Ontology-driven Requirements Engineering – A case study of OntoREM in the aerospace context

Mario Kossmann[1], Mohammed Odeh[2]
[1]AIRBUS / University of the West of England
mario.kossmann@airbus.com
[2] Software Engineering Research Group –
Centre for Complex Cooperative Systems
University of the West of England
mohammed.odeh@uwe.ac.uk

INCOSE, International Symposium 2010

Abstract

OntoREM is an Ontology-driven Requirements Engineering Methodology (process, methods and tools) that aims to improve the quality of requirements while also reducing the time and cost needed to develop, maintain and re-use requirements. In order to evaluate the potential of such an ontology-driven approach, OntoREM was applied to the aircraft operability (AO) domain and generic AO requirements for the wing design were developed. These requirements were subsequently compared to corresponding AO requirements that were developed for the wing design of two different development contexts for which a traditional requirements process was applied. Similarly, the elapsed process times to develop the requirements were compared, as measured for OntoREM and estimated for the traditional requirements process. The preliminary outcomes of this case study strongly suggest that the application of the OntoREM approach led to better quality requirements in considerably less time and hence at less cost per requirement. Further saving potential exists through increasing automation of OntoREM. In addition, there are several, additional advantages of applying this methodology that, for example, enable the re-use of domain ontologies including requirements in even less time i.e. at lower cost in the future, not to mention the advantages of explicitly capturing domain knowledge.

An industrial scale pilot application of OntoREM has been proposed in order to mature the methodology and its underlying IT infrastructure for the anticipated global use in the context of entire aircraft development programmes in order to prepare for full integration into the Airbus Product Development Process (PDP).

Introduction

Requirements engineering (RE) continues to face challenges especially in large, trans-national organisations, which are engineering-focused and produce complex, long-lead products and services in trans-national and multi-disciplinary contexts. Recent publications have identified difficulties to deploy the RE process in such environments and maintain the process over time [1, 2].

As a result, requirements are often immature and of low quality, and the RE process is likely to take longer than planned, and hence, will be more costly than originally budgeted for. This, in turn, will severely affect the successful completion of projects or programmes. Also, essential resources are retained longer in delayed programmes and, therefore, they may not be available for new programmes [1, 2].

A more knowledge-driven as opposed to process-driven approach to RE may bear the potential to produce better quality requirements, faster and cheaper. Process-driven RE is when the RE process has been pre-defined and the requirements development phase is carried out following that process in guiding all relevant activities and related deliverables. Each defined process step vis-à-vis milestones results in a number of deliverables (e.g. instances of the requirements document). However, rework (i.e. doing the same work again, as opposed to ‘refinement’, which means the necessary creation of more detailed information) at a later point in time will be needed, because at the time the process steps are carried out, some relevant information may not be available yet. In other words, deliverables may not be complete yet. Any such rework will cause delays and additional costs not only for the team in question, but also potentially for other teams involved in a given project or programme [2].

Knowledge-driven RE is when requirements development is carried out following the availability of the relevant knowledge. Depending on updates of knowledge (such as domain concepts, relationships between them, etc.), specific RE activities will be initiated, i.e. steps of the RE process will be triggered when needed or possible, based on the available domain knowledge. The outcome of such steps would be the relevant sets of mature requirements that are not necessarily identical with the documents described above, but they would be complete, based on readily available knowledge. There may also be the need (although not as frequently) to do iteration loops, but the difference is that this would only be the case if the agreed knowledge has changed. In the previous case, documents are released, knowing that the information needed is not even available and rework will definitely be required [2].

The focus moves from process steps for defined deadlines (but immature deliverables) to the knowledge needed, from which will flow the deliverables when feasible. There is a lot of documented but unpublished (for obvious reasons) evidence indicating that the former approach invariably leads to significant corrective rework, additional costs and delays. The latter approach will not require as much rework (because misunderstandings are reduced, and activities are driven not by unrealistic deadlines but by the availablity of relevant knowledge) and enable re-use of both requirements and design solutions including validation and verification plans etc. However, this is anticipated to be more difficult to project manage with current PM tools [2]. A recent process model, namely TUREP (Towards a Universal Requirements Engineering Process) adopts project and configuration management workflows as integral components in the RE process that are both linked to the overall systems engineering process [3].

Some work has been done to explore knowledge-driven RE as opposed to merely process-driven RE in the context of the OntoREM research project – a joint project between the University of the West of England and Airbus. Within this project, an Ontology-driven Requirements Engineering Methodology has been developed in order to explore the potential benefits of such an approach over traditional RE approaches by means of a case study in the aerospace industry [4].

The present paper will give a brief overview of the OntoREM methodology and how it was applied to a case study within the aerospace context in order to evaluate the potential of such an ontology-driven approach, before discussing both the selected evaluation framework and the findings of the case study.

Overview of OntoREM

OntoREM is one example of how a knowledge-driven as opposed to merely process-driven approach to RE could be put into practice in the future to help overcome some of the problems mentioned above. Gruber (1992) defines ontology as ‘a specification of a conceptualization’. In this context, *specification* is defined to mean a ‘formal and declarative representation’, whereas *conceptualization* means ‘an abstract, simplified view of the world’ [5].

Figure 1 depicts a high-level overview of the OntoREM approach. In the problem space, for instance during the development of a new aircraft system, some typical problems and needs in the given context are likely to be already well known.

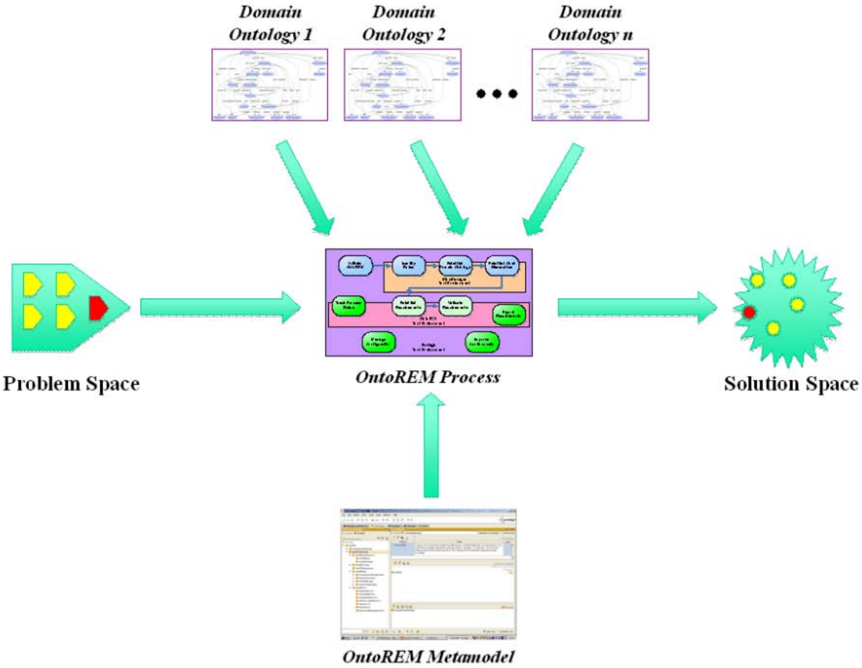

Figure 1: Overview of OntoREM

However, new areas in the problem domain need to be explored or elicited (red) and considered jointly with the existing parts (yellow) to be the specific problem space, when defining the solution space (requirements specification). The latter, in turn, is likely to contain existing parts of the solution that are available (yellow) that may or may not be sufficient to satisfy the problem domain. In most cases, some new, additional parts of the solution space (red) will have to be developed in order for the overall solution to satisfy the problem at hand. The outcome in the solution space is the requirements specification for a new aircraft system, which is likely to embody both existing and newly identified requirements [4].

In order to elicit the missing parts of the problem space and develop the missing parts of the solution space, the OntoREM process consists of a number of workflows and associated activities that are potentially iterative and concurrent, and conducted with all relevant domains, re-using where possible existing domain ontologies or knowledge bases, or building new domain ontologies where needed (with the help of relevant stakeholders and domain experts). The OntoREM process, the concepts and relationships needed to apply it, roles, tools, as well as goal and requirement templates are specified in the OntoREM Metamodel (as an ontology model) [4].

The Case Study

Objectives

The present case study aimed to achieve a number of objectives: First, to contribute to improve the overall requirements process and content for a given development context in terms of requirements quality and cost; second, to produce relevant data that could be directly compared with data from previous development contexts and thereby form the basis to answer the question whether an ontology-based approach to requirements engineering bears the potential to develop better quality requirements in less time and at less cost; and third, to further evolve and critically evaluate OntoREM, with the aim to produce a more mature version of the methodology that could be applied at a larger, industrial scale in order to prepare for full integration into the Airbus Product Development Process (PDP).

Selection of scope

The requirements developed using the OntoREM approach were focused on AO requirements for the wing design. *First*, non-functional requirements have been identified as being very challenging to develop [1, 10] and AO requirements consist, to a large extent, of non-functional requirements, some of which are specific to the wing domain, while others are of relevance across a number of aircraft domains such as the fuselage, landing gear or fuel system domains. *Second*, the time and budget restrictions imposed on the OntoREM project required a focused approach, limiting investigation to a smaller area of particularly interesting and challenging requirements to be developed using OntoREM [4]. The case study took place from June to September 2009. *Third*, the data available from recent development contexts allowed for a direct comparison with the AO requirements that were developed during the wing case study. *Fourth*, while applying OntoREM using the wing AO requirements, a number of requirements elicitation methods could be used in the framework of OntoREM, for instance a goal-based approach [6, 7, 8, 9]. *Finally*, the selected context, i.e. the wing engineering domain, is a typical (and arguably representative) example of a trans-national, highly complex, multi-functional and multi-disciplinary engineering environment, as it is often found within the aerospace industry. Such environments have been identified to be very challenging contexts for RE [1, 2].

Evaluation framework

The evaluation framework was used to compare the achieved results of the application of OntoREM for the development of generic AO requirements for the wing design of a specific aircraft programme with AO requirements that had been developed for the wing design of two previous development contexts. It consists of a number of quality, time and cost criteria that were observed during the application of OntoREM, and measured (regarding quality) and estimated (regarding process times and costs) following the application of the traditional requirements process. The OntoRAT tool (Ontology-driven Requirements Analysis Tool) was developed during the OntoREM project and deployed to both enhance the development of requirements as integral part of OntoREM, and assess the quality of those requirements [4, 10]. The interested reader is invited to look at reference [4] for a description of the OntoREM Metamodel that specifies (in the form of ontology) the OntoREM relationships that are used by the OntoRAT tool; and reference [10] for a more detailed view on the OntoRAT tool itself.

Table 1 explains the main criteria used including their definition, a statement of the purpose, and the selected approach to assess the relevant criteria for all three cases that were compared. The three quality criteria were selected because they look at the quality of individual requirements from three different, complementary angles and allow for direct comparison between corresponding requirements that were developed in the three different development contexts considered. The differences in how the criteria were measured in development contexts X and Y as opposed to Z can be explained by limitations of the available format of the requirements from contexts X and Y.

Table 1: Main criteria of the evaluation framework

Criterion	*Definition*	*Purpose*	*Measurement/estimation*
Q_1 (Quality Indicator 1)	This indicator expresses in percent the presence (on average for all AO requirements) of the following mandatory sub-components of each requirement statement (Halligan, 2010): 'Actor', 'condition for action' (only if applicable), 'shall action', 'object of action' and 'destination of action' [11].	To measure and compare the completeness of individual requirement statements.	Contexts X and Y: Manual assessment and counting. OntoREM (Context Z): Use of OntoRAT tool to analyse the newly developed requirements for completeness of individual statements, mandatory attributes and links, and automatic compilation of statements into the harmonised structure of each individual requirement statement.
Q_2 (Quality Indicator 2)	This indicator expresses in percent the presence (on average for all AO requirements) of the following mandatory attributes: 'Requirement status', 'rationale', 'source' (or link information), 'owner' and 'stakeholder'.	To measure and compare the completeness of individual requirements in terms of recorded attribute and traceability information.	Contexts X and Y: Manual assessment and counting. OntoREM (Context Z): Use of OntoRAT tool to analyse the newly developed requirements for completeness of individual statements, mandatory attributes and links, and automatic compilation of statements into the harmonised structure of each individual requirement statement.
Q_3 (Quality Indicator 3)	This indicator expresses in percent the overall compliance of all AO requirement statements with the following conditions: 1. The actor of the requirement statement is the same per sub-section of the requirements document ('it' is not allowed). 2. The structure of the statement is as specified in the OntoREM requirements template (based on Halligan, 2010 [11]). However, a different order of the sub-components 'condition for action' (only if applicable), 'destination of action' and 'constraint on action' following the actor is tolerated. 3. There is only one requirement in each requirement statement. 4. There is no mandatory sub-component missing (the 'shall action' must contain the word 'shall', e.g. not 'should', 'will' or 'must').	To measure and compare the degree of compliance of individual requirements with a harmonised structure of the requirement statements. The latter facilitates in particular the analysis and negotiation, and validation phases of the requirements process.	Contexts X and Y: Manual assessment and counting. OntoREM (Context Z): Use of OntoRAT tool to analyse the newly developed requirements for completeness of individual statements, mandatory attributes and links, and automatic compilation of statements into the harmonised structure of each individual requirement statement.

	If one of the conditions is not met, the requirement in question is counted 0, if all conditions are met it is counted 1. The overall indicator is calculated by taking the sum of all recorded values and dividing it by the number of requirements considered.		
NAO (Number of AO requirements)	Number of requirements compared.	To calculate times/costs per requirement.	Contexts X and Y: Manual counting. OntoREM (Context Z): Read-out in Protégé.
TAO (Development time per requirement)	Time in minutes per requirement that was spent by all relevant participants in the requirements process during the elicitation, analysis and negotiation, documentation, and validation phases of the requirements process.	To be able to compare process times of individual workflows or the entire requirements development process per requirement.	Contexts X and Y: Estimates by the requirement owners and relevant stakeholders. OntoREM (Context Z): Observed process times.
CAO (Development cost per requirement)	Cost in Euro per requirement that is calculated by multiplying the total requirements development time per requirement in hours by the relevant hourly rate in Euro.	To be able to compare the costs of requirements development per requirement.	Calculated from NAO and TAO using the relevant hourly rate.

Description of the Findings

The main outcome of applying OntoREM to develop generic AO requirements for the wing design were validated requirements (Context Z), as well as process measures taken during the application of OntoREM (times needed for individual workflows, with and without the relevant participants). In addition, the OntoREM methodology itself has been evolved and incrementally improved during the case study in light of feedback and experience.

Furthermore, corresponding requirements from two specific aircraft development contexts have been measured and the responsible key actors from these programmes estimated the relevant process times based on their own experience when applying the traditional requirements process (Contexts X and Y).

Based on the data collected (see tables 2-3), comparisons were made between the requirements quality and related process times as measured during the application of OntoREM (Context Z) and as measured and estimated following the application of the traditional requirements process (Contexts X and Y).

Table 2: Obtained values regarding the main criteria [%]

	Context X	*Context Y*	*Context Z*
Q1	68%	85%	100%
Q2	28%	96%	100%
Q3	17%	20%	100%
N*AO*	235	66	131
T*AO*	22%	100%	7%
C*AO*	22%	100%	7%

Table 3 provides a more detailed view of the relative process times that were needed to develop the requirements considered in all three cases, making the distinction between the traditional phases of requirements development, i.e. elicitation, analysis and negotiation, documentation, and validation (Kotonya and Sommerville, 1998 [12]).

Table 3: Process times per requirement [%]

	Context X	*Context Y*	*Context Z*
Elicitation	57	10	79
Analysis and negotiation	27	77	10
Documentation	1	10	3
Validation	16	4	8

Analysis of the Findings

The analysis can be summarised and grouped under the headings of requirements quality, time and cost of requirements development, as well as additional advantages of applying OntoREM. As mentioned above, the outcomes of the present case study have to be considered to be preliminary and thus more detailed critical evaluation will have to take place in the near future in order to confirm the obtained results. However, the preliminary results of this case study provide very clear indications regarding the potential of the proposed ontology-driven approach to RE.

(a) Requirements quality

The three quality indicators described in Table 1 are mainly concerned with the completeness of individual requirements statements, the completeness of individual requirements in terms of use of mandatory attributes and link information, and the structure of requirement statements. Figure 2 shows the obtained measurements for all 3 quality indicators in all the three cases considered. The maximum value that could be achieved for each indicator is 1.0 (= 100%) so that if all criteria are met to 100% the maximum value in the diagram is 3.0 (= 3 x 100%). Those requirements that were newly developed by applying OntoREM have met the defined three indicators to 100% each because the OntoREM Metamodel specifies the template for new requirements in such a way that the individual requirement statement has to be complete, all mandatory attributes have to be populated and links have to be established within the relevant domain ontology, as well as the requirement statement is automatically compiled to be in the specified structure. Any deviation of this would have been indicated while following the OntoREM process and few deviations have indeed been identified during the analysis phase and could be corrected immediately.

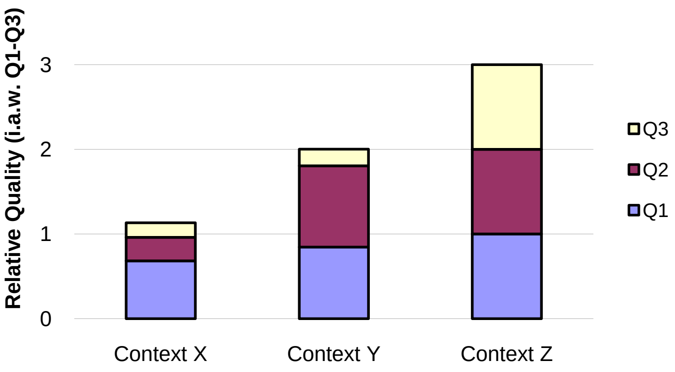

Figure 2: Requirements Quality Measurements

Looking at the first quality indicator, it can be argued that in context X individual requirement statements were already relatively complete on average, i.e. not many sub-components of the requirement statements were missing. In context Y the result obtained was even better, although there were still some incomplete requirement statements to be found.

Regarding the second quality indicator, it seems that in context X attributes and links were hardly used as required, which could be explained by the fact that the traditional RE process as applied had not yet reached a high level of maturity and engineers were not used to work with the same richness of information associated to each requirements in terms of attributes and links. So was it frequently perceived as too time consuming to link all requirements in the deployed requirements database or to fill all mandatory attributes of each requirements, let alone the complementary ones. In context Y, the use of attributes, at least concerning the mandatory ones, had increased significantly, which explains the high readings obtained in this context.

The structure of requirement statements that was subject to the third quality indicator has clearly not been in the focus within both contexts X and Y, where the majority of requirement statements are not in the same structure, the actors of requirements in the same sub-section of the requirements document are often not identical, there is frequent use of ‘it’ as the requirement actor, and so on. Having similar structure, however, significantly enhances the analysis of requirements as missing components, duplications and conflicts can be spotted much more easily.

To summarise, the obtained data have shown increased maturity of the application of RE from context X over context Y to context Z, with increased focus being placed on the requirement statement itself first, via the increased use of associated information in terms of attributes and links, to improving the structure of requirement statements with the aim to enhance requirements analysis and validation. The obtained quality readings strongly suggest that the application of OntoREM led to better quality requirements than was the case when using the traditional requirements process.

(b) Time and Cost of Requirements Development

To be in line with the assumption that process costs can be considered to be directly proportional to the process times measured or estimated during the present case study, time and cost will be discussed jointly. Figure 3 provides an overview of the process time/cost per requirement during the major phases in the requirements development process, in all the three cases considered. The total process time from elicitation to validation per requirement was longest in context Y, which was taken to be 100% for the purpose of easy comparison. The absolute measurements are not displayed for confidentiality reasons. All other percentage values are normalised against this total time in context Y.

Before looking at the differences in process times/costs between the two applications of the traditional requirements process (contexts X and Y) and the application of OntoREM (context Z), the significant differences between the contexts X and Y are discussed. The total process time/cost per requirement in context X was much lower than that in context Y. This can be attributed to the contractual character and the high level of detail of most AO requirements in context X. Many requirements could not be changed even if analysis was carried out, it was very difficult to delete or combine partly duplicated AO requirements, or to resolve requirement conflicts easily when they were identified. This also seems to explain the fact that most of the process time within context X was spent on elicitation rather than on analysis, as opposed to context Y where by far most of the process time was spent on analysis and negotiation (followed by the time needed for elicitation and documentation of the requirements). In light of the available information regarding contexts X and Y, it can be argued that the most realistic context with which the results of the application of OntoREM should be compared is context Y. First, the AO requirements have the right level of detail for the wing design, i.e. in general they are not too detailed. Second, because of the above, the requirements quality was higher than in context Y, and, last, it is a more recent development context and the application of the traditional requirements process had reached a higher level of maturity than was the case in context X.

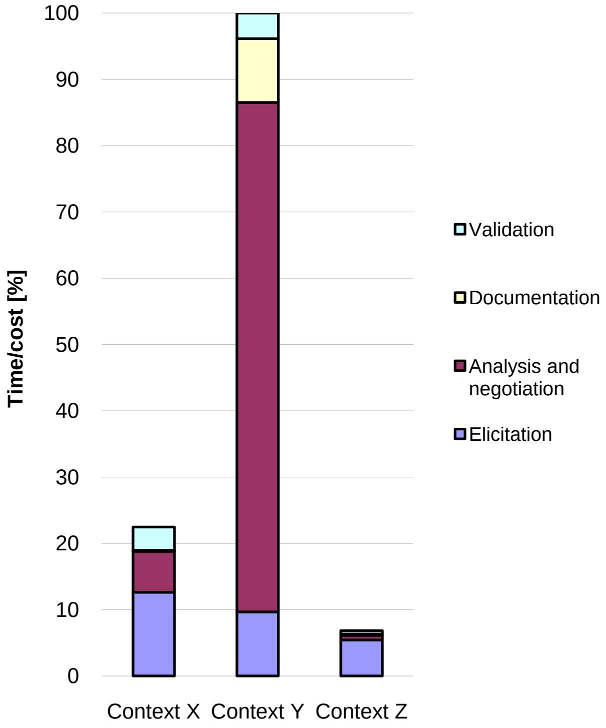

Figure 3: Development time/cost per requirement

On average, the development of new AO requirements using OntoREM took less than 10% of the corresponding process time that was spent in context Y (per requirement). Also, all main phases of the requirements process took considerably less time during the application of OntoREM compared to context Y.

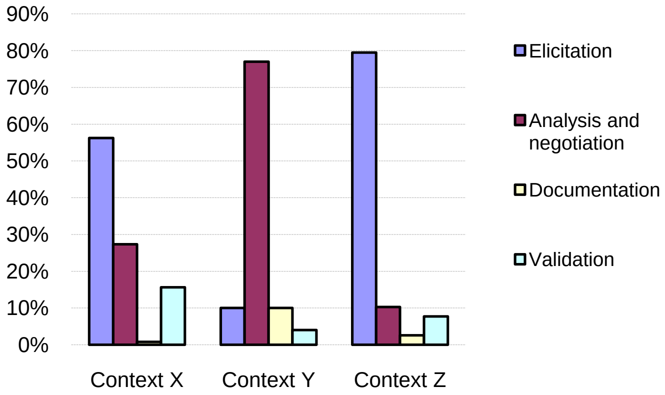

Figure 4: Relative development time per process phase

Figure 4 shows that using the OntoREM approach, the elicitation phase took about 80% of the total process time, whereas the analysis phase took under 10% of the total process time. In context Y, in contrast, the analysis and negotiation phase took by far the longest, i.e. 77% of the total process time in this context (using the traditional requirements process).

The fact that, during the OntoREM approach, analysis only took such a small percentage of the total process time seems to be due to the following two factors: First, goals and requirements that were elicited before the start of the analysis phase had been based on agreed domain ontology (including needs and derived goal hierarchies). Second, analysis (and later validation) was greatly enhanced and highly automated using the tool OntoRAT, which was developed within the context of the OntoREM project and is fully integrated into the methodology [10].

Furthermore, there is the additional potential to save process time through the increased automation of the OntoREM process. Figure 5 shows the main areas that were identified to offer further saving potentials through automation of tooling interfaces needed for data export and import within OntoREM, as well as enhancement of the default setting capability within the ontology editor. The most significant of these savings could be realised by automating the interface from the mind-mapping tool to the ontology editor (in this example application from MindManager to Protégé).

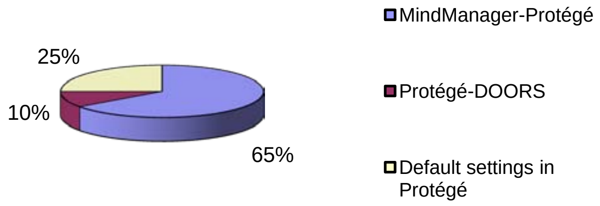

Figure 5: Further saving potential through automation

To summarise, context Y seems to be a more realistic context for comparison with the OntoREM application during this case study than context X. Compared with the application of the traditional requirements process, the total process time using OntoREM only took less than 10% of the total process time per requirement in context Y, with the main emphasis on the elicitation phase and tool-supported enhancement of analysis and validation. The obtained quality readings strongly suggest that the development of better quality requirements by applying OntoREM was achieved in considerably less time than was the case in the most realistic context to be compared with, i.e. context Y using the traditional requirements process. However, there is further saving potential through the increased automation of OntoREM tool interfaces (between some of the tools used within the methodology such as MindManager and Protégé), and enhancement of default settings in the ontology editor Protégé.

(c) Additional advantages

Not considering the development of requirements as such, one of the most significant, additional advantages of adopting the OntoREM process is the development of domain ontology that serves as the formal repository of validated domain knowledge, as this domain ontology allows the re-use of domain knowledge (including generic domain requirements). Also, once this domain ontology has been validated by the respective domain experts, it continues to be used for new projects with minimum involvement of the domain experts other than for the cyclic or periodic verifications to check the validity of newly incorporated concepts and also the integrity of the ontology itself. Thus, it is anticipated that the re-use of domain ontologies including requirements will be much less time-consuming.

Also, valuable feedback can be given to programme management such as percentages and absolute numbers of re-useable requirements for a given project, which allows for better estimations of the time needed to develop the project's requirements or risks due to low percentages of re-useable requirements etc.

To summarise, there are a number of additional advantages of adopting the proposed OntoREM approach that enable the re-use of domain ontologies including requirements in even less time (i.e. at lower cost) in the future, not to mention the benefits of explicitly capturing domain knowledge from a knowledge management viewpoint.

Conclusion

The application of OntoREM as an ontology-driven requirements engineering methodology, using three real cases from the aerospace industry, suggests that there is clear potential that when adopting such a methodology this will lead to better quality requirements, which are specified and validated in both significantly less time and at less cost. This can be further amortised over a number of cycles of domain ontology re-use and enrichment. Also, there is further saving potential through the increased automation of OntoREM tool interfaces and the enhancement of default settings in the ontology editor. Finally, there are a number of additional advantages of applying the OntoREM process from a knowledge management viewpoint, for instance the explicit capture and validation of domain knowledge.

Limitations of the present case study were for instance that the development of requirements could not be conducted (in all three contexts) in a controlled environment; and for each development context only one approach was applied, either a traditional RE process in contexts X and Y, or OntoREM in context Z.

The OntoREM prototype is ready for local use (and is actually being used) but cannot be considered sufficiently mature yet for an industrial scale deployment, for a number of reasons: first, part of the OntoREM tool environment consists of new software developments (e.g. the OntoRAT tool) and existing freeware that is readily available to download; second, trans-national use of OntoREM by many users simultaneously has not been tested yet; third, the concurrent application of several instances of OntoREM at different layers within an aircraft development programme has not been tested yet; fourth, change management issues need to be solved at the process and tool level and tested in the above-mentioned environments; and finally, the underlying IT infrastructure will need to be extended to include all other tool components and networking required for secure and reliable execution in trans-national, concurrent engineering contexts of industrial scale such as the civilian aerospace industry.

However, an industrial scale pilot application of OntoREM has been proposed in order to mature the methodology and its underlying IT infrastructure for global use in the context of entire aircraft development programmes, i.e. to prepare for full integration into the Airbus Product Development Process (PDP).

Although the detailed OntoREM methodology and some of the integrated tooling are not currently available to the wider public, the outcomes of the current research are intended to give useful indications to the wider SE community regarding the potential of ontology-driven approaches to RE and future areas of research.

References

[1] M. Kossmann, M. Odeh, A. Gillies, 'Tour d'horizon' in Requirements Engineering – Areas left for exploration, INCOSE, 2007.

[2] M. Kossmann, M. Odeh, A. Gillies, R. Wong, From Process-Driven to Knowledge-Driven Requirements Engineering Using Domain Ontology, IS'08, INCOSE, 2008.

[3] M. Odeh, TUREP: Towards a Universal Requirements Engineering Process, Proceedings of the ACIT2009 Conference, Sanaa, Yemen, 2009.

[4] M. Kossmann, M. Odeh, A. Gillies, S. Watts, Ontology-driven Requirements Engineering with Reference to the Aerospace Industry, ICADIWT '09, IEEE, London, UK, 2009.

[5] T. R. Gruber, A translation approach to portable ontology specifications, Knowledge Acquisition, vol. 5, no. 2, pp. 199-220, 1992.

[6] A. van Lamsweerde, E. Letier, Integrating Obstacles in Goal-Driven Requirements Engineering, O-8186-8368-6/98, IEEE 1998.

[7] P. Giorgini, J. Mylopoulos, R. Sebastiani, Goal-oriented requirements analysis and reasoning in the Tropos methodology, Engineering Applications of Artificial Intelligence 18 (2005) 159-171, Elsevier Ltd. (www.elsevier.com/locate/engappai), 23 January 2005.

[8] J. Mylopoulos, L. Chung, E. Yu, From Object-oriented to Goal-oriented Requirements Analysis, Communications of the ACM January 1999/Vol. 42, No. 1.

[9] J. Mylopoulos, Information Modeling in the Time of the Revolution, Information Systems Vol. 23, No. 3/4, pp. 127-155, 1998.

[10] Y. Al-Hroub, M. Kossmann, M. Odeh, Development of an Ontology-driven Requirements Analysis Tool (OntoRAT) – A Use-case-driven Approach, ICADIWT '09, IEEE, London, UK, 2009.

[11] R. Halligan, Requirements Quality Metrics: The Basis of Informed Requirements Engineering Management, http://www.ppi-int.com/ newsletter/ SyEN-013.php, 27 February 2010.

[12] G. Kotonya, I. Sommerville, Requirements Engineering: Processes and Techniques, Wiley, Chichester, 1998.

Biography

Mario Kossmann is currently pursuing his Ph.D. studies at the University of the West of England and a Systems Engineer and Capability Integrator for Airbus, having previously worked for Blohm & Voss as a Systems Engineer, Technical Manager and also Consultant in Services Marketing. He has served as an officer with the German and French navies, and was awarded a MEng in Aerospace Engineering from the University of the Federal Armed Forces in Munich (Germany), and an MBA from the University of Warwick (UK). Mario is the author of the book 'Delivering Excellent Service Quality in Aviation' (Ashgate 2006). In 2005, Mario transferred from Airbus Deutschland to Airbus UK, tasked with the trans-national harmonisation and coordination of the SE related training and support activities, before assuming responsibility for research and development of Requirements Based Engineering (RBE) within Airbus. Currently, Mario is the Requirements Management process, methods and tools leader within Airbus. He is also the Airbus lead investigator on the OntoREM project.

Mohammed Odeh, Ph.D., leads the Software Engineering Research Group (SERG) in the Centre for Complex and Cooperative Systems (University of the West of England). He has more than 24 years experience in the field of software engineering with fourteen years in the banking industry. His research interests are focused on: service oriented software engineering models and processes, model-based software cost estimation, knowledge and business process driven requirements engineering with particular interest in bridging the gap between process, information, and knowledge models. He has over 50 journal, conference and book chapter publications. Dr. Odeh has been supervising a number of Ph.D. research students with successful completions. Also, he has been a co-investigator on three EU funded projects and a principal investigator on recent industrial knowledge transfer and research collaboration with Airbus. Dr. Odeh is the UWE-Bristol lead investigator on the OntoREM project with Airbus.

Acknowledgements:

The authors would like to thank Stephen Watts, Andy Gillies and Younes Al-Hroub for their helpful advice and kind support in many fruitful discussions, the latter in particular also for his efforts in developing the OntoRAT tool.