restrictions and rules, while in more complex definition (Japanese candlestick strategy) additional concepts were also introduced. During the experiment 3 different trading strategies were defined: simple trading strategy, simple moving average strategy and Japanese candlestick strategy. Input trading data consisted of quotes data from Apple Inc. (AAPL), Google Inc. (GOOG) and UltraShort S&P500 ProShares (SDS) in a 1 month period. Instances were imported at runtime from World Wide Web (Yahoo! Finance) and File system text files (CSV).

The process of ontology creation and exporting it as functional component is depicted in Figure 5. Chart represents ontology completeness price and number of ontology elements regarding to iterations in the process. During the process of ontology construction based on ROD approach the user was continuously supported by ontology evaluation and recommendations for progressing to next steps. When user entered a phase and started performing tasks associated with the phase (detail description is given in Section III.A), ontology completeness was evaluated as depicted in Figure 1 and further presented in Section III.B. While OC was less than 100% user followed instructions for improving ontology as depicted in Figure 3. Results of OC evaluation are available in a simple view, where basic statistics about ontology is displayed (number of concepts, properties, rules, individuals, etc.), progress bar depicting completeness, and details about evaluation, improvement

recommendations and history of changes. The core element is progress bar that denotes how complete ontology is and is accompanied with a percentage value. Following are recommendation for ontology improvement and their gains (e.g., remove circulatory errors (+10%), describe concepts in natural language (+8%), connect concepts (-+7%), etc.).

When improvement is selected (e.g., remove circulatory errors) the details are displayed (gain, task and details). As depicted in Figure 3 circulatory error can be eliminated with removing the 2 way connection between concepts ‘Stock’ and ‘Financial instrument’ and by doing that gaining 10% in ontology completeness.

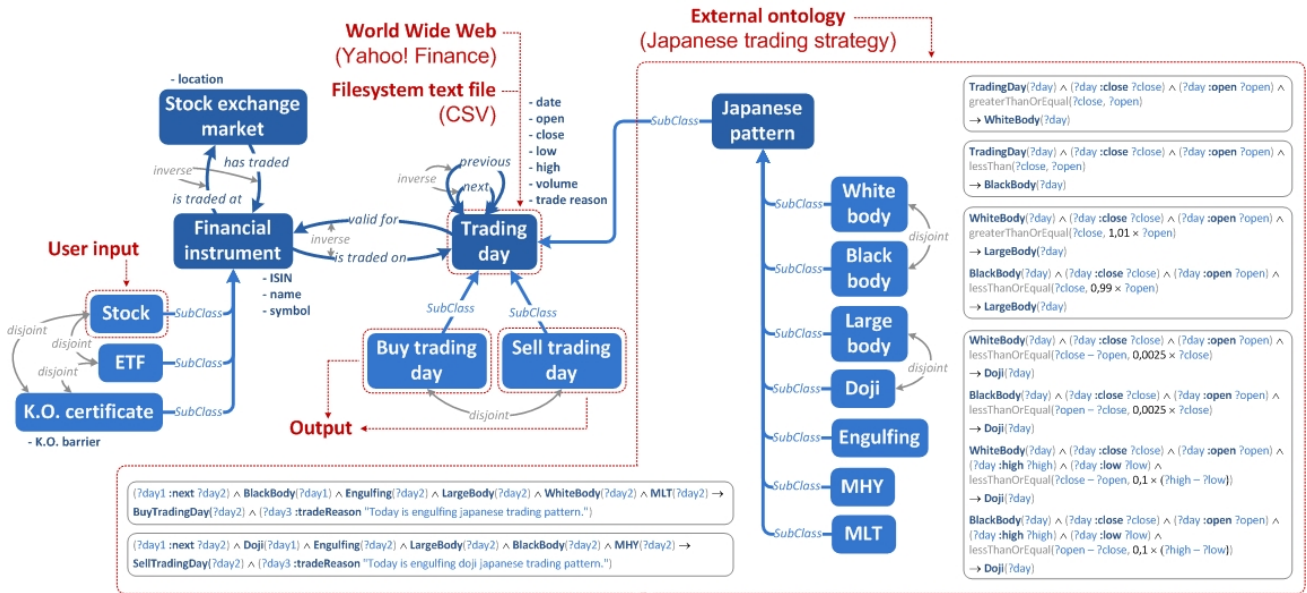


Figure 4: Trading ontology example

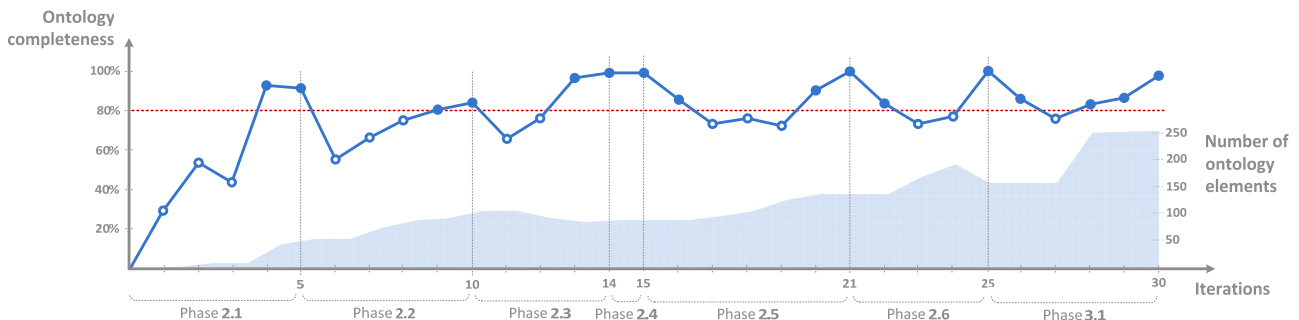


Figure 5: Ontology completeness assessment and number of ontology elements through phases of ROD process

The improvement and planned actions are also clearly graphically depicted on radar chart (see Figure 3). The shaded area with strong border lines presents current situation, while red dot shows TO-BE situation if we follow selected improvement. When OC price crosses a threshold value (in this experiment 80%, as depicted in Figure 5) a recommendation to progress to a new phase is generated. We can see from our example that for instance recommendation to progress from phase 2.5 to phase 2.6 was generated in 20th iteration with OC value of 91,3%, while in 19th iteration OC value was 76,5%. As Figure 5 depicts ontology completeness price and number of ontology elements are displayed. While progressing through steps and phases it's

seen that number of ontology elements constantly grows. On the other hand OC price fluctuates – it's increasing till we reach the threshold to progress to next phase and decreases when entering new phase. Based on recommendations from the system, user improves the ontology and OC price increases again.

V. CONCLUSION AND FUTURE WORKS

Available methodologies and approaches usually require very knowledgeable users and developers, while authors propose ROD approach that is more suitable for less technically knowledgeable users. With ROD approach and accompanying IntelliOnto tool business users get an

instrument for ontology modeling that doesn't require very extensive knowledge of ontology languages but still follow and utilize the possibilities of Semantic Web vision. It has been demonstrated on a case study from financial trading domain that a user can build Semantic Web application for financial trading based on ontologies that consumes data from various sources and enable interoperability. The solution can easily be packed into a functional component and used in various systems. By following ROD approach for building Semantic Web applications against existing approaches following advantages can be highlighted: (1) the required technical knowledge for ontology modeling is decreased, (2) the process of ontology modeling doesn't end with the last successful iteration, but continues with post-development activities of using ontology as a functional component in several scenarios and (3) continuous evaluation of developing ontology and recommendations for improvement.

REFERENCES

- [1] Bechhofer, S. & GOBLE, C., 2001, Thesaurus construction through knowledge representation. *Data & Knowledge Engineering*, 37, pp. 25-45.
- [2] Brambilla, M., Celino, I., Ceri, S. and Cerizza, D., 2006, A Software Engineering Approach to Design and Development of Semantic Web Service Applications. In 5th International Semantic Web Conference, Athens, USA.
- [3] Corcho, O., Fernandez-Lopez, M. and Gomez-Perez, A., 2003, Methodologies, tools and languages for building ontologies. Where is their meeting point? *Data & Knowledge Engineering*, 46, pp. 41-64.
- [4] Dahlgen, K., 1995, A linguistic ontology. *International Journal of Human-Computer Studies*, 43, pp. 809-818.
- [5] Davies, J., Studer, R. and Warren, P., 2006, *Semantic Web Technologies - trends and research in ontology-based systems* (Chichester, England: John Wiley & Sons).
- [6] Dong, Y. and Li, M. S., 2004, HyO-XTM: a set of hyper-graph operations on XML Topic Map toward knowledge management. *Future Generation Computer Systems*, 20, pp. 81-100.
- [7] Fahad, M. and Quadir, M. A., 2008, Ontological errors - Inconsistency, Incompleteness and Redundancy. In International Conference on Enterprise Information Systems (ICEIS) 2008, Barcelona, Spain.
- [8] Fernandez-Lopez, M., Gomez-Perez, A., Sierra, J. P. and Sierra, A. P., 1999, Building a chemical ontology using Methontology and the Ontology Design Environment. *Intelligent Systems*, 14.
- [9] Heflin, J. and Hendler, J., 2000, Searching the web with SHOE Artificial Intelligence for Web Search, pp. 36-40 (Menlo Park, USA, AAAI Press).
- [10] Jovanović, J. & Gašević, D., 2005, Achieving knowledge interoperability: An XML/XSLT approach. *Expert Systems with Applications*, 29, pp. 535-553.
- [11] Miller, G. A., 1995, WordNet: a lexical database for English. *Communications of the ACM*, 38, pp. 39-41.
- [12] Mylopoulos, J., 1998, Information modeling in the time of the revolution. *Information Systems*, 23, pp. 127-155.
- [13] Nicola, A. D., Navigli, R. & Missikoff, M., 2005, Building an eProcurement ontology with UPON methodology. In 15th e-Challenges Conference, Ljubljana, Slovenia.
- [14] Park, J. and Hunting, S., 2002, XML Topic Maps: Creating and Using Topic Maps for the Web (Boston, USA: Addison-Wesley).
- [15] Porzel, R. and Malaka, R., 2004, A Task-based Approach for Ontology Evaluation. In ECAI 2004, Workshop on Ontology Learning and Population, Valencia, Spain.
- [16] Rao, J., Dimitrov, D., Hofmann, P. nad Sadeh, N., 2006, A Mixed Initiative Semantic Web Framework for Process Composition. In 5th International Semantic Web Conference, Athens, USA.
- [17] Sanjuan, E. and Ibekwe-Sanjuan, F., 2006, Text mining without document context. *Information Processing & Management*, 42, pp. 1532-1552.
- [18] Schreiber, G., Akkermans, H., Anjewierden, A. et al., 1999, *Knowledge Engineering and Management - The CommonKADS Methodology* (London, England: The MIT Press: Cambridge, Massachusetts).
- [19] Sure, Y., 2003, *Methodology, Tools & Case Studies for Ontology based Knowledge Management* Institute AIFB, pp. 332 (University of Karlsruhe).
- [20] Uschold, M. & Grueninger, M., 1996, Ontologies: principles, methods and applications. *Knowledge Sharing and Review*, 11.
- [21] Uschold, M. & King, M., 1995, Towards a methodology for building ontologies. In Workshop on Basic Ontological Issues in Knowledge Sharing (IJCAI '95), Montreal, Canada.
- [22] Veale, T., 2006, An analogy-oriented type hierarchy for linguistic creativity. *Knowledge-Based Systems*, 19, pp. 471-479.
- [23] Waterson, A. and Preece, A., 1999, Verifying ontological commitment in knowledge-based systems. *Knowledge-Based Systems*, 12, pp. 45-54.