International Journal on New Computer Architectures and Their Applications (IJNCAA) 2(1): 167-182
The Society of Digital Information and Wireless Communications, 2012 (ISSN: 2220-9085)

Employing Semantic Web technologies in financial instruments trading

Dejan Lavbič[1] and Marko Bajec[1]

[1] University of Ljubljana, Faculty of Computer and Information Science,
Tržaška cesta 25, 1000 Ljubljana, Slovenia
{Dejan.Lavbic, Marko.Bajec}@fri.uni-lj.si

ABSTRACT

The lack of rapid approaches in ontology development led us to define Rapid Ontology Development (ROD) approach. The following approach includes the development process and constant evaluation of the steps in the process. The ROD approach was then applied to development of Financial Instruments and Trading Strategies (FITS) ontology. We pointed out that ontology development process does not conclude with successful definition of schematic part of ontology, but that developers should continue with post development activities where additional axiomatic information and instances with dynamic imports from various sources are defined. The result is executable ontology as part of Semantic Web application that can employ data from several data sources. The overall process of ontology development presented in this paper is suitable for users without extensive technical and programming skills.

KEYWORDS

Ontology, semantic web, financial instruments, trading strategies, rapid ontology development.

1 INTRODUCTION

The adoption of Semantic Web technologies is less than expected and is mainly limited to academic environment. We are still waiting for wide adoption in industry. We could seek reasons for this in technologies itself and also in the process of development, because existence of verified approaches is a good indicator of maturity. There are various technologies available that consider different aspects of Semantic Web, from languages for capturing the knowledge, persisting data, inferring new knowledge to querying for knowledge etc. Regarding the development process, there is also a great variety of methodologies for ontology development, as it will be further discussed in section 2, but simplicity of using approaches for ontology construction is another issue. Current approaches in ontology development are technically very demanding and require long learning curve and are therefore inappropriate for developers with little technical skills and knowledge. Besides simplification of the development process ontology completeness is also a very important aspect. In building ontology, majority of approaches focus on defining common understanding of a problem domain as a schematic model of the problem and conclude the development after few successful iterations. Post development activities that deal with defining instance data and employing developed ontology in Semantic Web application are usually omitted.

In this paper we apply Rapid Ontology Development (ROD) approach to construct Financial Instruments and Trading Strategies (FITS) ontology. The goal was to develop ontology by constructing schematic part of ontology including axiomatic information to fully support trading by employing reasoning. Furthermore this TBox part of ontology was combined to instance data (ABox) to construct knowledge base and therefore build mash up Semantic Web application to support financial instruments trading by applying various trading strategies. Target users of this approach are ones without extensive technical knowledge of data acquisition and ontology modeling but experts in financial trading. The main guideline in constructing ontology was to develop it to the level that enables direct employment in an application, which differs from majority of existing approaches where ontologies are mainly developed only to formally define the conceptualization of the problem domain.

The remainder of this paper is structured as follows. First some related work is presented in section 2 with emphasis on ontology development methodologies and applications of financial ontologies. The following section 3 introduces our approach for facilitating Semantic Web applications construction. The details of case study from the domain of financial instruments and trading strategies is further presented in section 4. First FITS ontology is presented, followed by semantic integration of data sources and then technological details about the prototype are depicted. Finally in section 5 conclusions with future work are given.

2 RELATED WORK

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. From taxonomy [1-3] as knowledge with minimal hierarchical structure, vocabulary [4, 5] with words and synonyms, topic maps [6, 7] with the support of traversing through large amount of data, conceptual model [8, 9] that emphasizes more complex knowledge and logic theory [1, 10, 11] with very complex and consistent knowledge.

Ontologies are used for various purposes such as natural language processing [12], knowledge management [13], information extraction [14], intelligent search engines [15], digital libraries [16], business process modeling [17-19] 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. Several methodologies exist for ontology manipulation and will be briefly presented in the following section. CommonKADS [20] is in fact not a methodology for ontology development, but is focused towards knowledge management in information systems with analysis, design and implementation of knowledge. CommonKADS puts an emphasis to early stages of software development for knowledge management. Enterprise Ontology [21] recommends three simple steps: definition of intention; capturing concepts, mutual relation and expressions based on concepts and relations; persisting ontology in one of the languages. This methodology is the groundwork for many other approaches and is also used in several ontology editors. METHONTOLOGY [22] is a methodology for ontology creation from scratch or by reusing existing ontologies. The framework enables building ontology at conceptual level and this approach is very close to prototyping. Another approach is TOVE [23] where authors suggest using questionnaires that describe questions to which ontology should give answers. That can be very useful in environments where domain experts have very little expertise of knowledge modeling. Moreover authors of HCONE [24] present decentralized approach to ontology development by introducing regions where ontology is saved during its lifecycle. OTK Methodology [25] defines steps in ontology development into detail and introduces two processes – Knowledge Meta Process and Knowledge Process. The steps are also supported by a tool. UPON [26] is an interesting methodology that is based on Unified Software Development Process and is supported by UML language, but it has not been yet fully tested. The latest proposal is DILIGENT [13] and is focused on different approaches to distributed ontology development.

In the domain of finance several ontologies and implementations of Semantic Web based application exits. Finance ontology [27] follows ISO standards and covers several aspects (classification of financial instruments, currencies, markets, parties involved in financial transactions, countries etc.). Suggested Upper Merged Ontology (SUMO) [28] also includes a subset related to finance domain, which is richly axiomatized, not just taxonomic information but with terms formally defined. There are also several contributions in financial investments and trading systems [29-31]. Several authors deal with construction of expert and financial information systems [32-35].

3 FACILITATING SEMANTIC WEB APPLICATIONS CONSTRUCTION

3.1 Problem and proposal for solution

This paper describes semantic mash up application construction based on ontologies. The process is supported by continuous evaluation of ontology where developer is guided throughout the development process and constantly aided by recommendations to progress to next step and improve the quality of the final result. Our main objective is to combine dynamic (Web) data sources with a minimal effort required from the user. The results of this process are data sources that are later used together with ontology and rules to create a new application. This final result includes ontology that not only represents the common understanding of a problem domain but is also executable and directly used in the semantic mash up application.

Existing approaches for ontology development and semantic mash up application construction are complex and they require technical knowledge that business users and developers don’t possess. As mentioned in section 2 vast majority of ontology development methodologies define a complex process that demands a long learning curve. The required technical knowledge is very high therefore making ontology development very difficult for non-technically oriented developers. Also majority of reviewed methodologies include a very limited evaluation support of developed ontologies and if this support exists it is limited to latter stages of development and not included throughout the process as is the case with our approach. Another problem that also exists is that the development process of ontology is completed after the first cycle and not much attention is given to applicability of ontology in an application.

3.2 Rapid Ontology Development

The process for ontology development ROD [36] that we follow in our approach is based on existing approaches and methodologies but is enhanced with continuous ontology evaluation throughout the complete process.

Developers start with capturing concepts, mutual relations and expressions based on concepts and relations. This task can include reusing elements from various resources or defining them from scratch. When the model is defined, schematic part of ontology has to be binded to existing instances of that vocabulary. This includes data from relational databases, text files, other ontologies etc. The last step in bringing ontology into use is creating functional component for employment in other systems.

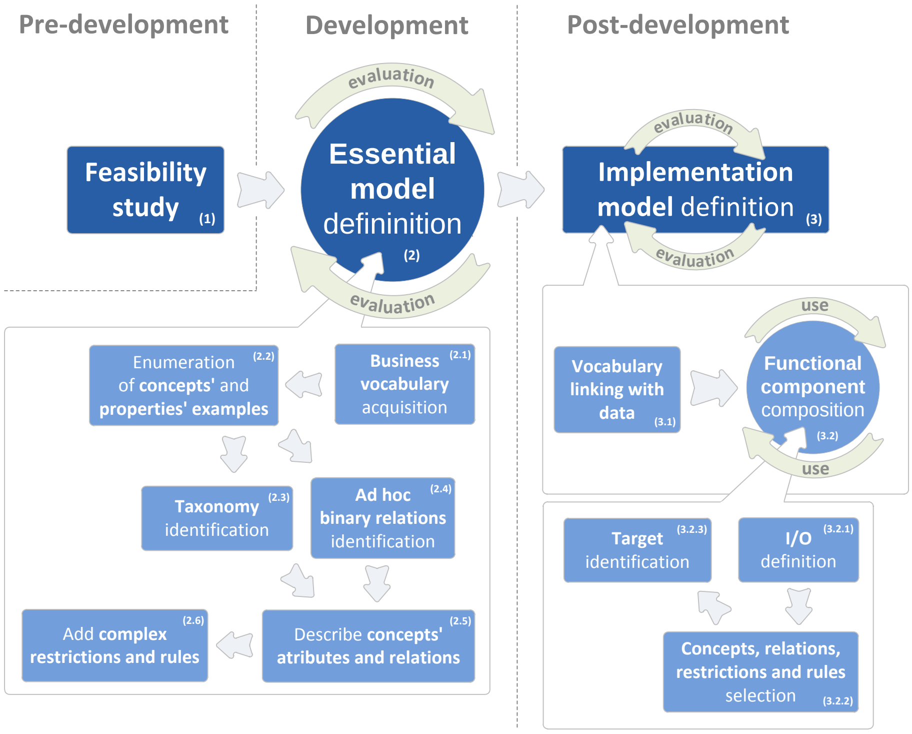

Fig. 1: Process of Rapid Ontology Development (ROD)

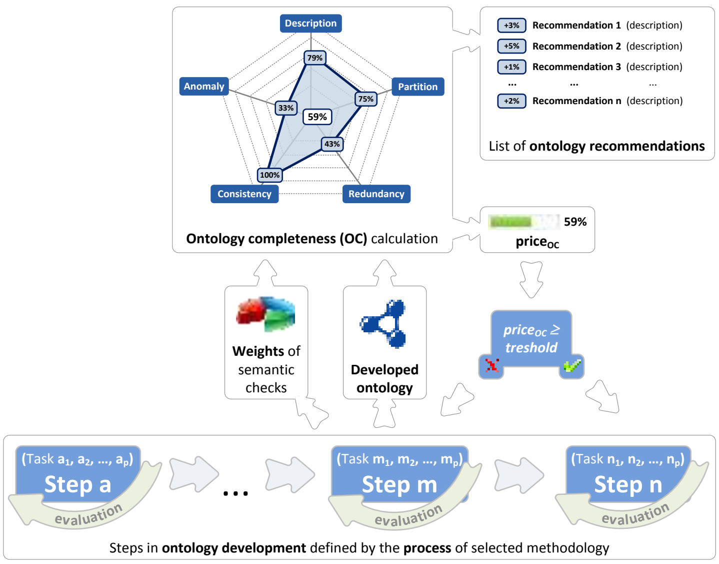

Fig. 2: OC calculation

The ROD development process can be divided into the following stages: *pre-development*, *development* and *post-development* depicted in Fig. 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.

The role of constant evaluation as depicted in Fig. 2 is to guide developer in progressing through steps of ROD process or it can be used independently of ROD process. In latter case, based on semantic review of ontology, enhancements for ontology improvement are available to the developer in a form of multiple actions of improvement, sorted by their impact. Besides actions and their impacts, detail explanation of action is also available (see Fig. 3). When OC measurement reaches a threshold (e.g. 80%) developer 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 developer is aided in progressing through steps of ROD process from business vocabulary acquisition to functional component composition.

Fig. 3: Ontology completeness and improvement recommendation

Ontology completeness (OC) indicator used for guiding developer in progressing through steps of ROD process and ensuring the required quality level of developed ontology 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. Based on these input the output value in an interval *[0, 1]* is calculated. The higher the value, more complete the ontology is. OC is weighted sum of semantic checks, while weights are being dynamically altered when traversing from one phase in ROD process to another. OC can be further defined as

$$OC = \sum_{i=1}^{n} w'_i \cdot leafCondition_i$$

Where *n* is the number of leaf conditions and *leafCondition* is leaf condition, where semantic check is executed. For relative weights and leaf condition calculation the following restrictions apply $\sum_i w'_i = 1$, $w'_i \in [0,1]$ and $leafCondition \in [0,1]$. 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 Fig. 4 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 [37]. 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 are implemented as semantic checks while all preceding elements are aggregation with appropriate weights.

Algorithm for ontology completeness (OC) price is depicted in Fig. 5, where *X* is condition and $w = w(X, Y)$ is the weight between condition *X* and condition *Y*.

Each leaf condition implements a semantic check against ontology and returns value $leafCondition \in [0,1]$.

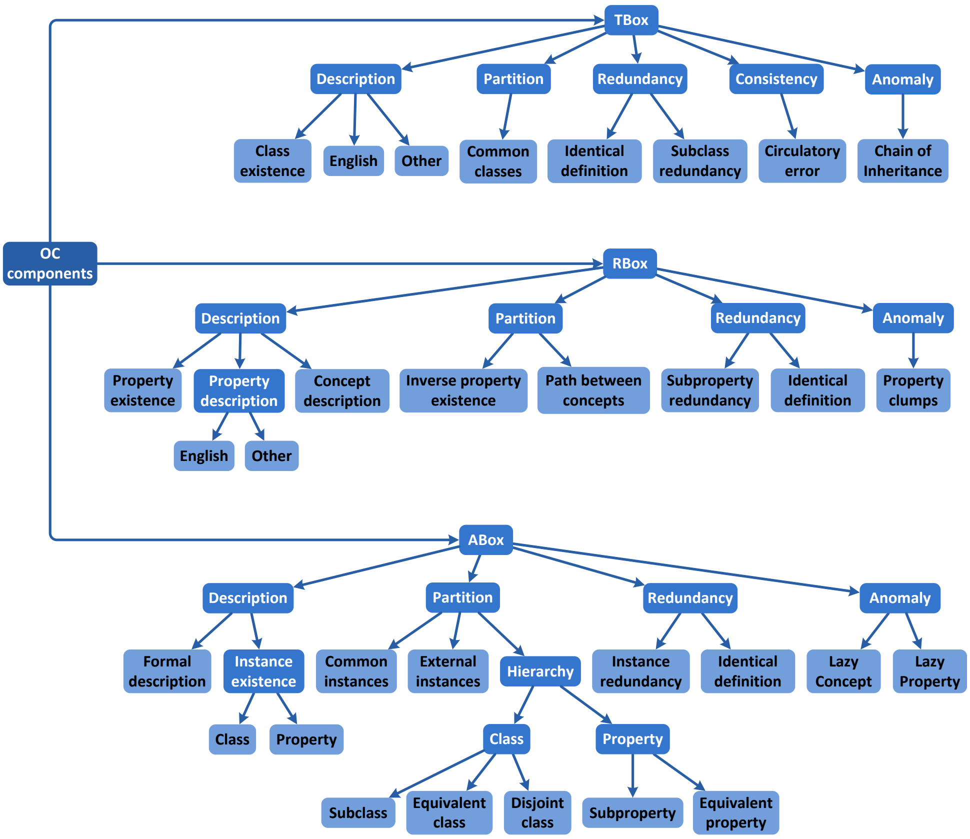

Fig. 4: Ontology completeness (OC) tree of conditions, semantic checks and corresponding weights

' Evaluation is executed on top condition "OC components" with weight 1

Evaluate (X, w)

$price_{OC} = 0$

mark condition X as visited

if not exists sub-condition of X

' Execute semantic check on leaf element

return $w \cdot exec(X)$

else for all conditions Y that are sub-conditions of X such that Y is not visited

' Aggregate ontology evaluation prices

if $w(X,Y) \neq 0$

$price_{OC} \mathrel{+}= Evaluate(Y, w(X,Y))$

return $w \cdot price_{OC}$

End

Fig. 5: Ontology completeness evaluation algorithm

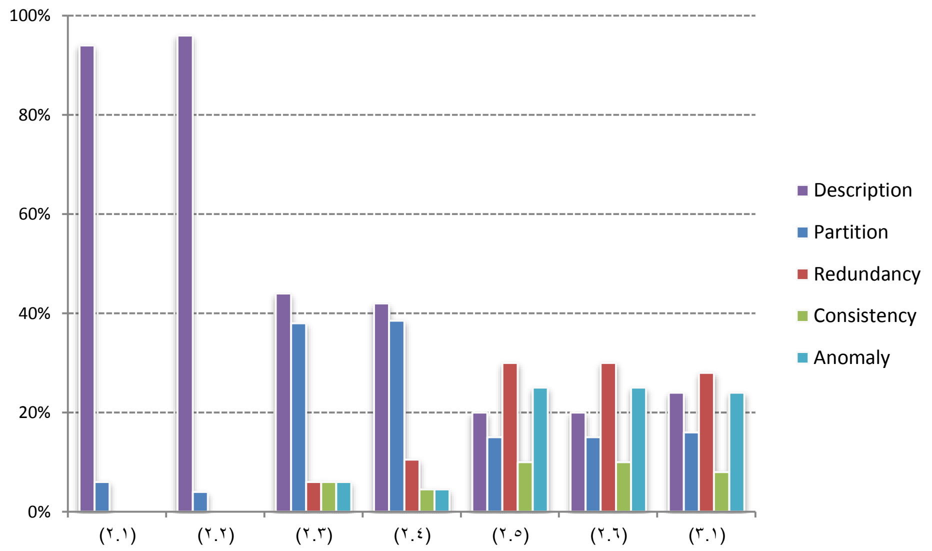

Fig. 6: Impact of weights on OC sublevels in ROD process

Fig. 6 depicts the distribution of OC components (description, partition, redundancy, consistency and anomaly) regarding individual phase in ROD. In first two phases 2.1 and 2.2 developer deals with business vocabulary identification and enumeration of concepts' and properties' examples. Evidently with aforementioned steps emphasis is on description of ontology, while partition is also taken into consideration. The importance of components description and partition is then in latter steps decreased but it still remains above average. In step 2.3 all other components are introduced (redundancy, consistency and anomaly), because developer is requested to define taxonomy of schematic part of ontology. While progressing to the latter steps of ROD process emphasis is on detail description of classes, properties and complex restriction and rules are also added. At this stage redundancy becomes more important. This trend of distributions of weights remains similarly balanced throughout the last steps 2.5 and 2.6 of development phase. In post-development phase when functional component composition is performed, ontology completeness calculation is mainly involved in redundancy, description and anomaly checking.

4 CASE STUDY

4.1 FITS ontology

The problem domain presented in this paper is financial trading and analysis of financial instruments. As already discussed in related work section there are several financial instruments ontologies already present. The purpose of our work was to extend these approaches to the information system level, couple the ontology with reasoning capabilities, define inputs, outputs, dynamic imports and build fully executable Semantic Web solution for financial instruments analysis and trading strategies. For this purpose basic Financial Instruments (FI) ontology was developed following ROD approach (see Fig. 7). The FI ontology introduces basic concepts, including financial instrument, stock exchange market, trading day and analysis. Further details in form of taxonomy are provided for financial instruments, trading day and analysis.

While FI ontology defines elementary entities from financial trading domain, are ontologies that capture trading strategies more complex, including advanced axioms and rules. In our case we have define four different trading strategies: (1) simple trading strategy (STs), (2) strategy of simple moving averages (SMAs), (3) Japanese candlestick trading strategy (JCTs) and (4) strategy based on fundamental analysis (FAs).

Every user has a possibility to define its own trading strategy whether from scratch or reusing existing ones. The main purpose of trading strategies is to examine the instances of *FI:TradingDay* concept and decide whether the instance can be classified into *FI:SellTradingDay* or *FI:BuyTraddingDay*. An example of this process can be found on Fig. 8 where and excerpt from JCTs is presented.

The JCTs is based on price movements which enable to identify patterns from daily trading formations. In this strategy price of a financial instrument is presented in a form of candlestick (low, open, close, high) and several patterns are identified (e.g. doji, hammer, three white soldiers, shooting star etc.). This strategy is rather complex but by following ROD approach (presented in section 3.2) domain experts can define it without being familiar with technical details of knowledge declaration and encoding.

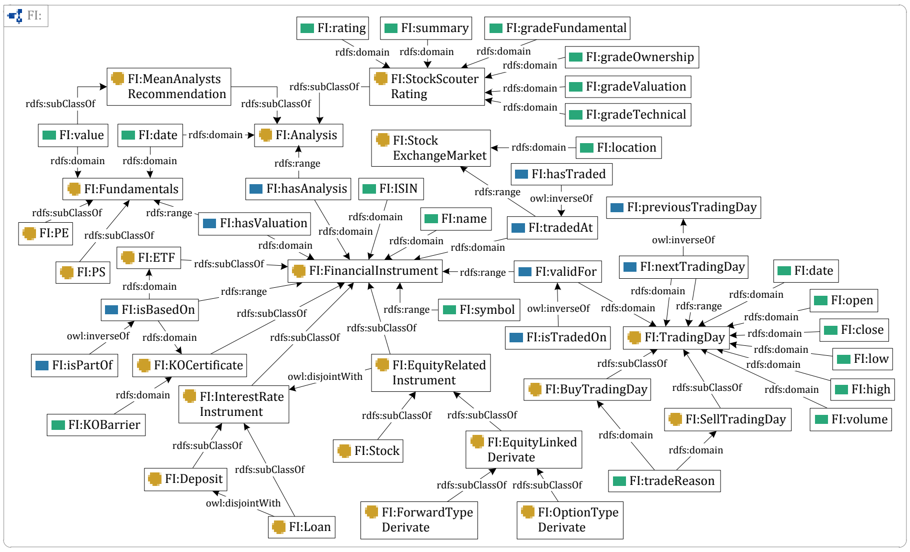

Fig. 7: Excerpt from FITS ontology

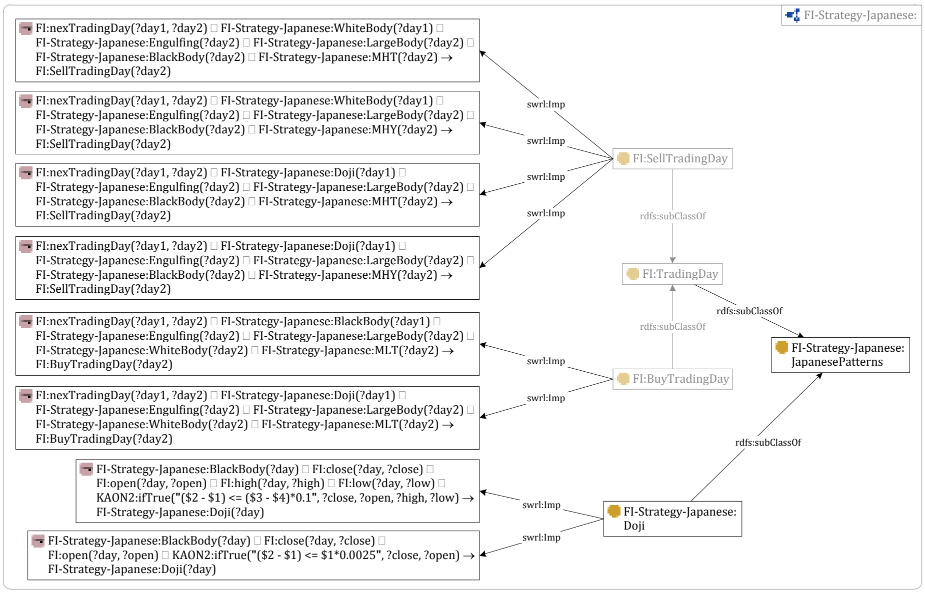

Fig. 8: Excerpt from Japanese candlestick trading strategy

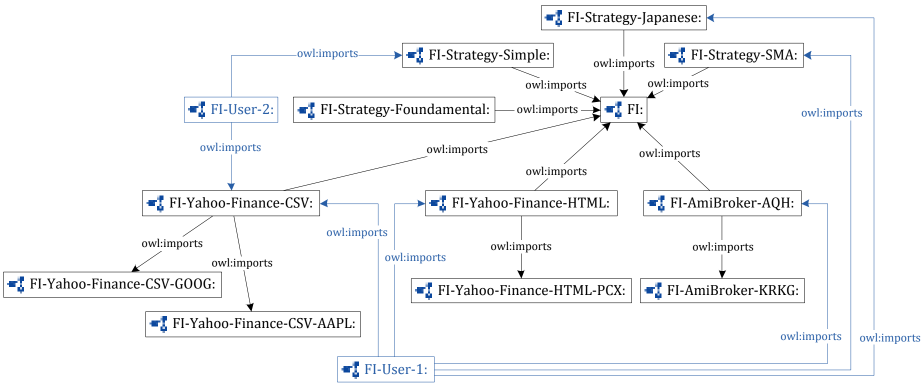

Fig. 9: Composition of final ontology for employment in Semantic Web application

After the selection of desired trading ontologies or composition of existing ones user can define the final ontology (see Fig. 9) which is then coupled with reasoning engine to allow the execution and performing trading analysis on real data available from several sources. At this point the schematic part of ontology (TBox component) is defined and further it still needs to be associated to instances (ABox component) by semantic integration of several data sources, which we will address in the following section 0.

4.2 Semantic integration of data sources

In ROD approach there are several imports available: (1) existing ontologies, (2) relational or analytical databases, (3) CSV file and (4) semi structured data sources (e.g. HTML). In the process of creating FITS ontology the most prominent approach was reusing data from semi structured sources, mainly from HTML pages. When building executable ontology we relied on publicly available data about trading financial instruments, which are available on web pages and in vast majority in an unstructured form. Therefore linking wizard from ROD approach was used which incorporated the technology of regular expression and XQuery formulation for extracting data from semi structured data sources.

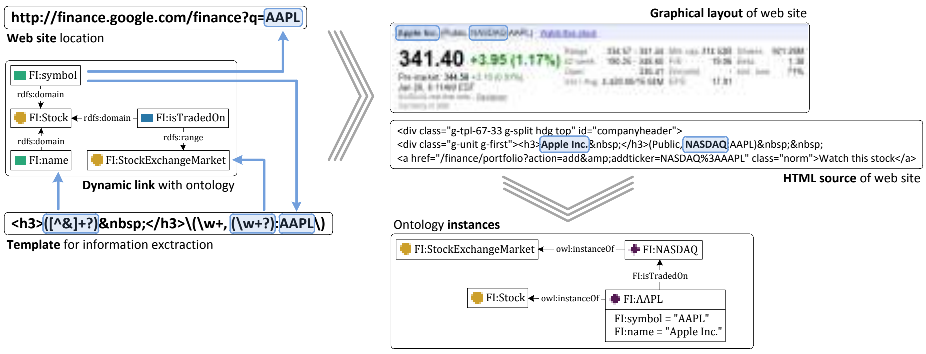

Fig. 10: Dynamic import of data property values related to financial instrument concept from Google Finance web data source

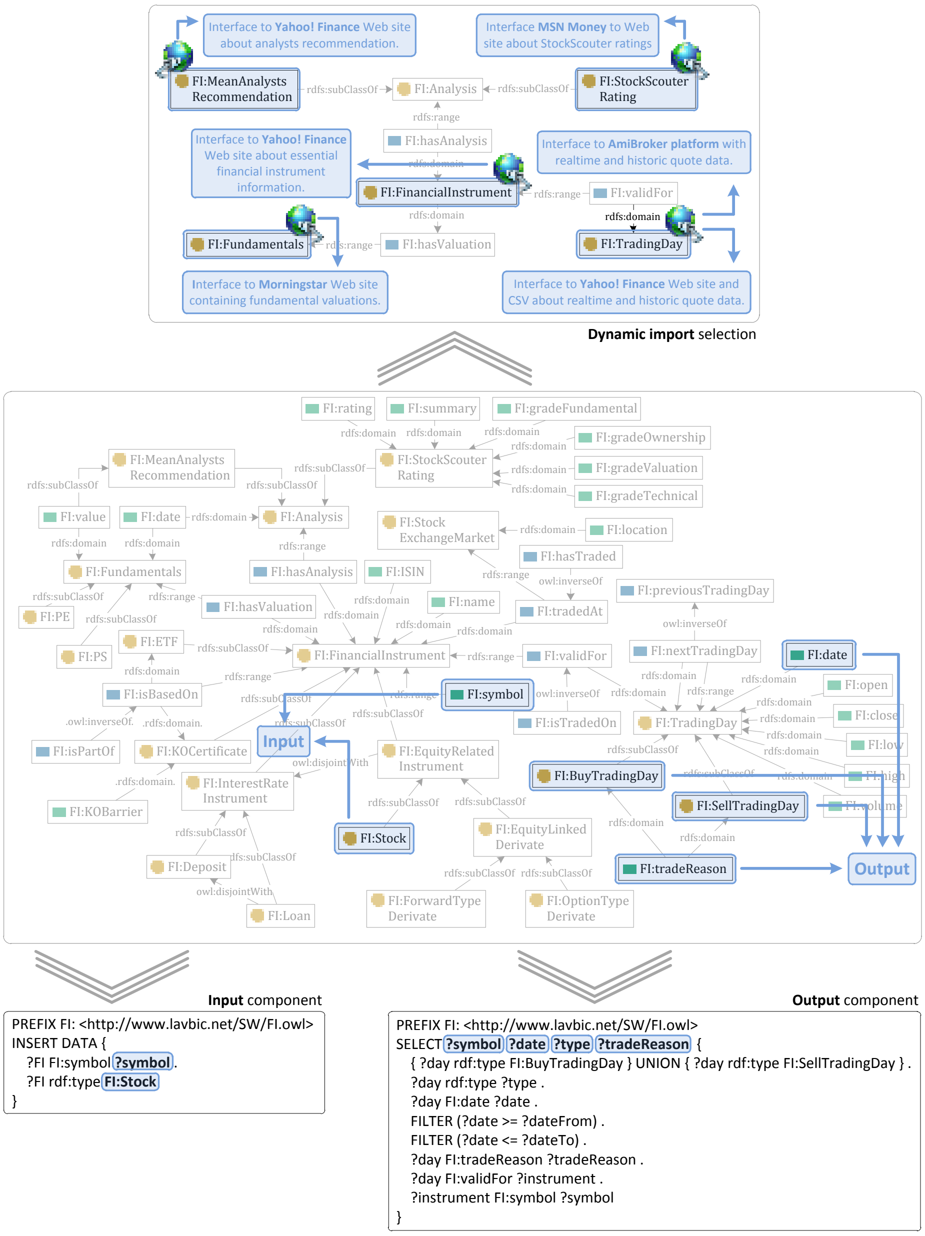

Fig. 11: Dynamic import selection with input and output definition

The role of semantic integration of data sources is to define wrapper to selected data sources and establish dynamic link between ontology entities (e.g. classes, properties etc.) and data source. An example of a simple web site wrapper is depicted in Fig. 10. This wrapper takes as an input financial instrument's symbol and uses Google Finance web page to extract information about financial instrument's name and stock exchange market where is being traded. As a result individuals are added or altered to the knowledge base with FITS ontology. These dynamic links can be defined for every selected entity as depicted in Fig. 11. For our case study there are 6 links defined. As analysis is concerned, mean analysts ratings are extracted from Yahoo! Finance web site, while stock scouter ratings are extracted from MSN money web site. All the essential data about the financial instruments are retrieved from Yahoo! Finance web site, while data about fundamental analysis are obtained from Morningstar web site. The quotes data are transferred from various sources, including historical data from Yahoo! Finance web site and real-time data from AmiBroker trading platform.

The last step in defining the Semantic Web application is to outline the input and the output component. The user can choose within the graphical interface which ontology entities will be used for input and which for output. In our case the input includes the symbol of stock that we want to trade and the outputs includes instances of trading days with buy or sell signals and trade reasons.

4.3 Technology

The selected language for ontology presentation is OWL DL, since it offers the highest level of semantic expressiveness for selected case study and is one of the most widely used and standardized language that has extensive support in different ontology manipulation tools. Besides OWL logical restrictions, Semantic Web Rule Language (SWRL) rules were also employed due to its human readable syntax and support for business rules oriented approach to knowledge management [38].

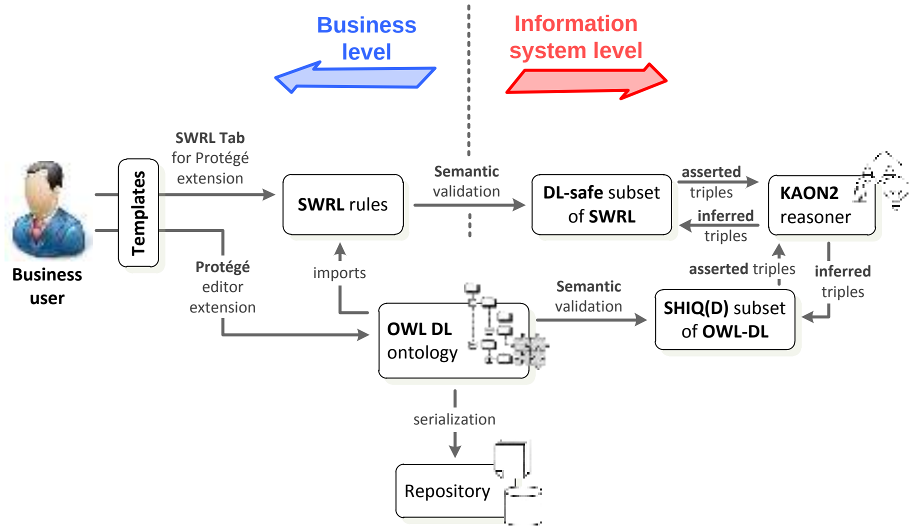

Fig. 12: Prototype of selected case study

The ontology manipulation interface for business users is based on Protégé Ontology Editor and Knowledge Acquisition System and SWRL Tab for Protégé. It enables entering OWL individuals and SWRL rules where a step further is made towards using templates for entering information (see Fig. 12). At the information system level KAON2 inference engine is used to enable inference capabilities. Due to limitations of SHIQ(D) subset of OWL-DL and DL-safe subset of SWRL language, before inference is conducted, semantic validation takes place to ensure that all preconditions are met.

Fig. 13 depicts an example of firing trading rules on a real case scenario. The selected quote is HPQ (Hewlett-Packard) in the trading period of 3 months where several trading rules from Japanese trading strategy are being fired. From the GUI user can always select which subset of trading strategies is used (see section 4.1) and get details about the pattern found.

Fig. 13: GUI example of a Japanese trading strategy analysis on HPQ stock in the period from November 2010 to February 2011

5 CONCLUSIONS AND FUTURE WORK

Current methodologies and approaches for ontology development require very experienced users and developers, while we propose ROD approach that is more suitable for less technically oriented users. With constant evaluation of developed ontology that is introduced in this approach, developers get a tool for construction of ontologies with several advantages: (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. In ontology evaluation several components are considered: description, partition, redundancy, consistency and anomaly. Description of ontology's components is very important aspect mainly in early stages of ontology development and includes existence of entities, natural language descriptions and formal descriptions. This data is furthermore used for advanced axiom construction in latter stages. Partition errors deal with omitting important axioms and can be in a form of common classes, external instances, hierarchy of entities etc. Redundancy deals with multiple information being inferred more than once and includes identical formal definition and redundancy in hierarchy. With consistency the emphasis is on finding circulatory errors, while anomalies do not cause inaccurate reasoning about concepts, but point to badly designed areas in ontology. This includes checking for chain of inheritance, property clumps, lazy entities etc. It has been demonstrated on a case study from financial trading domain that a developer 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. The results from using ROD approach is that the resulting artifact is executable ontology that is available in open format (e.g. OWL and SWRL language) and available for further inclusion. When reusing and building additional applications users have free selection of inference engines and also ontology manipulation tools. Added value is also defined in dynamic imports of data (instances in knowledge base) that can be acquired also at the runtime level.

The future work includes (i) improvement of developed ontology and combining it with other approaches that mainly focus on schematic part of ontology and extend the possible use cases, (ii) improvement of ontology completeness indicator by including more semantic checks and (iii) providing wider support for functional components and

(iv) creating a plug-in for most widely used ontology editors for constant ontology evaluation. One of the planned improvements is also integration with popular social networks to enable developers rapid ontology development based on reuse and therefore employ the community effort in curation process.

6 REFERENCES

1. Corcho, O., M. Fernandez-Lopez, and A. Gomez-Perez, *Methodologies, tools and languages for building ontologies. Where is their meeting point?* Data & Knowledge Engineering, 2003. **46**(1): p. 41-64.
2. SanJuan, E. and F. Ibekwe-SanJuan, *Text mining without document context.* Information Processing & Management, 2006. **42**(6): p. 1532-1552.
3. Veale, T., *An analogy-oriented type hierarchy for linguistic creativity.* Knowledge-Based Systems, 2006. **19**(7): p. 471-479.
4. Bechhofer, S. and C. Goble, *Thesaurus construction through knowledge representation.* Data & Knowledge Engineering, 2001. **37**(1): p. 25-45.
5. Miller, G.A., *WordNet: a lexical database for English.* Communications of the ACM, 1995. **38**(11): p. 39-41.
6. Dong, Y. and M.S. Li, *HyO-XTM: a set of hyper-graph operations on XML Topic Map toward knowledge management.* Future Generation Computer Systems, 2004. **20**(1): p. 81-100.
7. Park, J. and S. Hunting, *XML Topic Maps: Creating and Using Topic Maps for the Web*. 2002, Boston, USA: Addison-Wesley. 640.
8. Jovanović, J. and D. Gašević, *Achieving knowledge interoperability: An XML/XSLT approach.* Expert Systems with Applications, 2005. **29**(3): p. 535-553.
9. Mylopoulos, J., *Information modeling in the time of the revolution.* Information Systems, 1998. **23**(3-4): p. 127-155.
10. Waterson, A. and A. Preece, *Verifying ontological commitment in knowledge-based systems.* Knowledge-Based Systems, 1999. **12**(1-2): p. 45-54.
11. Dzemyda, G. and L. Sakalauskas, *Optimization and Knowledge-Based Technologies.* Informatica, 2009. **20**(2): p. 165-172.
12. Staab, S., et al. *A system for facilitating and enhancing web search*. in *International working conference on artificial and natural neural networks: Engineering applications of bio-inspired artificial neural networks (IWANN '99)*. 1999.
13. Davies, J., R. Studer, and P. Warren, *Semantic Web technologies - trends and research in ontology-based systems*. 2006, Chichester, England: John Wiley & Sons. 327.
14. Wiederhold, G., *Mediators in the architecture of future information systems.* IEEE Computer, 1992. **25**(3): p. 38-49.
15. Heflin, J. and J. Hendler, *Searching the web with SHOE*, in *Artificial Intelligence for Web Search*. 2000, AAAI Press: Menlo Park, USA. p. 36-40.
16. Kesseler, M., *A schema based approach to HTML authoring.* World Wide Web Journal, 1996. **96**(1).
17. Brambilla, M., et al. *A Software Engineering Approach to Design and Development of Semantic Web Service Applications*. in *5th International Semantic Web Conference*. 2006. Athens, USA.

18. Ciuksys, D. and A. Caplinskas, *Reusing ontological knowledge about business process in IS engineering: process configuration problem.* Informatica, 2007. **18**(4): p. 585-602.

19. Magdalenic, I., D. Radosevic, and Z. Skocir, *Dynamic Generation of Web Services for Data Retrieval Using Ontology.* Informatica, 2009. **20**(3): p. 397-416.

20. Schreiber, G., et al., *Knowledge engineering and management - The CommonKADS methodology*. 1999, London, England: The MIT Press: Cambridge, Massachusetts.

21. Uschold, M. and M. King. *Towards a methodology for building ontologies*. in *Workshop on basic ontological issues in knowledge sharing (IJCAI '95)*. 1995. Montreal, Canada.

22. Fernandez-Lopez, M., et al., *Building a chemical ontology using methontology and the ontology design environment.* Intelligent Systems, 1999. **14**(1).

23. Uschold, M. and M. Grueninger, *Ontologies: principles, methods and applications.* Knowledge Sharing and Review, 1996. **11**(2).

24. Kotis, K. and G. Vouros. *Human centered ontology management with HCONE*. in *IJCAI '03 Workshop on Ontologies and Distributed Systems*. 2003.

25. Sure, Y., *Methodology, Tools & Case Studies for Ontology based Knowledge Management*, in *Institute AIFB*. 2003, University of Karlsruhe. p. 332.

26. Nicola, A.D., R. Navigli, and M. Missikoff. *Building an eProcurement ontology with UPON methodology*. in *15th e-Challenges Conference*. 2005. Ljubljana, Slovenia.

27. Vanderlinden, E. *Finance ontology*. 2011; Available from: http://fadyart.com/en/index.php?option=com_content&view=article&id=121:finance-ontology&catid=40:ontology-descriptions&Itemid=68.

28. Farrar, S., W. Lewis, and T. Langendoen. *A Common ontology for linguistic concepts*. in *Knowledge Technologies Conference*. 2002. USA, Seattle.

29. Zhang, Z., C. Zhang, and S.S. Ong, *Building an Ontology for Financial Investment.* Lecture Notes in Computer Science, 2000. **1983**(2000): p. 379-395.

30. Mellouli, S., F. Bouslama, and A. Akande, *An ontology for representing financial headline news.* Journal of Web Semantics, 2010. **8**(2-3): p. 203-208.

31. Qin, H. and M.D. Taffet, *Vocabulary use in XML standards in the financial market domain.* Knowledge and Information Systems, 2004. **6**(3): p. 269-289.

32. Chen, Y., L. Zhou, and D.S. Zhang, *Ontology-supported web service composition: An approach to service-oriented knowledge management in corporate financial services.* Journal of Database Management, 2006. **17**(1): p. 67-84.

33. Cheng, H., Y.C. Lu, and C. Sheu, *An ontology-based business intelligence application in a financial knowledge management system.* Expert Systems with Applications, 2009. **36**(2): p. 3614-3622.

34. Castells, P., et al., *Semantic web technologies for economic and financial information management*, in *Semantic Web: Research and Applications*, C. Bussler, et al., Editors. 2004, Springer-Verlag Berlin: Berlin. p. 473-487.

35. Ying, W., et al., *Design and Development of Financial applications using ontology-based Multi-Agent*

Systems. Computing and Informatics, 2009. **28**(5): p. 635-654.

36. Lavbič, D. and M. Krisper, *Facilitating Ontology development with continuous evaluation.* Informatica, 2010. **21**(4): p. 533-552.

37. Fahad, M. and M.A. Quadir. *Ontological errors - Inconsistency, Incompleteness and Redundancy*. in *International Conference on Enterprise Information Systems (ICEIS) 2008*. 2008. Barcelona, Spain.

38. Horrocks, I., et al., *OWL rules: A proposal and prototype implementation.* Journal of Web Semantics, 2005. **3**(1): p. 23-40.