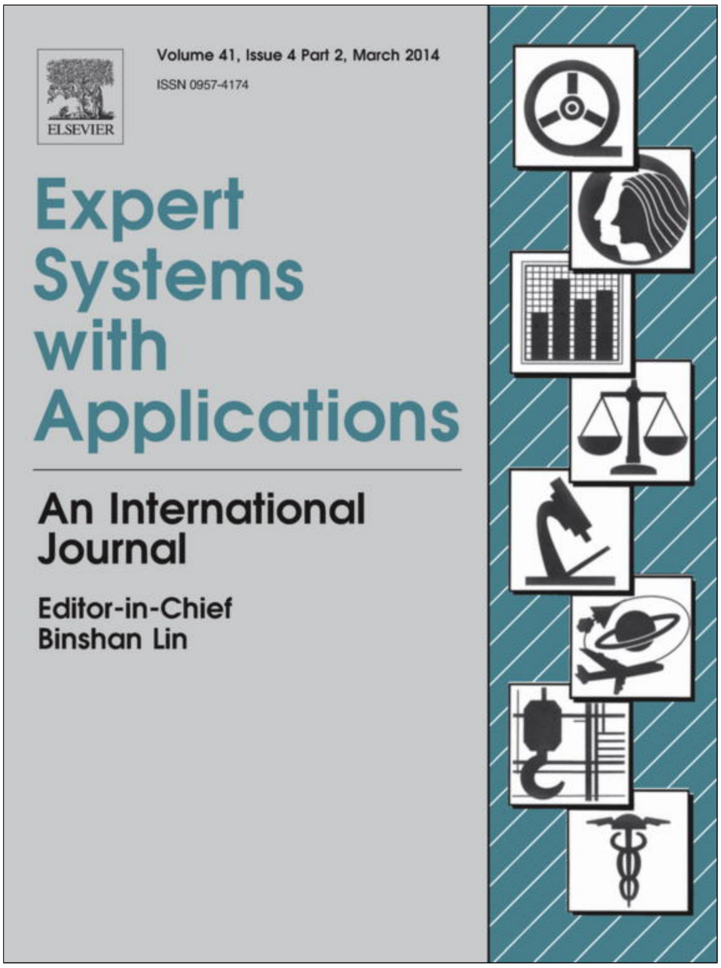
Provided for non-commercial research and education use.
Not for reproduction, distribution or commercial use.



This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

http://www.elsevier.com/authorsrights

Author's personal copy

Expert Systems with Applications 41 (2014) 1576–1583

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Approaching the feasibility of SBVR as modeling language for ontology development: An exploratory experiment

Emiliano Reynares [a,*], María Laura Caliusco [a], María Rosa Galli [a,b]

[a] CIDISI Research Center, UTN Santa Fe, B. Lavaysse 610, S3004EWB Santa Fe, Argentina
[b] INGAR-UTN-CONICET, Avellaneda 3657, S3002GJC Santa Fe, Argentina

ARTICLE INFO

Keywords:
Exploratory experiment
Ontology development
UML
SBVR

ABSTRACT

Ontology development techniques still constitute an open research area despite its importance in semantic aware information systems. Until now, most methods have used UML in supporting ontology development process. Recent works propose the mapping of business rules expressions to ontology statements as a building technique by means of SBVR language. However, there is still no experimental research comparing such approaches.

Aim of this work is to evaluate the feasibility of mapping business domain expressions to ontology statements. An exploratory experiment comparing performance of techniques based on UML and SBVR languages is presented. Comparison is rooted in the quality assessment of the ontologies developed by 10 equally sized groups randomly conformed by 30 undergraduate engineering students and applying such techniques.

Developed ontologies largely outperform the minimally acceptable quality, according to the considered quality assessment framework. There is no statistical significant difference between the quality scores of the ontologies developed by means of UML and SBVR techniques, in any of the assessed quality dimensions.

The feasibility of mapping business domain expressions to ontology statements is shown: ontologies developed by means of a SBVR based approach at least equate the quality of ontologies developed by using an UML based method. Results confirm previous research about the effectiveness of UML approaches for conceptualizing lightweight ontologies while stressing the potential of the SBVR language to express complex notions of a domain of interest. The potential of SBVR to OWL 2 mappings as an ontology development technique worthy of further study is highlighted.

© 2013 Elsevier Ltd. All rights reserved.

1. Introduction

Use of semantic technologies for creating more intelligent and effective enterprise information systems has increased considerably in the last years. Several examples highlight the strong benefits and wide applicability of ontologies in such kind of systems. For example, ontology reasoners could be used for automatically proving consistency of business models (Ceravolo, Fugazza, & Leida, 2007; Karpovic & Nemuraite, 2011; Alberts & Franconi, 2012; Franconi & Mosca, 2012; Beydoun, Low, Tran, & Bogg, 2011). Ontologies intended to be used in the analysis stage of a software development process could be generated from main business knowledge sources (Calero, Ruiz, & Piattini, 2006; Myrgioti, Bassiliades, & Miliou, 2013). Ontologies could be also used to encapsulate the declarative specification of business knowledge into information software systems, enabling unambiguous representation of knowledge and efficient management of highly dynamic environments (Demuth & Liebau, 2007; Ruotsal, 2010; Reynares, Caliusco, & Galli, 2012; Shue, Chen, & Shiue, 2009; Chen, Huang, Bau, & Chen, 2012). Despite those significant applications, ontology development methodologies remain to be an open research area where the proposals can be grouped in two main approaches.

The former involves the best practices from the Knowledge Engineering field (Gómez-Pérez, Fernández-López, & Corcho, 2004). Such practices are not usually a part of the toolbox involved in the development of software information system and their performing by software engineering professionals and researchers implies additional learning experience.

With the aim of avoiding such issue, a second group of methodologies stems its characteristics from widely used standards in the software engineering field (Nicola, Missikoff, & Navigli, 2009). Usually, such approaches made use of the Unified Modelling Language (UML) (Object Management Group (OMG), 2011) and the Object Constraint Language (OCL) (Object Management Group (OMG),

* Corresponding author. Tel.: +54 0342 460 2390x258/106.
E-mail addresses: ereynares@frsf.utn.edu.ar (E. Reynares), mcaliusc@frsf.utn.edu.ar (M.L. Caliusco), mrgalli@santafe-conicet.gov.ar (M.R. Galli).

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2013.08.054

Author's personal copy

2012) for the conceptual modelling stage of the ontology development process (Wang & Chan, 2001; Guizzardi, Herre, & Wagner, 2002; Nicola et al., 2009).

UML is a general purpose modelling language conceived by the software engineering community and later standardized by OMG. It is strongly rooted in a set of graphic notation techniques to create visual models of object oriented software systems. Meanwhile, OCL is an adopted OMG standard used to describe expressions on UML models. These expressions typically specify invariant conditions that must hold for the system being modelled or queries over objects described in a model. The widespread and accepted nature of UML into the software engineering community, its standardized graphical representation of models, the wide range of available tools, and the extensible nature of the language are the main advantages of its use in an ontology building process. Despite such benefits, the lack of reliable set semantics and model theory for UML prevents the use of automated reasoners on UML models. Moreover, as UML lacks a formal model theoretic semantics, OCL also has neither a formal model theory nor a formal proof theory, and thus cannot be used for automated reasoning (Object Management Group (OMG), 2009).

Recent works propose the mapping of business rules expressions to ontology statements as a building technique (Ceravolo et al., 2007; Demuth & Liebau, 2007; Karpovic & Nemuraite, 2011; Alberts & Franconi, 2012; Franconi & Mosca, 2012; Reynares, Caliusco, & Galli, 2013). The Semantics of Business Vocabulary and Business Rules (SBVR) supports that approach by providing business people with a linguistic way to semantically describe business concepts and specify business rules (Object Management Group (OMG), 2008) in an independent way of any information system design. SBVR has been conceptualized by OMG for business people and designed to be used for business purposes. The linguistic nature of the proposal enables the expression of business knowledge through statements rather than diagrams. That is rooted in the insight that diagrams are helpful for depicting structural organization of concepts but they are impractical as a primary means of defining vocabularies and expressing business rules. SBVR is rooted in first-order predicate logic with some restricted extensions into higher-order logics and some limited extensions into modal logic. Such sound theoretical foundation on formal logic is a key feature in automated reasoning contexts and presents a clear advantage over the use of UML/OCL models.

Experimental research comparing the approaches before mentioned is still lacking. Aim of this work is to evaluate the feasibility of mapping business domain expressions to ontology statements. Such feasibility study is performed by means of an exploratory experiment rooted in the quality assessment of a set of ontologies developed by applying the UML and SBVR based approaches. Ontologies were developed from the point of view of software engineering practitioners, i.e., they were developed by several groups of undergraduate engineering students.

The rest of this paper is organized as follows. Section 2 provides some conceptual foundations about ontologies. Section 3 depicts the experiment design. Section 4 and Section 5 present experiment results and the analysis of such results, respectively. Finally, discussions and future research directions are shown in Section 6 while conclusions are presented in Section 7.

2. Conceptual foundations: ontologies

Although several definitions of *ontology* can be found in literature, all of them share a common understanding: ontology is envisioned as a structure defining concepts used to represent knowledge and their relationships. Ontology model adopted by this paper enrols to such common understanding. It is independent of any ontology implementation language – although it can be mapped into most of them (Breitman & Leite, 2004) –, and it is defined as follows:

Definition 1. An ontology is a 5-uple $O := \{\mathcal{C}, \mathcal{R}, \mathcal{H}, \text{rel}, \mathcal{A}\}$ composed by:

- Two disjoint sets, $\mathcal{C}$ (concepts) and $\mathcal{R}$ (relations).
- A concept hierarchy, a directed relation $\mathcal{H} \subseteq \mathcal{C} \times \mathcal{C}$ which is called concept hierarchy or taxonomy. So, $\mathcal{H}(\mathcal{C}1, \mathcal{C}2)$ means $\mathcal{C}1$ is a sub-concept of $\mathcal{C}2$.
- A function rel: $\mathcal{R} \rightarrow \mathcal{C} \times \mathcal{C}$ that relates the concepts non taxonomically.
- A set of ontology axioms $\mathcal{A}$ expressed in appropriate logical language.

As depicted in next sections, quality assessment was performed over a set of OWL 2 ontologies. The OWL 2 Web Ontology Language (OWL 2) is the latest version of an ontology language proposed by the World Wide Web Consortium (W3C) for the development of the Semantic Web (World Wide Web Consortium (W3C), 2009b), but it has gradually evolved as a de facto standard for a broad spectrum of applications.

OWL 2 ontologies provide classes, properties, individuals, and data values, and are stored as Semantic Web documents. An OWL 2 ontology is a formal description of a domain of interest which allows useful inferences to be drawn by being rooted in three syntactic categories interpreted under a standardized semantics. Such semantic is compatible with the model theoretic semantics of the SROIQ Description Logic (DL), which is a fragment of first order logic with useful computational properties (Nardi et al., 2003). The three syntactic categories of OWL 2 can be quickly expressed in terms of the general definition presented above, and are depicted following:

- *Entities* such as classes, properties, and individuals. They are the basic elements of an ontology and are identified by Internationalized Resource Identifiers (IRIs) (Internet Engineering Task Force (IETF), 2005). For example, a class *a:Person* can be used to represent the set of all people, the object property *a:parentOf* can be used to represent the parent–child relationship and the individual *a:Peter* can be used to represent a particular person called "Peter".
- *Expressions*, representing complex notions in the domain being described. For example, a class expression describes a set of individuals in terms of the restrictions on the individuals characteristics.
- *Axioms*, which are statements asserted to be true in the domain being described. For example, a subclass axiom state that the class *a:Student* is a subclass of the class *a:Person*.

It can be noted that modelling of individuals – i.e., particular instances of given classes – is possible in developing of OWL 2 ontologies but it is not part of the theoretic definition before presented. However, such difference is overcome given that the instantiation of ontologies is not a part of the experiment.

3. Experiment design

This paper presents an exploratory experiment evaluating the feasibility of mapping business domain expressions to ontology statements. Exploratory research is conducted for a problem that has not been clearly defined, or when the aim of the study is to gain familiarity with the problem and to acquire a new insight. Exploratory research may use a variety of methods such as trial studies,

Author's personal copy

interviews, group discussions, experiments, or other tactics for the purpose of gaining information. Although this kind of research draws conclusions only with extreme caution given its fundamental nature, it helps researchers to have sufficient information to improve the planning of future experiments by determining the best research design, data collection methods, selection and prediction of the number of required subjects, and the expected power of the test in order to formulate further hypotheses to be tested. Exploratory studies are thus an important mechanism for generating hypotheses and guiding further research activities (Kitchenham et al., 2002).

3.1. Objective and hypothesis

The experiment approaches the feasibility of mapping business domain expressions to ontology statements. Feasibility is evaluated by testing the hypothesis:

Hypothesis 1. Ontologies developed by means of a SBVR based approach at least equate the quality of ontologies developed by using an UML based method.

Hypothesis testing is rooted in the quality assessment of the ontologies developed by several groups of undergraduate engineering students applying techniques based on UML and SBVR languages.

3.2. Context, experimental units and treatments

The experiment was performed in the context of a graduate degree course at the Argentinian Technological University in the province of Santa Fe. It was included in the last year of the course program of Information System Engineering and entitled "Ontology-based Informations Systems Development". Experimental units were 10 equally sized groups, randomly conformed by the 30 engineering students attending the course. By taking part in the experiment, participants earned educational credits.

The experiment involves two treatments:

1. Ontology development by applying the UPON method depicted in Nicola et al. (2009).
2. Ontology development by applying the SBVR mapping approach proposed in Reynares et al. (2013).

A random process restricted to the generation of equally sized samples was used for the allocation of experimental units to the treatments. In this way, 5 groups followed the UML approach while the remaining 5 groups applied the SBVR mapping approach.

UPON (Nicola et al., 2009) is an incremental ontology building methodology whose characteristics stem from the Unified Process (UP) (Jacobson, Booch, & Rumbaugh, 1999) and uses UML to support the preparation of all blueprints of the ontology development. The process starts by identifying and gathering the relevant domain terms in a lexicon, which is progressively enriched with definitions and yields a glossary. Populating it with the basic ontological relationships allows for producing a semantic network, until further enrichments and a final formalization produces the sought domain ontology.

Business rules mapping approach proposed in Reynares et al. (2013) is solely focused on the automatable generation of an OWL 2 ontology (World Wide Web Consortium (W3C), 2009a) by applying a set of transformations over the SBVR specifications of a business domain. As a consequence, experimental units applying this treatment had to identify business rules without any methodological guide previous to perform the mappings proposed.

Participants answered a survey about their previous knowledge in relation to the topics involved in the experiment. Answers were restricted to a numeric range from 1 – *no knowledge* – to 10 – *senior professional knowledge* – and it was a blind survey so as to grant the most honest results. Even so, the subjective bias introduced in the self evaluation of knowledge should be noted. Figures below show the frequency comparison between the answers obtained by each considered treatment. Frequency comparisons about knowledge of the use of UML artifacts (Q1), development of logical statements (Q2) and knowledge of the ontology development field (Q3) are shown in Figs. 1–3, respectively.[1]

Mann–Whitney–Wilcoxon (MWW) test was used to asses the differences between the answers in treatments. MWW test (also named *U statistic* or *U*) is a non-parametric test of the null hypothesis that two populations are the same against an alternative hypothesis – a particular population generally tends to have larger values than the other –. It has greater efficiency than the *t*-test on non-normal distributions and it is nearly as efficient as the *t*-test on normal distributions (Mann & Whitney, 1947; Fay & Proschan, 2010).

Table 1 shows the critical intervals of U for the two equally sized samples A and B (with size $n = 15$), for directional and non-directional test and for the most commonly used levels of significance. Null hypothesis must be accepted if the observed value of U is larger than the lower limit and smaller than the upper limit. Table 2 presents the observed value of U for both treatments in each above presented question. Observed values allow to conclude that there is no statistical significant difference between both treatments in knowledge about the use of UML artifacts (Q1), development of logical statements (Q2) and development of ontologies (Q3).

3.3. Task and material

The performed task was the ontological specification of the policies governing the student fellowship program of the university. Ontologies obtained in the experiment were implemented in the OWL 2 language. A natural language written document stating the policies governing the student fellowship program of the university was the main element for performing the task.[2] In relation to tools, both treatments made use of text editors to model the business domain according to the followed approach. Ontology implementation was performed by means of Protégé, a free and open source ontology editor.[3]

4. Results

Treatments performance was compared by assessing the quality of the developed ontologies. Quality evaluation task was performed by means of OQuaRE (Duque-Ramos, Fernández-Breis, Stevens, & Aussenac-Gilles, 2011; Duque-Ramos et al., 2013), a framework conceived for that purpose and based on the SQuaRE standard for software quality evaluation (International Organization for Standardization (ISO), 2005).

OQuaRE considers ontologies as artifacts obtained by means of a building process and evaluates them independently of any particular development process. It provides an automatable approach which enables the objective assessment of ontology quality and makes quality evaluation reproducible.

OQuaRe defines a quality model and quality metrics for ontology evaluation. Quality model is divided into a series of dimen-

[1] Full survey and results can be found in https://code.google.com/p/ontology-development-exploratory-experiment/.

[2] Institutional document depicting such policies can be found in goo.gl/3txhF9.

[3] Support, downloads and documentation can be found in http://protege.stanford.edu/.

Author's personal copy

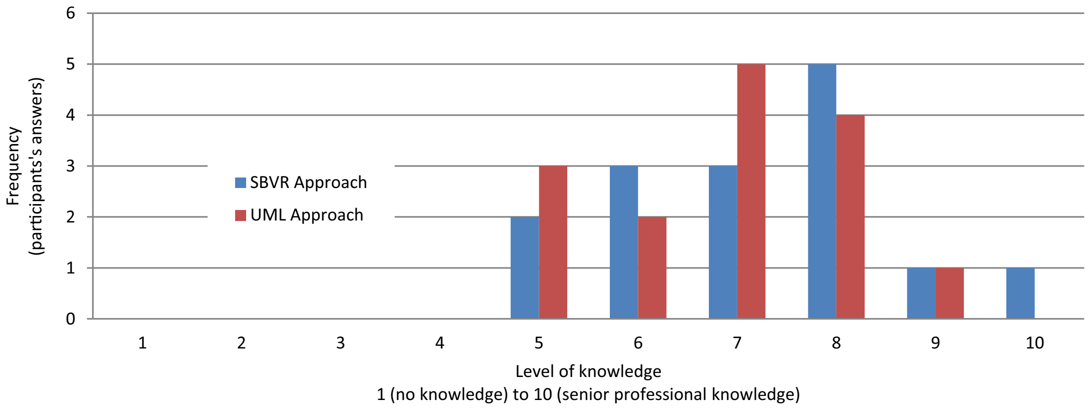


Fig. 1. Comparative frequency distribution of answers in treatments to Q1: "What's your level of knowledge of the use of UML artifacts?"

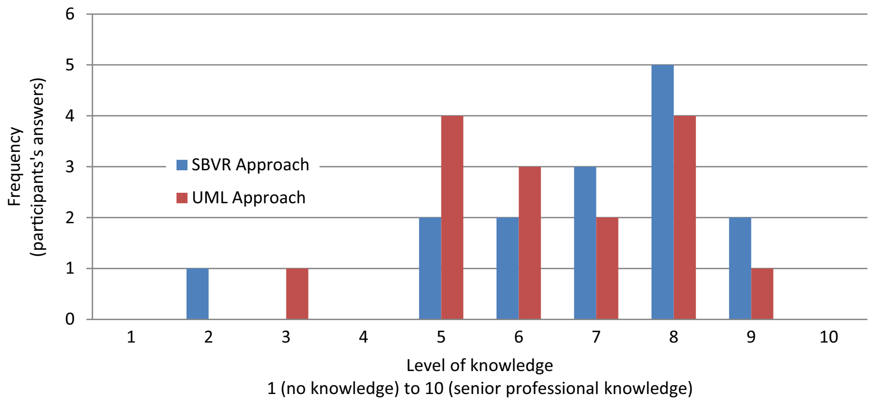


Fig. 2. Comparative frequency distribution of answers in treatments to Q2: "What's your level of knowledge of the development of logical statements?"

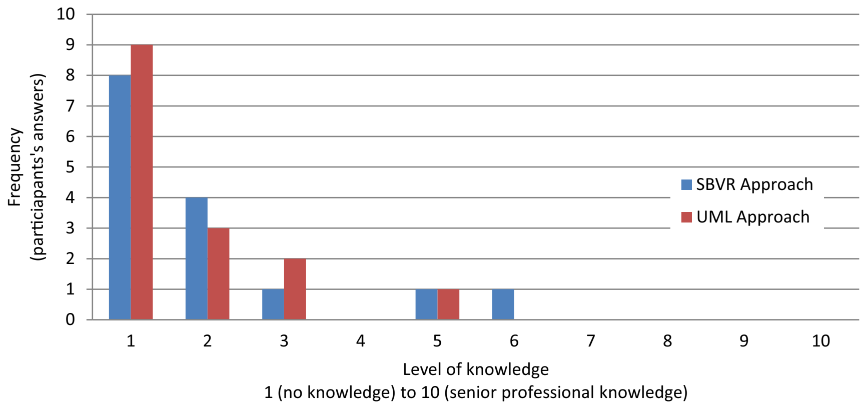


Fig. 3. Comparative frequency distribution of answers in treatments to Q3: "What's your level of knowledge of the ontology development field?"

Table 1
Critical intervals of U for the two equally sized samples A and B (with size $n = 15$).

Critical intervals of U			
	Level of significance for directional test		
	0.05	0.025	0.01
	Non-directional test		
	–	0.05	0.02
Lower Limit	72	64	56
Upper Limit	153	161	169

Table 2
Values of U for both treatments about Q1, Q2 and Q3.

	Values of U for	
	UML approach	SBVR approach
Q1: About the use of UML artifacts	127	98
Q2: About logical statements development	135	90
Q3: About the development of ontologies	120.5	104.5

Author's personal copy

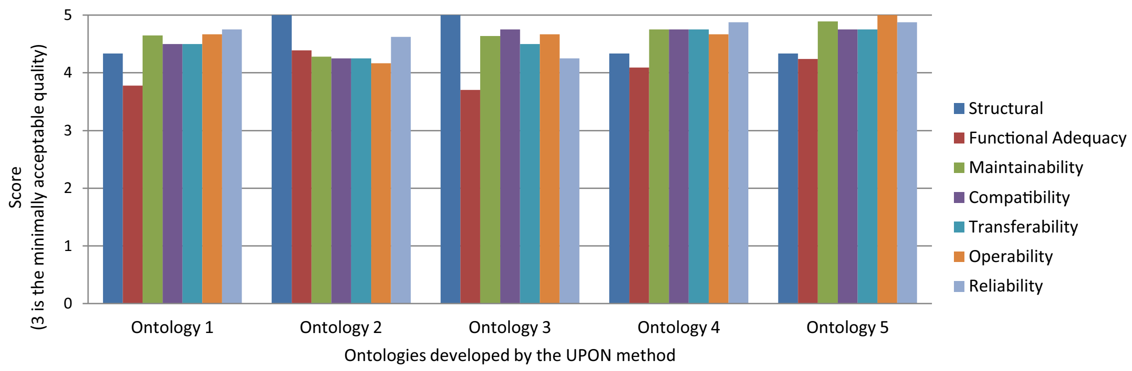


Fig. 4. Dimensions quality scores of the ontologies developed by the UML approach.

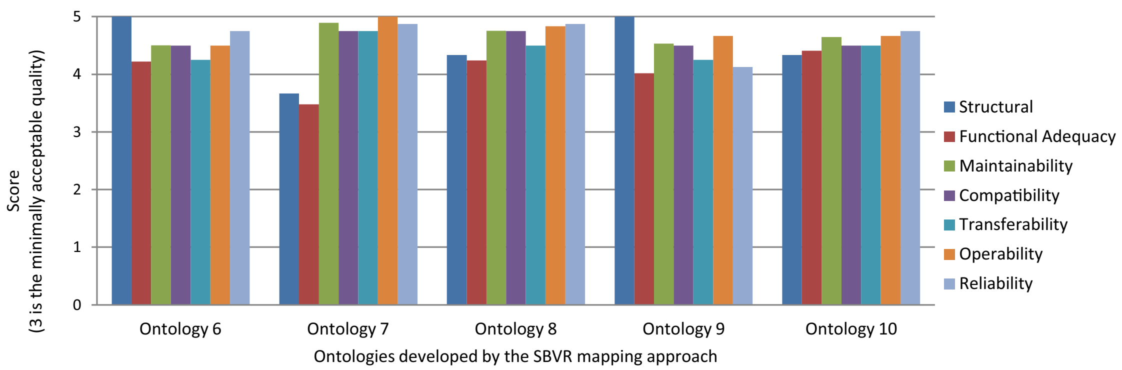


Fig. 5. Dimensions quality scores of the ontologies developed by the SBVR mapping approach.

sions – or characteristics – organized into subdimensions – or *subcharacteristics* – which are evaluated by applying a set of automatable metrics. OQuaRE defines the criteria to transform the quantitative scores of each metric into a 1–5 range and establishes that *1 means not acceptable, 3 is minimally acceptable and 5 exceeds the requirements*. After such transformation, score for each subcharacteristic is the mean of its associated metrics while the score of each characteristic is the mean of its subcharacteristics. The set of characteristics scores is the quality assessment result, enabling the identification of strengths and flaws of the ontologies rather than simply pointing out a "best ontology".

Fig. 4 shows the quality scores for the ontologies developed by means of the UML approach. Fig. 5 shows the quality scores for the ontologies developed by means of the SBVR mapping approach. Dimensions evaluated in the experiment are defined as follows:

- *Structural* dimension involves formal and semantic properties that are important when evaluating ontologies since it accounts for quality factors such as consistency, formalisation, redundancy or tangledness.
- *Functional adequacy* dimension refers to the appropriateness of the ontology for its intended purpose, according to the categories identified by Stevens, Wroe, Gobel, and Lord (2008).
- *Maintainability* dimension is related to the capability of the ontologies to be modified for changes in the environment, in requirements or in functional specifications.
- *Compatibility* dimension refers to the ability of two or more ontologies to exchange information and/or to perform their required functions while sharing the same hardware or software environments. The compatibility dimension can be evaluated over a single ontology – although intuitively it involves properties about more than one ontology – given that it is quantitatively assessed by means of a set of metrics applied to each ontology separately.
- *Transferability* dimension is the degree to which the ontology can be transferred from one environment (e.g., operating system) to another.
- *Operability* dimension refers to the effort needed to use the ontology and, in the individual assessment of such use, by a stated or implied set of users.
- *Reliability* dimension is the capability of the ontology to maintain its level of performance under stated conditions for a given period of time.

Three metrics were left out of consideration at the assessment of the aforementioned quality dimensions. *Annotation richness* – i.e., mean number of annotations per class – and *class richness* – i.e., mean number of instances per class – metrics were not assessed since the annotation and instantiation of ontologies were not a part of the task of the experiment. *Attribute richness* metric – i.e., mean number of attributes per class – was not assessed because "attributes" are not a part of the structural specification of the implementation language of the assessed ontologies (World Wide Web Consortium (W3C), 2009b).

OQuaRE also defines *performance efficiency* and *quality in use* dimensions. *Performance efficiency* exposes the relationship between the level of performance of the ontology and the amount of used resources, under stated conditions, taking into account elements such as time of response or memory consumption. *Quality in use* refers to the degree to which the ontology used by specific users meets their needs to achieve specific goals. However, such

Author's personal copy

Table 3
Mean values of each assessed quality dimension by followed approach.

	UML approach	SBVR approach
Structural	4.60	4.47
Functional Adequacy	4.05	4.05
Maintainability	4.64	4.67
Compatibility	4.60	4.60
Transferability	4.55	4.45
Operability	4.63	4.73
Reliability	4.68	4.68
Global Mean	4.54	4.52

Table 4
Critical intervals of U for the two equally sized samples A and B (with size $n = 5$).

Critical intervals of U			
	Level of significance for directional test		
	0.05	0.025	0.01
	Non-directional test		
	–	0.05	0.02
Lower Limit	4	2	1
Upper Limit	21	23	24

Table 5
Values of U by quality dimension and by followed approach.

	Values of U for	
	UML approach	SBVR approach
Structural	11	14
Functional Adequacy	13.5	11.5
Maintainability	12.5	12.5
Compatibility	11.5	13.5
Transferability	9	16
Operability	14.5	10.5
Reliability	13	12

dimensions were left out of consideration because there was a lack of metrics for their subcharacteristics.[4]

5. Result analysis

A first result to be highlighted is the level of quality shown by the ontologies: according to the meaning assigned for OQuaRE to the values of the 1–5 ranking system, all ontologies largely outperform the minimally acceptable quality in all considered dimensions.

The mean value of each assessed quality dimension – shown in Table 3 – also provides important insights. First column of the table shows the mean value of each quality dimension over the ontologies developed by following the UML approach. Second column shows such values over the ontologies developed by means of the SBVR approach. By comparing row by row, it can be observed that the quality dimensions means are almost identical between the followed approaches. Last row of the table shows the global mean score of ontology quality – calculated as the mean of all the quality scores – according the followed approach. Again, such values are not only almost identical, but also are very close to the maximal dimension quality score.

Besides the numerical analysis before presented, MWW test was used to asses the differences between the quality scores of the ontologies produced by both treatments. Table 4 shows the critical intervals of U for the two equally sized samples A and B (with size $n = 5$), for directional and non-directional test and for the most commonly used levels of significance. Table 5 presents the observed value of U for both treatments in each assessed quality dimension. Observed values allow to conclude that there is no statistical significant difference between both treatments in any of the assessed quality dimensions of the ontologies. These test results highlight the feasibility of mapping business domain expressions to ontology statements by supporting the hypothesis of the experiment: ontologies developed by means of a SBVR based approach at least equate the quality of ontologies developed by using an UML based method.

Table 6
Description Logic expressivity by ontology and by followed approach.

DL Expressivity		
UML approach	Ontology 1	$\mathcal{ALCHI(D)}$
	Ontology 2	$\mathcal{AL(D)}$
	Ontology 3	$\mathcal{ALF(D)}$
	Ontology 4	$\mathcal{ALE(D)}$
	Ontology 5	$\mathcal{ALEHF(D)}$
SBVR approach	Ontology 6	$\mathcal{ALCHQ(D)}$
	Ontology 7	$\mathcal{ALCOQ(D)}$
	Ontology 8	$\mathcal{ALCHOQ(D)}$
	Ontology 9	$\mathcal{ALCIQ(D)}$
	Ontology 10	$\mathcal{ALEF(D)}$

6. Discussion and future work

The experiment presented in this work was biased against the SBVR approach in two ways. First, participants had previous knowledge related to the use of UML artifacts while knowledge about SBVR language was inexistent. Second, UML based method provided methodological guides for each stage of ontology building process while SBVR based approach just specify a set of mappings from logical statements to OWL 2 expressions.

Results obtained by the experiment confirm previous research in the field, i.e., UML approaches are highly effective as a mean for conceptualizing lightweight ontologies (Gómez-Pérez et al., 2004). Although the SBVR to OWL 2 mapping technique performed in the experiment also present such a feature, it is interesting to stress the potential of the SBVR language to express complex notions of a domain of interest. A hint revealing such potential is found in the ontologies generated by the SBVR approach: they made use of the full expressive power of the OWL 2 language while their counterparts – i.e., ontologies generated by the UML approach – involve a reduced subset of such expressive power.[5] Table 6 shows the expressive power of the developed ontologies while Table 7 depicts the Description Logic constructors and language names (Nardi et al., 2003). Although the expressive power of UML can be complemented by means of the OCL language, advantage of SBVR is given by its sound theoretical foundation on formal logic.

It is necessary to mention that the developed ontologies were examined by a domain expert before performing the quality assessment task depicted in Section 4. That examination was aimed at providing some feedback about the relation between the reality being modelled and the ontological representation of such a reality. Although each ontology showed distinctive characteristics, all of them were – according to the subjective assessment of the domain expert – right representations of the domain of interest.

This experiment is an initial attempt to asses the feasibility of SBVR based approaches for ontology development, given that –

[4] A full description of the applied quality model and the obtained results can be found in https://code.google.com/p/ontology-development-exploratory-experiment/.

[5] A full description of the obtained results can be found in https://code.google.com/p/ontology-development-exploratory-experiment/.

Author's personal copy

Table 7
Description Logic constructors and language names: *A* refers to atomic concepts, *C* and *D* to any concept definition, *R* to atomic roles and *S* to role definitions.

Construct	Syntax	Language
Concept	A	
Role name	R	
Intersection	C ∩ D	$\mathcal{FL}_0$
Value restriction	∀ R.C	$\mathcal{FL}^-$ $\mathcal{AL}$
Limited existential quantification	∃ R	$\mathcal{S}$
Top or Universal	⊤	
Bottom	⊥	
Atomic negation	¬ A	
Negation	¬ C	$\mathcal{C}$
Union	C ∪ D	$\mathcal{U}$
Existential restriction	∃ R.C	$\mathcal{E}$
Number restrictions	$(\geq nR)$ $(\leq nR)$	$\mathcal{N}$
Nominals	$\{a_1...a_n\}$	$\mathcal{O}$
Role hierarchy	$(R \subseteq S)$	$\mathcal{H}$
Inverse role	R^-	$\mathcal{I}$
Quantified number restriction	$(\geq nR.C)$ $(\leq nR.C)$	$\mathcal{Q}$

up to the knowledge of the authors - there is no empirical research in the field. Although any laboratory experiment suffers from a certain lack of realism, a field study is quite difficult to be designed and executed. It should be also noted that generalizations of results is limited due to the exploratory nature of the experiment. Even with such limitations, experiment results allow the authors to formulate further hypotheses to be tested in subsequent experiments. According to that, future work involves the performing of a controlled experiment which will allow for testing more complex hypotheses regarding the development of ontologies by means of the SBVR approach. For example, it would be interesting to assess the approach performance – in terms of ontology quality, time and effort – in comparison with the Ontology Definition Metamodel (ODM) proposal (Object Management Group (OMG), 2009). Moreover, such experiments will adopt a more complex and systematic way of assessing the developed ontologies. First, the assessment of the semantic gap between reality and the developed ontologies by means of a systematic and quantitative process would eliminate the subjective bias of the domain expert. Finally, assessment and customization of the OQuaRE evaluation framework could provide more useful insights according to the particular nature of the ontologies developed by this kind of techniques.

7. Conclusions

Discussions on integrating Software and Ontology Engineering approaches tends to be academic, neglecting important issues such as applicability and providing little guidance for software engineers. A core requirement for the use of ontologies by software engineers in information systems is the availability of proved and tested techniques which guarantee efficient engineering of high-quality ontologies.

An exploratory experiment comparing performance of undergraduate engineering students applying UML and SBVR based approaches for ontology development has been presented in this paper. Using of MWW test has enabled to assert that there is no statistically significant differences between both treatments in knowledge about the main topics involved in the experiment, allowing to leave out of consideration knowledge issues on the observed ontology quality scores.

Performance of the approaches has been measured by assessing several quality dimensions of the ontologies according to OQuaRE, an automatable framework based on the SQuaRE standard for software quality evaluation which enables objective assessment and makes quality evaluation reproducible. MWW test was also used to asses the differences between the quality scores of the ontologies produced by both treatments, concluding that there were no statistically significant differences between both treatments in any of the quality dimensions. Results obtained by such quality comparison test highlight the feasibility of mapping business domain expressions to ontology statements by supporting the hypothesis of the experiment: ontologies developed by means of a SBVR based approach at least equate the quality of ontologies developed by using an UML based method. Moreover, the results obtained by the experiment confirm previous research about the effectiveness of UML approaches for conceptualizing lightweight ontologies while stressing the potential of the SBVR language to express complex notions of a domain of interest. These findings highlight the potential of SBVR to OWL 2 mappings as an ontology development technique worthy of further study.

Acknowledgements

The authors are grateful to 'Universidad Tecnológica Nacional (UTN)', 'Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)', and 'Agencia Nacional de Promoción Científica y Tecnológica (ANPCyT)' for their financial support.

References

Alberts, R., & Franconi, E. (2012). An integrated method using conceptual modelling to generate an ontology-based query mechanism. In P. Klinov, & M. Horridge (Eds.), *OWL: Experiences and Directions Workshop 2012 (OWLED). Volume 849 of CEUR Workshop Proceedings*. Greece.
Beydoun, G., Low, G., Tran, N., & Bogg, P. (2011). Development of a peer-to-peer information sharing system using ontologies. *Expert Systems with Applications, 38*, 9352–9364.
Breitman, K., & Leite, J. (2004). Lexicon based ontology construction. In C. Lucena, A. Garcia, A. Romanovsky, J. Castro & P. S. C. Alencar (Eds.), Software engineering for multi-agent systems II LNCS 2940 (pp. 41–45). Springer, Berlin Heidelberg.
Calero, C., Ruiz, F., & Piattini, M. (2006). *Ontologies for software engineering and software technology*. Heidelberg: Springer.
Ceravolo, P., Fugazza, C., & Leida, M. (2007). Modeling semantics of business rules. In *Ecosystems, digital technologies conference. DEST 07* (pp. 171–176).
Chen, R.-C., Huang, Y.-H., Bau, C.-T., & Chen, S.-M. (2012). A recommendation system based on domain ontology and SWRL for anti-diabetic drugs selection. *Expert Systems with Applications, 39*, 3995–4006.
Demuth, B., & Liebau, H.-B. (2007). An approach for bridging the gap between business rules and the semantic web. *Proceedings of the International Symposium on Advances in Rule Interchange and Applications*. In A. Paschke & Y. Biletskiy (Eds.). *Lecture Notes in Computer Science* (vol. 4824). Berlin: Springer.
Duque-Ramos, A., Fernández-Breis, J. T., Iniesta, M., Dumontier, M., Aranguren, M. E., Schulz, S., Aussenac-Gilles, N., & Stevens, R. (2013). Evaluation of the OQuaRE framework for ontology quality. *Expert Systems with Applications, 40*, 2696–2703.
Duque-Ramos, A., Fernández-Breis, J. T., Stevens, R., & Aussenac-Gilles, N. (2011). OQuaRE: A SQuaRE-based approach for evaluating the quality of ontologies. *Journal of Research and Practice in Information Technology, 43*, 159–176.
Fay, M., & Proschan, M. (2010). Wilcoxon–Mann–Whitney or t-test? On assumptions for hypothesis tests and multiple interpretations of decision rules. *Statistics Surveys, 4*, 1–39.
Franconi, E., & Mosca, A. (2012). The formalisation of ORM2 and its encoding in OWL2. Technical Report KRDB Research Centre Technical Report KRDB12-2, Faculty of Computer Science, Free University of Bozen-Bolzano Italy.
Gómez-Pérez, A., Fernández-López, M., & Corcho, O. (2004). *Ontological engineering*. Heidelberg: Springer.
Guizzardi, G., Herre, H., & Wagner, G. (2002). Towards ontological foundations for UML conceptual models. *On the Move to Meaningful Internet Systems, 2002 – DOA/CoopIS/ODBASE 2002 Confederated International Conferences DOA, CoopIS and ODBASE 2002* (pp. 1100–1117). London, UK: Springer-Verlag.
International Organization for Standardization (ISO) (2005). ISO/IEC 25000:2005, Software Engineering – Software Product Quality Requirements and Evaluation (SQuaRE).
Internet Engineering Task Force (IETF) (2005). RFC 3987: Internationalized Resource Identifiers (IRIs).
Jacobson, I., Booch, G., & Rumbaugh, J. (1999). *The unified software development process*. Massachusetts, USA: Addison-Wesley.

Author's personal copy

Karpovic, J., & Nemuraite, L. (2011). Transforming SBVR business semantics into Web Ontology Language OWL 2: Main concepts. In *Proceedings of 17th international conference on information and software technologies (IT 2011)*.
Kitchenham, B., Pfleeger, S., Pickard, L., Jones, P., Hoaglin, D., El Emam, K., & Rosenberg, J. (2002). Preliminary guidelines for empirical research in software engineering. *IEEE Transactions on Software Engineering, 28*, 721–734.
Mann, H. B., & Whitney, D. R. (1947). On a test of whether one of two random variables is stochastically larger than the other. *Annals of Mathematical Statistics, 18*, 50–60.
Myrgioti, E., Bassiliades, N., & Miliou, A. (2013). Bridging the HASM: An OWL ontology for modeling the information pathways in haptic interfaces software. *Expert Systems with Applications, 40*, 1358–1371.
Nardi, D., Brachman, R. J., Baader, F., Nutt, W., Donini, F. M., Sattler, U., Calvanese, D., Mlitor, R., Giacomo, G. D., Ksters, R., Wolter, F., McGuinness, D. L., Patel-Schneider, P. F., Mller, R., Haarslev, V., Horrocks, I., Borgida, A., Welty, C., Rector, A., Franconi, E., Lenzerini, M., & Rosati, R. (2003). *The Description Logic Handbook – Theory, Implementation and Applications*. Cambridge University Press.
Nicola, A. D., Missikoff, M., & Navigli, R. (2009). A software engineering approach to ontology building. *Information Systems, 34*, 258–275.
Object Management Group (OMG) (2008). Semantics of Business Vocabulary and Business Rules (SBVR). Version 1.0: Formal Specification.
Object Management Group (OMG) (2009). Ontology Definition Metamodel (ODM). Version 1.0.
Object Management Group (OMG) (2011). Unified Modeling Language (UML). Version 2.4.1.
Object Management Group (OMG) (2012). Object Constraint Language (OCL). Version 2.3.1.
Reynares, E., Caliusco, M. L., & Galli, M. R. (2012). EDON: A method for building an ontology as software artefact. In *41st Argentine Conference on Informatics – 13th Argentine symposium on software engineering (JAIIO – ASSE 2012)*, La Plata, Buenos Aires, Argentina.
Reynares, E., Caliusco, M. L., & Galli, M. R. (2013). An automatable approach for SBVR to OWL 2 mappings. In *XVI Ibero-American Conference on Software Engineering (CIbSE 2013)*. Montevideo, Uruguay.
Ruotsal, T. (2010). Methods and applications for ontology-based recommender systems (Ph.D. Thesis). Ph.D. thesis Alto University School of Science and Technology Finland.
Shue, L.-Y., Chen, C.-W., & Shiue, W. (2009). The development of an ontology-based expert system for corporate financial rating. *Expert Systems with Applications, 36*, 2130–2142.
Stevens, R., Wroe, C., Gobel, C., & Lord, P. (2008). Application of ontologies in bioinformatics. In S. Staab & R. Studer (Eds.). *Handbook of Ontologies in Informations Systems* (pp. 635–658). Springer.
Wang, X., & Chan, C. (2001). *Ontology modeling using UML*. Canada: Calgary.
World Wide Web Consortium (W3C) (2009a). OWL 2 web ontology language. Document overview.
World Wide Web Consortium (W3C) (2009b). OWL 2 web ontology language. Structural specification and functional-style syntax.