Unifying Business Concepts for SMEs with Prosecco Ontology

Grzegorz J. Nalepa[*], Mateusz Ślażyński[†], Krzysztof Kutt[‡], Edyta Kucharska[§], Adam Łuszpaj[¶]
AGH University of Science and Technology
Al. Mickiewicza 30, 30-059, Krakow, Poland
[*]gjn@agh.edu.pl, [†]mslaz@agh.edu.pl, [‡]kkutt@agh.edu.pl, [§]edyta@agh.edu.pl, [¶]adam@softhis.com

Abstract—Knowledge management in business information systems often requires a unified dictionary of business concepts, that allows for a transparent integration of such systems. Thanks to it sharing the conceptualization between users becomes possible, and better decision support facilities can be provided. The Prosecco project is a research and development project aims to address the needs and constraints of small and medium enterprises by designing methods that will significantly improve BPM systems. In this paper we focus on the development of ontology-based mechanisms allowing for creating taxonomies of business logic concepts unifying system objects. Building a taxonomy of business concepts shared by number of SMEs targeted in the project and then turning it into a formalized ontology integrating the software components is a major challenge. The paper demonstrates how this ontology is used to unify vocabulary of business processes and rules. The original contribution of this research discussed in the paper is the design and implementation of the ontology, and the demonstration of its practical use in the system.

I. INTRODUCTION

Providing a unified dictionary of business concepts is often a critical aspect of knowledge management in business information systems. Such a dictionary allows for a transparent integration of these systems. Moreover, it allows for sharing the conceptualization between different users providing better decision support facilities. For over a decade, it has been a common approach to support this task with Semantic Web technologies, including ontologies. Specifically formalized ontologies e.g. in OWL are of great practical importance. They become not only a tool for capturing the conceptual description of a business system, but also provide a technical backbone for software modules it is composed of. From the technical point of view, building ontologies is a knowledge engineering task that is currently mostly well supported. However, number of challenges remain, including practical integration of dedicated ontologies in a business information systems, often including business process and business rules (BR) management modules. Possible unification of these through an ontology is a non trivial task of great importance.

The *Prosecco* (Processes Semantics Collaboration for Companies) project [1] is a 32 month research and development project funded by NCBR (2012-2015). Provisioning of Business Process Management (BPM) systems is an important activity of main IT vendors. However, such systems are dedicated

mainly for large companies, organizations and agencies. The motivation of the project is to address the needs and constraints of small and medium enterprises (SME) by designing methods that will significantly improve Business Process Management (BPM) systems. The main goal is to provide technologies that improve and simplify the design and configuration of BPM systems integrated with Business Rules Systems, targeting the management quality and competitiveness improvement. Moreover the project aims at fostering decision making and strategic planning in the SME market sector (mainly in the selected services sector). Specific objectives of the project include: a) development of business process modelling methods taking into account semantic dependencies between business process models and rule models, b) providing recommendation methods for analysis of semantically described business process models, and even more importantly, c) the development of ontology-based mechanisms allowing for creating taxonomies of business logic concepts unifying system objects.

In this paper we focus on the above mentioned objective c). In fact, building a taxonomy of business concepts shared by number of SMEs targeted in the project and then turning it into a formalized ontology integrating the software components is a major challenge. The paper demonstrates how this ontology is used to unify vocabulary of business processes and rules. The original contribution of this research discussed here is the design and implementation of the ontology. Furthermore, we demonstrate its practical use in the Prosecco BPM system.

The rest of the paper is composed as follows: In Section II we briefly discuss the architecture of the Prosecco system. Then in Section III we describe the design process of the ontology. Section IV demonstrates the capturing the semantics of business logic components of the system. Related work is included in Section V. The paper ends with a brief evaluation and summary in Section VI.

II. PROSECCO SYSTEM ARCHITECTURE

From the technical point of view, some of the main objectives of the Prosecco system that meets the project goals include the development of the:

1) integrated business logic model composed of business processes and rules,
2) runtime environment suitable for execution of the model,
3) recommendation modules for the design and use of business artifacts,

---
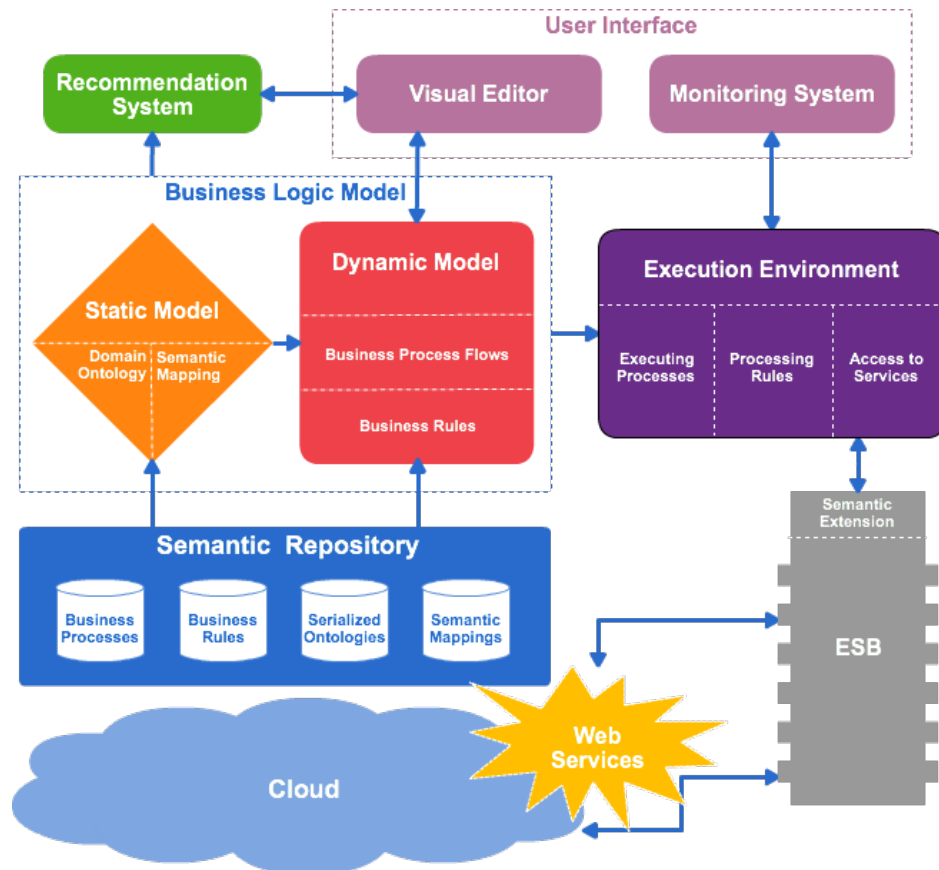
[1]See http://prosecco.agh.edu.pl for the project website.

Figure 1. Outline of the Prosecco system architecture.

4) repository of business objects,
5) enterprise service bus (ESB) integrating the system components in a cloud environment, and
6) system ontology based on the taxonomy of shared business concepts.

The outline of the system can be observed in the Fig. 1.

The end users of the Prosecco system are SMEs, in fact selected employees as well as management of companies. The system aims at supporting carrying out of the main business processes of a company. Moreover, these processes are accompanied by business rules, capturing the details of the business logic, including lower level processing, as well as high-level constraints. Within the system these are modeled with the help of BPMN (Bussiness Process Model and Notation), and appropriate BR models identified with SBVR (Semantics of Business Vocabulary and Business Rules) and implemented with the help of the Drools BRMS as well as the HeaRT rule engine.

The process models and rule models are based on concepts captured during initial structured interviews with SMEs. The resulted taxonomy of concepts was used to design and implement the Prosecco ontology that works as the main unifying backbone of the system. It contains all the business terms needed for expressing and capturing the artifacts in business process models and rules. Moreover, it allows to monitor the

execution of these models on a semantic level. The execution environment uses the ontology, so users can actually trace how these concepts are used in the executed processes and rules.

From the point of view on the integrators of the system, the ontology supports recommendation mechanism that allows for adaptation of business process and rule models to the needs of specific companies. Furthermore, the repository of business objects uses semantic annotation based on this ontology. It makes it possible for an easy retrieval of needed objects based on semantic queries. In the next section the design of the Prosecco ontology is described.

III. DESIGN OF THE PROSECCO ONTOLOGY

A. Requirements for the Ontology

The ontology that will be a part of a management system has to identify key concepts and relations describing static aspect of opertions of considered SMEs. Furthermore, it must be integrated with business processes and rules that describe the dynamic aspect of SME management.

Based on Prosecco system architecture described above, some assumptions for the Prosecco Ontology were established:

- Ontology should be modularized: each module should describe different domain (different topic).
- Ontology is designed to be used as dictionary of concept describing elements of business processes and rules.

- Ontology should be defined in a simple description logic language (OWL Lite$_A$, OWL 2 QL, or eventually OWL 2 RL).
- Each concept and role must be documented with short description (what exactly the term means) and possible connections with other ontology elements.
- Ontology is designed for Polish companies so all concepts should be written in Polish language and adjusted to Polish law.
- There should be a possibility to extend ontology by Prosecco system users.

B. How the Ontology was built

There are some methods and tools that support ontology engineering, but there are no standard approaches how to develop ontologies in general [1]. Brief survey [2] indicates that:

- Existing methods are relatively old.
- The methods can be grouped into categories: incremental and iterative, or more comprehensive ones.
- Most methodologies consists of same main steps: assessment, deployment, testing and refinement.
- Most studies suffer from lack of information about tools.
- Only few recent studies suggests decrease in research activity in this field.

These conclusions encouraged Prosecco analysts to review the most significant and best-established methodologies for creating and managing ontologies [1], [2]. Three main approaches were considered: TOVE (*Toronto Virtual Enterprise*) [3], Enterprise Model Approach [4] and METHONTOLOGY [5]. Analysis of these methodologies resulted in specifying number of steps that are required to develop proper ontology. These steps designate the sequence of work in Prosecco project:

1) *motivating scenarios*: exemplary problems that ontology should resolve,
2) *competency questions*: which questions the ontology should answer,
3) *knowledge acquisition*: interviews with experts, domain texts analysis,
4) *conceptualisation*: extraction of concepts, objects and relations between them,
5) *integration*: consideration of existing ontology reuse,
6) *implementation*: expressing ontology in terms of a formal system,
7) *evaluation*: validation and verification, check for completeness, redundancy and contradictions,
8) *documentation*: comments and documents describing the concepts and relations in ontology.

In Prosecco project each step was performed incrementally in few iterations.

Besides comprehensive methodologies mentioned above, there are projects that provide "good practices" for selected steps in ontology engineering process, e.g. Ontolingua, CommonKADS, KACTUS, PLINIUS, ONIONS, Mikrokosmos, MENELASPHYSSYS, SENSUS (for overview see [1]).

Examined methodologies and "good practices" postulate separation of informal and formal part of ontology development. Some of them suggest introducing a middle representation, that will be a connector between unstructured text and a set of formal axioms. This layer can use structured language or simple graphical language (see IDEF5 methodology [6]). In Prosecco project three "formalization levels" were used:

1) *Unstructured notes*, gathered during interviews with experts (SME workers).
2) *Taxonomy in structured language*, prepared during conceptualisation step. Written down using Prolog language and visualised using custom scripts (see Fig. 2).
3) *Formal ontology*, written in OWL 2 language.

There are three main approaches to defining taxonomies [7]:

- *top-down*: the most general concepts are defined at the beginning and then more specific ones are gradually determined,
- *bottom-up*: low-level concepts and relations are defined at first and then they are generalized,
- *middle-out*: conceptualisation begins with identifying the most relevant concepts; more general and specified ones are determined as needed. It was selected as the best suitable approach for the Prosecco project.

C. Description of the Ontology

Prosecco Ontology was partitioned into several parts. Each represents a different area in SME management:

1) Project artifacts (PL: *Artefakty*) – components connected with planning and implementation of the project.
2) Organizations (PL: *Organizacje*) – types of companies and their main properties (e.g. e-mail, VAT number).
3) Organization structure (PL: *StrukturyOrganizacji*) – elements that describe company structure, e.g. customer care department.
4) Documents (PL: *Dokumenty*) – concepts and relations associated with various kinds of documents.
5) People (PL: *Osoby*) – depiction of people: key properties (e.g. name, surname) and occupation.
6) Methodologies (PL: *Techniki*) – things connected with methods and tools used in company, e.g. code repository.
7) Resources (PL: *Zasoby*) – grouping into human resources, tangible and intangible resources.
8) Events (PL: *Zdarzenia*) – event types and their properties.
9) Prosecco (PL: *Prosecco*) – main module, the parent of other modules that integrates them and adds additional values common to all of them, e.g. uid or name.

Each part consists of at least one concept, object or data-property from another module. These nine parts constitute modules of the Prosecco Ontology that consists of: 86 classes, 70 object properties and 64 data properties. All of them are described by 1042 axioms. Classes are arranged in a hierarchy using `rdfs:subclassof` properties. Besides this simple generalization/specialization relations, each class can be inferred from object and data properties that this class has, e.g.
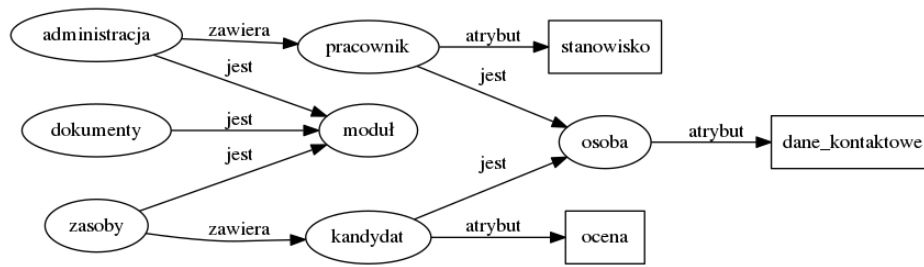
Figure 2. Visualisation of middle layer taxonomy.

`Project` is something that aggregates some `Tasks` or something that is managed by the `Project leader`. Axioms allows inference only in one way: if something aggregates `Tasks`, it must be a `Project`. Fact that something is a `Project` does not infers conclusion that it must aggregates `Tasks`. Example (in Polish) is presented on Fig. 3. For axioms that connect classes with object and data properties, cardinality was also defined as it is presented in Table I.
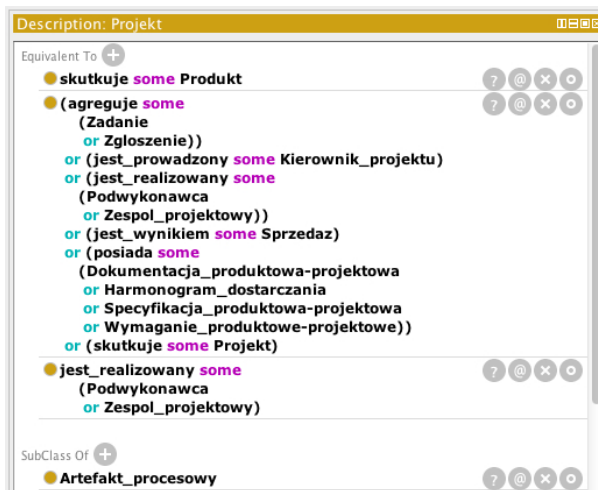

Figure 3. Exemplary concept definition in Prosecco Ontology.

Cardinality	Protégé code	Example
0, 1	max 1	home_address max 1 string
1	exactly 1	created exactly 1 dateTime
$[0, \infty)$	only	description only string
$[1, \infty)$	some	*Not used yet. Defined for future use*

Table I
CARDINALITY DEFINED IN AXIOMS THAT CONNECTS CLASSES
WITH OBJECT AND DATA PROPERTIES.

D. Tools and resources used

Different tools were used in different steps during the development process of the Prosecco Ontology.

Taxonomy in structured language:

- *SWI-Prolog*[2] for interpreting concepts and rules that were recorded with prepared templates for concepts `c(?Id, ?Name)`, relations `r(?Subject, ?Predicate, ?Object)`

---

[2]See: http://www.swi-prolog.org/.

and attributes `a(?Class, ?AttributeName, ?ListOfPossibleValues)`

- *Custom scripts* for validation and visualisation using *Graphviz* tool[3] (see Fig. 2).

Formal ontology:

- *Protégé*[4] editor was used for preparing Ontology in OWL 2 language. It was selected because of its documentation, users support, official handbooks (e.g. [8]) and huge amount of design patterns [9].
- *Git repository* for collaborative work with following iterations of Ontology. *OntoCVS*[5] plugin was used for better tracking of Ontology changes.
- *Ontology modularization tool*. Ontology was implemented as a coherent model. In last step it was divided into modules. Different ontology modularization techniques were analysed. One of the most important is the Atomic Decomposition method [10]. It is based on atoms that consist of a set of axioms that occur together. Correlation between axioms inside atom is so strong, that this axioms have to be together. Single module is a single atom or a set of atoms. Using atomic decomposition alorithm [11] results in a set of modules. For Prosecco Ontology this algotihm generates 62 modules, where one of them consists of 574 axioms (of overall number 1042 axioms). These modules were not functional and decision about manual modularization was made.

The whole development process of the ontology lasted 12 months, including preliminary analytic meetings with SMEs. The development team included 4 business analytics and 4 developers; a 2 person evaluation team was also included.

IV. CAPTURING THE SEMANTICS OF BUSINESS LOGIC

In order to integrate the complete ontology model into the Prosecco architecture some additional works had to be performed. Different formal models, designed to capture dynamics of business processes, were created by separate teams, and they had to be matched with the ontology's concepts. This refinement process can be divided into three separate steps.

Firstly, both BPMN and rule models had very uneven granularity levels: abstract concepts were often mixed with detailed and concrete names of particular tools and documents. Due

---

[3]See: http://www.graphviz.org/.
[4]See: http://protege.stanford.edu/.
[5]See: https://code.google.com/p/ontovcs/.

to these inconsistencies integration of both dynamic models was nearly impossible. During the first step, all the concepts in business processes and rules were generalized or rejected where necessary to represent the same level of abstraction as ontology. The resulting models were smaller and more general than previous versions:

- Exemplary rule that was rejected due to the limitation only to selected SMEs: `It is necessary that each Employee has the available vacation days.`
- Exemplary rule before: `It is possible that a *programmer* can change a task status [...]` and after generalization: `It is possible that a` **specialist** `can change a task status [...]`.

Secondly, due to the incoherences in concepts' naming, both models have been aligned with names provided in the ontology. Every noun and verb in BPMN and rule models was compared against the ontology: both name and the possible usage described by object and data properties. If equivalent concept exists, it replaced word used in dynamic model. Furthermore ontology had to be filled with lacking concepts. In particular there were added many data type attributes due to the low-level characteristics of rules – every attribute used in conditional or decisive part had to be formally specified inside the related class. It was a very important step to achieve an automatic execution of hybrid BPMN/rule models.

As a result, both dynamic models of business logic shared the same taxonomy, along with attributes' types and relations between the concepts. Unfortunately, neither Prosecco's external services nor environments used to execute and validate these models do not support semantic data directly and there had to be introduced a solution to integrate them with ontology in an indirect manner. The proposed solution is based on automatically generated Plain Old Java Objects (so-called POJO), which represent ontological concepts in serializable and executable manner. The ontology is used to infer: a) types of classes (corresponding to the ontological classes), b) types of the class attributes (corresponding to data type properties), and c) relations between the classes (corresponding to object properties).

Within the practical implementation of the architecture depicted in Fig.1 we integrated the Prosecco ontology as the core of the system. The Prosecco system uses Activiti as the business process engine [12]. Apart from the type system generated according to ontology, Activiti uses also other semantization techniques. The Prosecco repository consists of several sub-repositories, including business process models for Activiti, rule models for Drools, user information in ACL, and system history managed by the Cassandra tool. Data repositories are separate from the type system and existing instances are continuously synchronized with ontology. Thanks to the use of the ontology the project assumes that all data types and their instances existing within the system are consistent with the ontology.

V. RELATED WORK

During the development of the Prosecco Ontology, existing organization ontologies were analysed and compared with Ontology discussed in this paper. Four models were considered:

1) *An organization ontology* [13] – developed by W3C and Epimorphics Ltd. and implemented in simple description logics language ($\mathcal{SIF}$(D)). It is aimed at describing basic organizational information in a number of domains. Comparing to the Prosecco Ontology, this model is less acurate and more general.
2) *IntelLEO Organization Ontology* [14] – created during IntelLEO project and currently not further developed. It models organization structures using people responsibilities and relations between them. This ontology models access rights what is outside the scope of Prosecco Ontology. On the other hand, it is not suitable for describing business processes.
3) *Ontology for organizations* [15] – is a part of a larger one that is used to annotate The Gazette[6] contents. It is best suitable for characterize activities of the organization (e.g. if it is a government or charity organization). This ontology does not provide concepts for describing organization structure, what was one of main goals of Prosecco Ontology.
4) *PROTON (PROTo ONtology)* [16] – upper-level ontology that is simple and well-documented. This is the biggest one among analysed models (it consists of 250 classes). As Prosecco Ontology, it describes projects, documents and organization structure. It lacks the support for business processes modelling.

Three more models draw attention, but due to the fact they are not public, there was no possibility to analyse them insightful and compare them to the Prosecco Ontology:

1) *Unified Enterprise Modelling Ontology (UEMO)* [17] – is based on UEML (Unified Enterprise Modeling Language) and depicts companies and information systems. It is coupled with BPM (Business Process Management).
2) *O-CREAM-v2, a core reference ontology for the CRM domain* [18] – is a very detailed CRM (Customer Relationship Management) ontology. One of its drawbacks is the lack of emphasis on services.
3) *WeCoTin* [19] – ontology designed to modelling process of matching the offer to the customer's requirements.

VI. EVALUATION AND SUMMARY

Knowledge management in business information systems often requires a unified dictionary of business concepts, that allows for a transparent integration of such systems. Thanks to it sharing the conceptualization between users becomes possible, and a better decision support facilities can be provided. In this paper we focused on the development of ontology-based mechanisms allowing for creating taxonomies of business logic concepts unifying system objects considered

---

[6]See: https://www.thegazette.co.uk/.

in the Prosecco project. We demonstrated how this ontology is used to unify vocabulary of business processes and rules in the Prosecco BPM system. The practical contribution discussed in the paper is the design and implementation of the ontology, and the demonstration of its practical use in the system.

The Prosecco Ontology, whose development and structure were described in the earlier sections, fulfills all the requirements of the project. As a result of careful design and iterative refactoring it can be considered as a formal definition of data and concepts appearing in the other parts of the Prosecco system, especially BPMN models and business rules. In comparison to less formalized data models it has certain advantages like mature tools, unambiguous semantics and possibility to formally prove coherence thanks to the available reasoners. Moreover, the formal approach to design system around a semantic model has proven to be a valid option among the standard solutions – the Prosseco Ontology is already successfully used to generate model layer of the system's services and future plans include usage of semantic queries to enhance data retrieval and storage.

A certain limitation of the current version of the ontology is the fact that it is based on the Polish vocabulary. However, it was a conscious decision and the requirement of the project related to the Polish regulations. In general most of the modules of the ontology could be easily adapted to other SMEs so the ontology would be suitable for international use. However, from the practical point of view an adaptation to some specific regional regulation (e.g. EU) should be considered. Certain parts specific to the Polish law would have to be replaced.

The biggest disadvantage of the proposed solution is lack of a direct integration between different modeling techniques and technologies; due to this limitation there was proposed an intermediate translation of semantic classes in a form of the corresponding Java classes. Currently the works are focused on the creation of environments which could help to model business logic enhanced by use of the ontological background.

Our future work include the use of the ontology to improve the usability of the execution environment. We are considering how semantic annotations could enhance intelligibility of the business logic. Thanks to the already integrated domain ontology, it would be easy to enhance rules with semantic annotations, leading to more meaningful design of the system, and possibly porting it to mobile platforms [20], and simplyfying a formalized modeling [21]. We are also working on a tighter integration of the components of the BPM system, including semantic tracking of business objects and a BP editor [22]. Moreover, we consider possible extensions of the ontology towards the needs of other SMEs from different sectors.

Acknowledgment

References

[1] D. Jones, T. Bench-Capon, and P. Visser, "Methodologies for ontology development," in *Proceedings of IT&KNOWS Conference of the 15th IFIP World Computer Congress*, 1998.

[2] M. Bergman, "A brief survey of ontology development methodologies," Aug 2010. [Online]. Available: http://www.mkbergman.com/906/a-brief-survey-of-ontology-development-methodologies/

[3] M. Gruninger and M. S. Fox, "The design and evaluation of ontologies for enterprise engineering," in *Workshop on Implemented Ontologies, European Workshop on Artificial Intelligence, Amsterdam, The Netherlands*, 1994.

[4] M. Uschold and M. King, "Towards a methodology for building ontologies," in *IJCAI-95 Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal, Canada*, 1995.

[5] M. Fernández-López, A. Gómez-Pérez, and N. Juristo, "Methontology: from ontological art towards ontological engineering," in *Proceedings of the Ontological Engineering AAAI-97 Spring Symposium Series*. American Association for Artificial Intelligence, 1997.

[6] P. C. Benjamin, C. P. Menzel, R. J. Mayer, F. Fillion, M. T. Futrell, P. S. deWitte, and M. Lingineni, "IDEF5 method report," Knowledge Based Systems, Inc, Tech. Rep., 1994.

[7] M. Uschold and M. Gruninger, "Ontologies: Principles, methods and applications," *The knowledge engineering review*, vol. 11, no. 02, pp. 93–136, 1996.

[8] M. Horridge, H. Knublauch, A. Rector, R. Stevens, and C. Wroe, "A practical guide to building OWL ontologies using the Protege-OWL plugin and CO-ODE tools edition 1.0," Aug. 2004. [Online]. Available: http://www.co-ode.org/resources/tutorials/ProtegeOWLTutorial.pdf

[9] "Ontologydesignpatterns.org." [Online]. Available: http://ontologydesignpatterns.org/

[10] C. Del Vescovo, B. Parsia, U. Sattler, and T. Schneider, "The modular structure of an ontology: Atomic decomposition," in *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, vol. 22, no. 3, 2011, pp. 2232–2237.

[11] D. Tsarkov, C. Del Vescovo, and I. Palmisano, "Instrumenting atomic decomposition: Software apis for owl," in *Proceedings of OWLED'13: the 10th International Workshop on OWL: Experiences and Directions*, 2013.

[12] T. Rademakers, T. Baeyens, and J. Barrez, *Activiti in Action: Executable Business Processes in BPMN 2.0*, ser. Manning Pubs Co Series. Manning Publications Company, 2012.

[13] D. Reynolds, "The organization ontology," W3C, Recommendation, Jan. 2014. [Online]. Available: http://www.w3.org/TR/vocab-org/

[14] J. Jovanovic and M. Siadaty, "IntelLEO organization ontology," IntelLEO, Working Draft, Apr. 2011. [Online]. Available: http://intelleo.eu/ontologies/organization/spec/

[15] J. Tennison, "The london gazette ontology, organisation module," 2008. [Online]. Available: https://www.thegazette.co.uk/def/organisation.owl

[16] K. Simov, A. Kiryakov, I. Terziev, D. Manov, M. Damova, and S. Petrov, "Proton ontology (proto ontology)," 2005. [Online]. Available: http://www.ontotext.com/documents/proton/protontop.ttl

[17] A. L. Opdahl, G. Berio, M. Harzallah, and R. Matulevičius, "An ontology for enterprise and information systems modelling," *Applied Ontology*, vol. 7, no. 1, pp. 49–92, 2012.

[18] D. Magro and A. Goy, "A core reference ontology for the customer relationship domain," *Applied Ontology*, vol. 7, no. 1, pp. 1–48, 2012.

[19] J. Tiihonen, M. Heiskala, A. Anderson, and T. Soininen, "Wecotin–a practical logic-based sales configurator," *AI Communications*, vol. 26, no. 1, pp. 99–131, 2013.

[20] G. J. Nalepa and S. Bobek, "Rule-based solution for context-aware reasoning on mobile devices," *Computer Science and Information Systems*, vol. 11, no. 1, pp. 171–193, 2014.

[21] M. Szpyrka, G. J. Nalepa, A. Ligęza, and K. Kluza, "Proposal of formal verification of selected BPMN models with Alvis modeling language," in *Intelligent Distributed Computing V. Proceedings of the 5th International Symposium on Intelligent Distributed Computing – IDC 2011, Delft, the Netherlands – October 2011*, ser. Studies in Computational Intelligence, F. M. Brazier, K. Nieuwenhuis, G. Pavlin, M. Warnier, and C. Badica, Eds. Springer-Verlag, 2011, vol. 382, pp. 249–255. [Online]. Available: http://www.springerlink.com/content/m181144037q67271/

[22] K. Kluza, K. Kaczor, and G. J. Nalepa, "Enriching business processes with rules using the Oryx BPMN editor," in *Artificial Intelligence and Soft Computing: 11th International Conference, ICAISC 2012: Zakopane, Poland, April 29–May 3, 2012*, ser. Lecture Notes in Artificial Intelligence, L. Rutkowski and [et al.], Eds., vol. 7268. Springer, 2012, pp. 573–581. [Online]. Available: http://www.springerlink.com/content/u654r0m56882np77/