Reconciling the Expectations of Ontology Engineering to the Process of Requirements Elicitation

Artur Machura[0000-0002-6175-8507]

University of Economics, Katowice, Poland
artur.machura@uekat.pl

Abstract The paper is devoted to research into methods for requirements elicitation. From our perspective, the use of ontology engineering is particularly interesting. The author focuses his attention on selected aspects of ontology engineering, resulting from the needs of his doctoral thesis, organized by the Design Science Research paradigm. The article seeks to recognize a certain level of interplay between ontology engineering and individual tasks of the process of requirements elicitation (a synergy). Therefore, it aims to explain possible expectations for ontology engineering, which result from the specific features of the process of requirements elicitation. In order to identify the guidelines for the developing domain ontologies based on the requirements elicitation process. The originality of the article results from the consideration of the interplay between ontology engineering and requirements engineering, relative to the context of software engineering. As a result, the software engineering layers propagated by Pressman have been extended to include another layer of management philosophy. The rationale behind this was the noticing of the impact of other concepts of philosophy on this work, such as axiology or epistemology.

Keywords: Ontology, Requirements elicitation, Ontology-driven requirement engineering

1 The Problem Addressed by the PhD Thesis

The PhD thesis "The production of domain ontologies to elicit the requirements of IT systems" tries to meet the needs of those IT system projects in which there is an identified problem of missing, incomplete and changing requirements. Solutions to this well-known problem of requirements engineering are sought in the use of the ontology engineering. The title of the thesis concerns concepts such as: domain ontologies or elicitation of requirements. In the work, an attention is mainly paid to knowledge contained in domain ontology. At the same time, the scope of this scientific work was limited by the research hypothesis entitled "The process of developing ontologies of domain-based business-IT alignment reduces the problem of missing, incomplete and changing requirements for information systems". It is essential for the work to set such specifications and properties of the process of developing ontologies of the domain 'business-IT alignment', which will provide the basis for reducing the problem of

missing, incomplete and changing requirements. It also means that it is important, from the perspective of the PhD thesis, to agree on the expectations for the process of creating domain ontologies. It consists of the following issues:

- what expectations for elicitation of requirements, does the use of ontology engineering bring?
- how does the elicitation of requirements influence the use of the ontology engineering workshop?
- what is the significance of the specificity of domain ontologies (various conceptual categories) for the production of ontologies?

Despite many possible issues of interest to the doctoral thesis, the paper attempts to define only the influence of ontology engineering on requirements engineering, to elicit requirements based on domain ontologies. The review of the literature devoted to the issue of the use of ontologies could be much larger if the scope of literature was not narrowed down by using the Population, Intervention, Comparison, Outcomes, Context (PICOC) conceptual framework [1]. As a result, the number of significant literature items has decreased, i.e. from several thousand to several hundred items. The paper only refers to selected publications related to the purpose of the paper.

2 Introduction and Purpose of the Paper

The purpose of the paper is the study on ontology engineering guidelines for elicitation the requirements for IT systems. In general, it has been arranged according to the classical hierarchy of software engineering issues [2]. Due to the inclusion of ontology engineering in the software production process. However, to explain the essential axiological context, close to the concept of ontology, a layer of management philosophy was added to the original version of the graphical representation of the layers (see Fig. 1). The Fig. 1 shows that groups of software engineering issues interact with each other. Therefore, the user of software engineering should understand the relationship of concepts contained in individual groups of software engineering.

Fig. 1. The pyramid of software engineering issue.

The next points of this paper have been organized by the pyramid of software engineering issue, presented on the Fig.1. The research summary has also been organized in accordance with the layers of software engineering presented in Fig. 1.

3 Research methods

The work described in this article is subordinated to the guidelines of the research paradigm of information systems in the science of creating artifacts – DSR (Design Science Research)[3]. Hevner and Chaterjee proposed DSR to solve technical and construction problems. DSR is recognized in scientific research, where the creation and evaluation of technical artifacts takes place, including artifacts useful for solving identified problems of economic organizations[4]. DSR is also as main research guide in the PhD thesis of the author of the paper. It means, among other things, that the results reported in this paper will be used in the thesis. Hevner and Chaterjee proposed a sequence of steps that allow for the implementation of activities according to the DSR paradigm: 1) Identifying the problem and motivation; 2) Definition of the study objectives; 3) Design and development; 4) Demonstration; 5) Evaluation; 6) Communication for the diffusion of knowledge in the research community.

In connection with the sequence of DSR steps and requirements of step 2, i.e. the definition of the objectives of the study, the author of the paper intends to achieve the following goals:

- Characterize the research goal required for the third research step, i.e. Design and development.
- To restrict the literature of the subject, supporting the achievement of the research objective.

Achievement of goals will be possible by choosing specific issues from the pyramid of software engineering, presented in Fig. 1.

4 Related works

4.1 Management philosophy

In order to characterize management philosophy, the author adopted the following approach to concepts such as: axiology, epistemology, ontology. Fig. 2 presents cooperative issues of management philosophy. It assumed that the definition of axiological criteria allows adopting the epistemological standpoint of knowing the organization only in the next step. In turn, it enables the ontology production. At the same time, it is managed by methodology, e.g. the one for requirements elicitation.

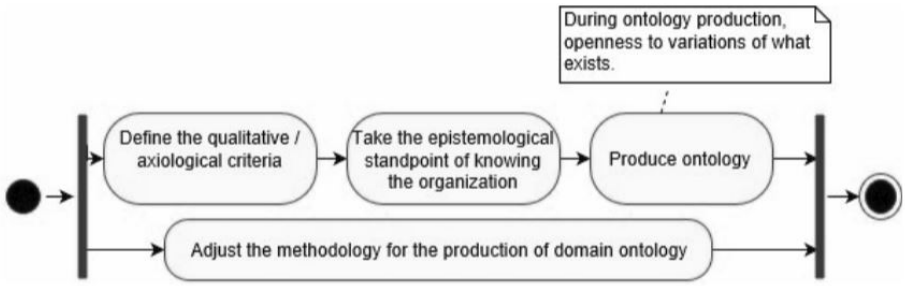

Fig. 2. Control flow between philosophical concepts. Source: Own study

The paper is based on axiological criteria resulting from ISO / IEC 25010: 2011[5]. The epistemological standpoint, due to the so-called strong sociology program [6] gets to know the organization through dialogue with its representatives. Ontology avoids a top-down assumption of what it can represent and remains open to any variant of it through the creation of e.g. organizational strategies. Different cooperative methods are formed during the consideration of each of the issues of management philosophy and are influenced by the position towards all listed aspects: axiology, epistemology, ontology.

4.2 Caring for quality

The ISO/IEC 25010: 2011 standard currently summarizes the quality attributes as well as properties for IT systems and requirements for them. Thus, this standard affects the perception of quality at the international level. The Table 1 summarizes these attributes, and the properties of these attributes.

Table 1. Quality attributes according to the ISO/IEC 25010:2011 standard. Source: [7]

Quality attributes	**Attribute properties**
Functional suitability	Functional completeness, Functional correctness, Functional appropriateness
Performance efficiency	Time behaviour, Resource utilisation, Capacity
Compatibility	Co-existence, Interoperability
Usability	Appropriateness recognisability, Learnability, Operability, User error protection, User interface aesthetics, Accessibility
Reliability	Maturity, Availability, Fault tolerance, Recoverability
Security	Confidentiality, Integrity, Non-repudiation,

	Accountability, Authenticity
Maintainability	Modularity, Reusability, Analysability, Modifiability, Testability
Portability	Adaptability, Installability, Replaceability

Considerations subordinated to the purpose of the article, based on the philosophy of management, pay attention to future system users. The so-called strong sociology recognizes society's knowledge as crucial. As for the aforementioned quality attributes, it should lead to reconciliation of all attributes these attributes with future users.

4.3 Process

The relationship between ontology engineering and software production was agreed in the context of the ODSD process (Ontology Driven Software Development)[8]. In this context, ontologies through technologies offered by ontology engineering, extend the possibilities of a typical MBSD (Model Based Software Development) approach[9]. It closely cooperates with the MDA (Model Driven Architecture)[10], which lists CIM (Computation Independent Model), PIM (Platform Independent Model), PSM (Platform Specific Model) models. The definition below is particularly important:

"CIM is a model independent of computation / computational processing, which is also identified with a business model or field, where the vocabulary of domain experts is included. The model shows what the system will be used for, but hides all technological information related to its further implementation" [11]

The considerations of this paper essentially oscillate around the CIM model. Requirements elicitation, de facto from users representing a certain organization, should first of all allow the responsibility of the CIM model to be met. It is related to the fact that, according to the best practices agreed in ODSD, the decisions of subsequent models depend on CIM model. The paper, focuses only on the elicitation of requirements.
The diagram in Fig 3 shows the flow of control between the activities concerned by this paper, such as: Creation and development of the domain ontologies, Implementation of IT project, and Exploitation of an IT solution.

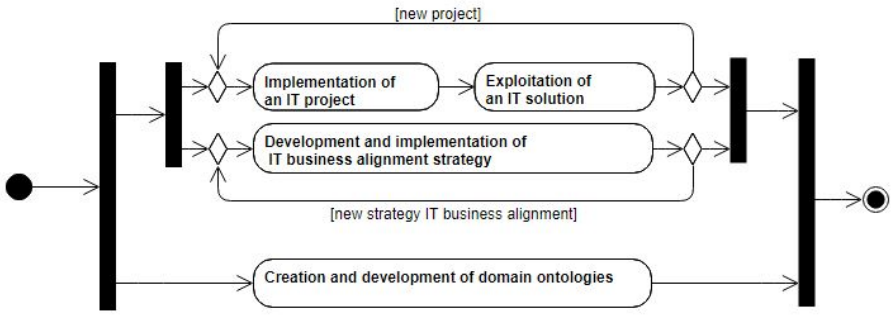

Fig. 3. Control flow between the envisioned activities: Creation and development of domain ontologies, Development and implementation of IT business alignment strategy, Implementation of IT project, Exploitation of an IT solution. Source: Own study

The considerations of this article pay attention primarily to the fact that these tasks are performed simultaneously. From what it can be concluded that the software development process could be open to accepting design artifacts coming from the IT business alignment strategy. And thus elicitation of requirements based on domain ontologies plays an important role here.

4.4 Methods

While discussing methods related to requirements engineering, attention is paid to the relationship of this work with system engineering and software design. In Pressman's book [2] excellent indication is located that the analysis of requirements depends on both system engineering and software design. In other words, we must not forget, during the analysis of the requirements, of both system engineering and software design. This basic specificity of software engineering methods can be referred to the modern ODSD process itself. From an extensive publication devoted to ODSD[8], one can conclude about the interplay between requirements engineering and ontology.

"*For requirements engineering, we suggest using a domain-independent ontology to determine the general concepts, requirements and dependencies to be detected when disclosing requirements."*

However, without diminishing the role of ontology to only defining the guidelines for the disclosure of requirements, attention here is paid to the definition that somehow opens the possible scope of ontology. "Ontology-driven (or ontology-based) RE form a RE process or at least a RE method comprehensively aided by ontologies. Therefore, ontologies are involved in the RE process. ODRE clearly states the method to integrate a proposed process on the continuous process RE" [12] The reconciliation of the ontology engineering guidelines in the ODRE architecture to the process of disclosing requirements draws attention to those ontology methods that support the following tasks [13]:

1. understanding the field of application,

2. identification of sources of requirements,
3. analysing the stakeholder,
4. selecting the techniques, approaches and tools;
5. eliciting the requirements.

However, attention should be paid to the popular definition of ontology "An ontology is a formal, explicit specification of a shared conceptualization". This definition was influenced by the work of the authors: Gruber [14], Borst [15], and Studer [16]. Indeed, the tasks of the process of requirements elicitation should meet the expectations of ontology engineering. Table 2 provides definitions of individual requirements elicitation tasks. The definitions in Table 2 specify the responsibility of the ontology at each stage of the requirements elicitation process.

Table 2. Theoretical responsibility for tasks of requirements elicitation. Source: Own study

Requirements elicitation process	**Definitions that determine the responsibility of ontology**
Understanding the field of application *(synonym of domain* [17]*)*	*Domain is an isolated context, with its own virtual address space, in which an application runs(...)* [18]
Identification of sources of requirements	*Requirements have many sources in typical software, and it is essential that all potential sources be identified and evaluated. This topic is designed to promote awareness of the various sources of software requirements and of the frameworks for managing them.* [19]
Analysing the stakeholder	*Analysing the stakeholderes - This step permitted to refine and keep only relevant stakeholders. Stakeholders did not have the same importance or would not be affected in the same way by the construction project.* [13]
Selecting the techniques, approaches and tools	*Selecting the techniques, approaches and tools - This step was often considered as a critical factor for this process success.* [13]
Eliciting the requirements	*Eliciting the requirements - All the information required is gathered at this step.* [13]

4.5 Tools

At the outset, it is noted that the process of elicitation requirements where the focus is placed on stakeholders and representatives of a certain organization, can generate any design artifacts. Based on the literature it is noted, that only a specific ontology categories are part of the analysis and specification of requirements [20] e.g.: application domain [21], application domain feature model [22], system behavior ontologies [23]. Meanwhile in software production many other categories of ontologies are distinguished, i.e. upper ontology [24], software process ontology [25], requirements ontology [26], software architecture ontology [27], and many others.
The point of view adopted in the article distinguishes between methods and accompanying tools. In the subject literature, one can identify those ontologies whose

application is top-down (e.g. OntoREM, an ontology-driven requirements engineering methodology) [28] and trying to create a kind of matrix for unidentified problems of what the ontology will present (e.g. ODM, an ontology definition meta model) [29]. The issue of marketization and industrialization of tools supporting the ontology engineering methods is also noteworthy. What is important in is whether the tooling environment provides the possibility of using, e.g. the traceability mechanism between ontology and the CIM model, as well as the wider process and even the life cycle of the software.

In the review of approaches in the use of ODSD [9], whose work oscillated around the so-called OMG meta-pyramid models, specified the following solutions:

1. The MOST project. The main goal of MOST is to enhance UML based syntactic modeling (structural modeling) by using OWL ontologies for the representation of static semantics of software systems. This is done by providing transformations from MDA models to OWL and integration these two technical spaces in of software development process.
2. A hybrid ODSD framework [30]. It uses SPARQL patterns in addition to OWL (and DL) for ontological modeling.
3. The DSL Meta-Model Ontology Based Approach. OWL ontologies are integrated in the model-based software technology that uses that uses automated program synthesis for generating software from models [31].
4. The Three ontology method [32] that uses domain, task and top-level ontologies for ontological modeling of a software system.

In addition, the same work has reached the very sources of the ODSD idea and it has been explained that ontologies as semantic declarative models can extend MDA. Starting in 2000, when W3C introduced Ontology Driven Architecture (ODA), this contributed to the introduction by the OMG Ontology Definition Metamodel (ODM), whose latest version is for the year 2014.

5 Research Results

The purpose of the work, through the analysis of the literature based on the DSR research paradigm, is illustrated in Fig. 4. It can be concluded that revealing requirements on the basis of ontologies is primarily a reconciliation of mutual meta-models between the source ontology and the target one containing requirements.

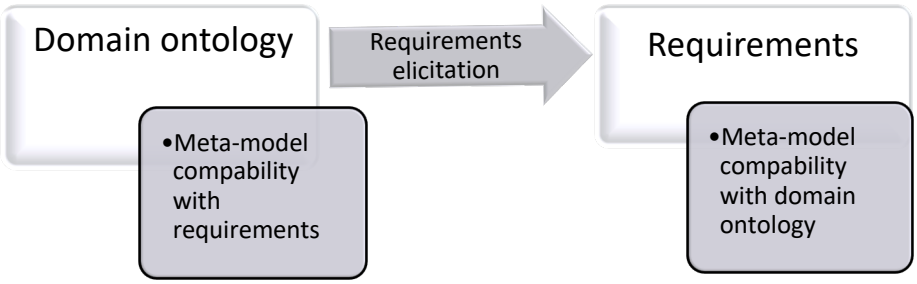

Fig. 4. Reference model for the elicitation of requirements based on ontology. Source: own study

The literature on the subject describes specific solutions relevant to specific fields of application (e.g. OntoREM), as well as those open to various variants of the application of ontologies (e.g. ODM). An objective choice between dedicated and universal solutions could be problematic, if it were not the software development process and subprocesses. The methods are directly dependent on this process (ODSD was selected in this paper). On the other hand, the classic layered approach to software engineering issues allows to look comprehensively to justify the choice. Nevertheless, the analysis of the literature allowed to formulate the conclusions contained in Table 3.

Table 3. The guidelines for requirements elicitation using ontologies. Source: Own study

The layers of the pyramid from Fig. 1	Elicitation of requirements with the use of ontologies
Tools	The multiplicity of possible categories of ontologies indicates the reconciliation of meta-models between the source ontology and the target ontology. Therefore, a solution was chosen that would allow defining both ontologies and meta-models. The ODM (Ontology Definition Meta Model) standard seems to meet these expectations. An example of analogous application is the MOST Project (Marrying Ontology and Software Technology). The software that supports the entire software production process is important here, which allows to use the functionality of, among others, traceability.
Methods	Methods of elicitation of requirements should be based on restrictive features of ontologies according to Gruber definition (conceptualization, formality, explicity), also ensuring compliance with the environment adequate to the selected development process.
The development process	The ODSD process specifies the use of the CIM model. With this model, CIM cooperates under the business analysis process, i.e. requirements elicitation, where tasks are specified: 1) Understanding the application field; 2) Identification of sources of requirements; 3)

	Shareholder analysis; 4) Selection of techniques, methods, tools; 5) Task to requirements elicitation.
Caring for quality	Agreeing with project stakeholders about the quality attributes: functional suitability, performance, compliance, usability, reliability, protection, ease of maintenance, portability.
Management philosophy	The context of management philosophy, suggests: axiological criteria based on the ISO / IEC 25010: 2011 standard, epistemological stand: strong sociology program, open ontology for any area of application, methods as flexible as possible for changes and ad-hoc expectations.

6 Summary

The paper tried to explain the expectations of ontology engineering with respect to the process and tasks devoted to requirements elicitation. As a result, it provides knowledge used in the PhD thesis devoted to the production of domain ontologies. Based on a review of direct literature and other similar reviews, it can be concluded that there are no uniform recommendations. At the same time, it determines the adoption of certain criteria, based on this paper in the position towards management philosophy. The whole of the issues raised in the paper has been arranged according to the classic layers of software engineering issues. As a result, it was assumed that aligning ontology engineering expectations for the process of elicitation requirements should be based on the position on these indissoluble issues:

- Reconciliation of stakeholders' expectations;
- Adoption of the position on quality attributes;
- Consolidation of the development process with the business analysis itself;
- Subordination of work methods to the required ontology features;
- Ensuring compatibility between meta-models of domain ontologies and requirements.

7 References

1. Booth, A., Sutton, A., Papaioannou, D.: Systematic approaches to a successful literature review. London: SAGE Publications (2012)
2. Pressman, R.S.: A practical approach to software engineering. Scientific and Technical Publishers (2004)
3. Hevner, A., Chatterjee, S.: Design Research in Informations Systems. Springer (2010)
4. Pańkowska, M.: Seminar for doctoral students. The paradigm of scientific research by Hevner et al., https://web.ue.katowice.pl/pank/DSRHevner.pdf
5. ISO Board: Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models. ISO (2011)

6. Sułkowski Ł.: Epistemology in management sciences. PWE Polish Economic Publisher (2005)
7. Kobyliński A.: The evolution of software product quality standards. In: Annals of the Collegium of Economic Economics. pp. 91–101. Warsaw School of Economics (2015)
8. Pan, J.Z., Staab, S., Aßmann, U., Ebert, J., Zhao, Y.: Ontology-Driven Software Development. (2013)
9. Haav, H.-M.: A Comparative Study of Approaches of Ontology Driven Software Development. INFORMATICA. 29, 439–466 (2018). doi:10.15388/Informatica.2018.175
10. OMG Board: Model Driven Architecture (MDA) MDA Guide rev. 2.0, https://www.omg.org/cgi-bin/doc?ormsc/14-06-01
11. Truyen F.: The fast Guide to model driven architecture (MDA), https://www.omg.org/mda/mda_files/Cephas_MDA_Fast_Guide.pdf
12. Siegemund K.: Contributions To Ontology-Driven Requirements Engineering, http://www.qucosa.de/fileadmin/data/qucosa/documents/16270/Thesis_13_03_2015_2-a.pdf, (2014)
13. Mauger, C., Schwartz, T., Dantan, J., Harbouche, L.: Improving users satisfaction by using requirements engineering approaches in the conceptual phase of construction projects: The elicitation process. In: 2010 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM). IEEE (2010)
14. Gruber, T.R.: A Translation Approach to Portable Ontology Specifications. Stanford, California (1993)
15. Borst, W.: Construction of Engineering Ontologies, PhD thesis, (1997)
16. Studer, R., Benjamins, R., Fensel, D.: Knowledge Engineering: Principles and Methods, "Data&Knowledge Engineering." Vol. 25 (1-2), s. 161-198 (1998)
17. Bjørner, D.: Software Engineering 3. Domains, requirements, and software design. Springer-Verlag (2006)
18. Stiefel, M., Oberg, R.J.: Application development using C♯ and .NET. Prentice Hall PTR (2002)
19. Bourque, P., Fairley, R. E.: Guide to the Software Engineering Body of Knowledge SWEBOK ® A Project of the IEEE Computer Society. IEEE (2014)
20. Bhatia, M.P.S., Kumar, A., Beniwal, R.: Ontologies for software engineering: Past, present and future. Indian J. Sci. Technol. 9, (2016). doi:10.17485/ijst/2016/v9i9/71384
21. Bhatia M., Kumar A., Beniwal, R.: Ontology Based Framework for Automatic Software's Documentation. In: Conference: 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom), pp. 421-424, IEEE (2015)
22. Bosch J.: Design and use of software architectures: adopting and evolving a product-line approach. Pearson Education (2000)
23. Caralt, J.C., Kim, J.W.: Ontology driven requirements query. In System Sciences. In: HICSS 2007. 40th Annual Hawaii International Conference, pp. 197c, IEEE (2007)
24. Ragala R., Vijayakumar V., Chauhan A.: Towards a Multi-level Upper Ontology/foundation Ontology Framework as Background Knowledge for Ontology Matching Problem. In: Procedia Computer Science 50:631-4. (2015)
25. Liao, L., Qu, Y., Leung, H.K.N.: A Software Process Ontology and Its Application. Studies on the Semantic Web, pp. 207-2017, IOS Press (2014)
26. Murugesh, S., Jaya, A.: Construction of Ontology for Software Requirements Elicitation. Indian J. Sci. Technol. IndJST. (Vol. 8), doi 10.17485/ijst/2015/v8i29/86271 (2015)
27. Inostroza, P., Astudillo, H.: Emergent Architectural Component Characterization using Semantic Web Technologies. Proc. Second Int'l Workshop Semantic Web Enabled Software Engineering (2006)

28. Kossmann, M., Wong, R., Odeh, M., Gillies, A.: Ontology-driven Requirements Engineering: Building the OntoREM Meta Model. In: 2008 3rd International Conference on Information and Communication Technologies: From Theory to Applications. pp. 1–6. IEEE (2008)
29. OMG Board: About the Ontology Definition Metamodel Specification Version 1.1, https://www.omg.org/spec/ODM/1.1/
30. Katasonov, A.: Ontology-driven software engineering: beyond model checking and transformations. . Int. J. Semant. Comput. 6(2), 205–242. (2012)
31. Haav, H.-M., Ojamaa, A.: Semi-automated integration of domain ontologies to DSL meta-models. Int. J. Intell. Inf. Database Syst. (IJIIDS), 10(1-2), 94–116. (2017)
32. Hoehndorf, R., Ngonga Ngomo, A.-C., Herre, H.: Developing consistent and modular software models with ontologies. Frontiers in Artificial Intelligence and Applications. pp. 399-412, IOS Press (2009)