Ontology-driven Requirements Engineering with Reference to the Aerospace Industry

Mario Kossmann[1]
Airbus
Mario.Kossmann@airbus.com
Mohammed Odeh[2]
University of West of England
Mohammed.Odeh@uwe.ac.uk
Andrew Gillies
University of West of England
Andrew.Gilles@uwe.ac.uk
Stephen Watts
Airbus
Stephen.Watts@airbus.com

[1&2] *for correspondence as Airbus and UWE Principal Investigators, respectively, of the OntoREM Project*

ABSTRACT: *This paper provides an overview regarding the application of an ontology-driven requirements engineering methodology, namely OntoREM, in the aerospace industry with the objective to assess the extent to which this approach has the potential to develop better quality requirements in less time and at less cost compared to traditional requirements engineering processes, taking the Airbus wing-engineering requirements as the case study. The OntoREM's Metamodel is briefly introduced, which embodies both the key process and task ontologies that guide the OntoREM related requirements engineering activities along with the supporting tooling environments, both existing and newly developed ones. Finally, the paper provides an overview of a small number of focused case studies in the aerospace context that have already been conducted and will help to further evolve and critically evaluate OntoREM with the objective to investigate the potential of a more knowledge-driven as opposed to process-driven approach to RE.*

Keywords: Ontology requirements, Aerospace, Ontology metamodel

Received: 14 September 2009, Revised 12 October 2009, Accepted 22 October 2009

1. Introduction

Alexander and Stevens (2002) reviewed a number of studies regarding the reasons of project failure such as the Standish Group's chaos reports and concluded that 'five of the eight major reasons for failure are requirements-based. The other three relate to management, while – perhaps surprisingly – none of them is technical' [1]. There is a lot of unpublished evidence in many industries that supports their findings. This is to say that requirements problems in the widest sense appear to be at the heart of the most serious problems that the majority of projects or programmes are facing. Hence, any approach to improve the Requirements Engineering (RE) process and the quality of requirements themselves is drawing the attention to where it belongs, i.e. to the most urgent root cause of project or programme difficulties with arguably the highest impact on product quality and financial performance.

A recent publication (Kossmann et al. 2007) – an overview of the current state of requirements engineering – focused on processes, the use of ontologies, the coverage of non-functional requirements, as well as the interfaces to other engineering disciplines. The authors concluded that some of the main areas to be explored, where further research appears to be very promising (i.e. where the need to investigate in more depth or extend empirical evidence to related fields had been identified by recent publications) are: (1) combining considerations of products and services together in RE, (2) focusing on the systematic determination of the goals of the served system and obstacle analysis at the goal level, (3) the use of domain ontologies (and semantic web technologies) in both the problem and solution spaces, and (4) the systematic inclusion of non-functional requirements (NFRs) [2].

In an attempt to address the identified issues, some work has been done to explore knowledge-driven RE (as opposed to merely processdriven 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 small number of limited case studies in the aerospace industry [3, 4].

The present paper looks at this ongoing investigation of knowledgedriven RE by means of the OntoREM project and gives an overview. Finally, the paper offers an overview of a number of case studies in the aerospace context that will allow for the further evolution and critical evaluation of OntoREM as a methodology and also answer the question of whether knowledge-driven approaches to RE could have the potential to develop better quality requirements in less time and at less cost.

2. From Process-Driven to Knowledge-driven Requirements Engineering

It appears that 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, RE seems to be particularly challenging. Recent publications have identified difficulties to deploy the RE process in such environments and maintain the process over time [5, 6]. As a result, requirements are often immature and of low quality, and the RE process is more likely to take longer than planned. Hence, RE becomes more costly than originally budgeted for. This, in turn, can severely affect the successful completion of projects. Also, essential resources are retained longer in delayed projects and, therefore, may not be available for new projects.

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 the 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 is needed, because at the time the process steps are carried out, part of the relevant information may not be available yet. In other words, deliverables are not 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 [3].

Knowledge-driven RE is when requirements development is carried out following the availability of the relevant information. 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 previously described, but they would be complete, based on the available information. 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 information has changed. In the previous case, documents are released, knowing that the information needed is not even available and rework will definitely be needed.

Thus, the focus moves from process steps with defined deadlines (but immature deliverables) to the knowledge needed, from which will follow the deliverables when and where feasible. There is a lot of unpublished 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.) but can be expected to be more difficult to project-manage with current PM tools [3].

OntoREM (Ontology-driven Requirements Engineering Methodology) 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. The OntoREM project is a joint research project of Airbus and the University of the West of England [3, 4].

Gruber (1993) defines ontology to be ‘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’ [7]. There are a number of different approaches to using domain ontology in order to enhance requirements such as the requirements elicitation method suggested by Saeki and Kaiya (2006), which makes use of domain ontology (as domain knowledge) in order to enhance completeness and consistency of requirements, which was evaluated using the case study of software music players [8]. However, OntoREM differs from other approaches in the field in that it aims to address and investigate at the same time the following aspects: (1) re-use of existing domain ontologies to develop requirements, (2) domain-process-driven elicitation of needs, (3) construction of goal hierarchy in ontology, (4) emphasis on nonfunctional requirements via soft goals, (5) reasoning at both goal level and requirements level, (6) role-driven process guidance, (7) concurrent work with different stakeholders (who are at different stages of the requirements process), (8) queries-driven enhancement of project management (PM) and RE activities, (9) queries (from a PM, RE, or domain viewpoint) by users without in-depth knowledge of the SQL or ontology, and (10) re-use of requirements from previous programmes or projects.

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 this given context are likely to be already well known. 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). In the latter one, in turn, there are likely to be existing parts of the solution already available (yellow) that may or may not be sufficient to satisfy the problem domain. Probably, some new, additional parts.

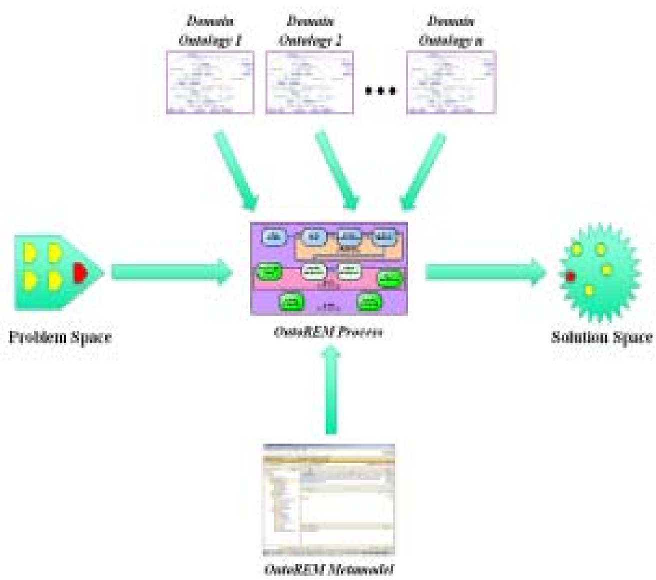


Figure 1. Overview of OntoREM

solution space (red) 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.

In order to elicit the missing parts of the problem space and develop the missing parts of the solution space, the OntoREM process is 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, roles, tools, as well as goal/soft goal and requirement templates are specified in the OntoREM Metamodel (as an ontology model).

Though OntoREM is not mature yet for industrial use, it is being evolved and will be tested to prepare for operational deployment at a larger scale. Some very promising results have already been achieved from a series of applications of individual aspects of the methodology in the context of the aerospace industry [3, 4]. We anticipate that OntoREM, at least within the business contexts mentioned above, may well show the potential to help organisations significantly increase the quality and completeness of their requirements, while potentially cutting costs and time needed to develop and manage changes over time. A comprehensive case study in the wing-engineering domain will enable critical evaluation of OntoREM compared to traditional RE processes that were used in previous Airbus wing developments.

As far as the above findings regarding the major reasons for project failures are concerned, the OntoREM project is, therefore, directly addressing the major root cause of problems that projects or programmes are likely to experience, i.e. problems related to the requirements process and the quality of requirements. In the following section, the OntoREM Metamodel is described in more detail, covering process and tooling aspects.

3. The Ontorem Metamodel

The OntoREM Metamodel is specified in the form of an ontology itself and consists of process related, role related and domain related components. It is a comprehensive specification of the ontologydriven requirements engineering methodology, including the underlying concepts in the RE domain and relationships between them. A Metamodel can be defined as ‘explicit model of the constructs and rules needed to build specific models within a domain of interest. This characterises a valid Metamodel as an ontology, since such constructs and rules represent entities in a domain and their relationships, i.e. a set of building blocks used to build domain models. In other words, a Metamodel is an ontology used by modellers’ [9].

The approach suggested by Noy and McGuinness [10] was initially followed to build the OntoREM Metamodel [4]. The ontology web language OWL DL (Description Logic) was selected to express the OntoREM Metamodel. The OWL vocabulary is built on top of the RDF(S) vocabulary (Resource Description Framework) and one of its important features is its ‘extreme richness for describing relations among classes, properties and individuals’ [9]. The Protégé tool [11] has been utilised as the ontology development environment for the OntoREM project, being the current leading ontology development environment [9].

The key to success when building ontology seems to be stakeholder and domain expert involvements through all the ontology development process to achieve a number of key objectives such as that the ontology content reflect accurate and recognised current domain knowledge.

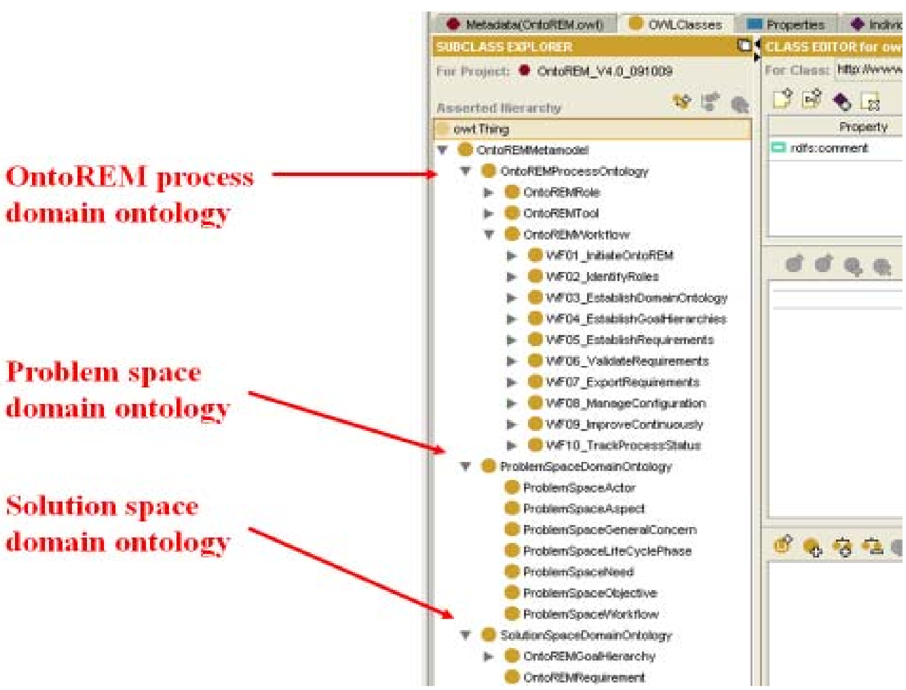


Figure 2. The OntoREM Metamodel

and concepts, and the relevant stakeholders in the ontology be convinced of its usefulness and accept it as the reference regarding the knowledge and concepts contained in it. However, because of limited stakeholder and domain expert availability compounded by the lack of other resources, the process of domain ontology development is often both iterative and concurrent [4].

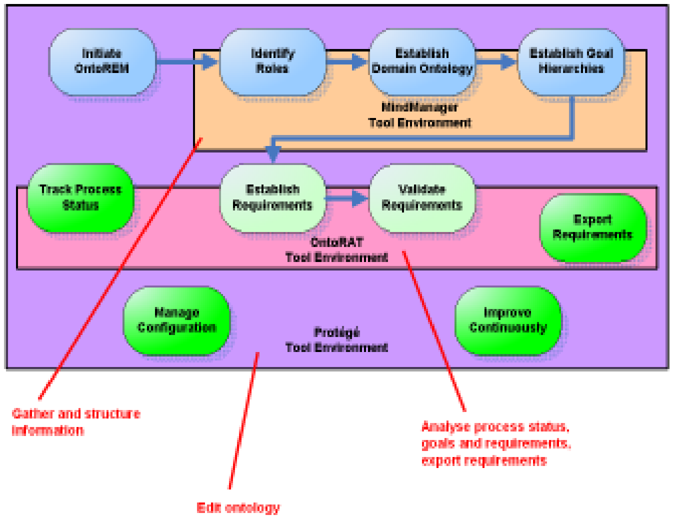


Figure 3. The OntoREM Process and Tooling Environment

The OntoREM Metamodel (see Figure 2) consists of the OntoREM Metamodel ontology and a number of domain ontologies. When applying OntoREM, existing domain ontologies are imported. Some additional domain ontologies may have to be developed with the relevant stakeholders and domain experts, if no relevant domain ontologies are available yet, or existing ones that are not up-to-date may have to be validated again with the relevant stakeholders and domain experts. Figure 2 provides a high-level overview of the three main parts of the OntoREM Metamodel, which is maintained in the form of an ontology itself: (1) the OntoREM process ontology, (2) the problem space ontology, and (3) the solution space ontology. The former contains the OntoREM domain knowledge (how to apply OntoREM?) and consists of the classes 'OntoREMProcess', 'OntoREMRole' and 'OntoREMTool' defining the OntoREM process, how it is conducted with which roles using what tools to support the process. The second main part concerns the specification of the problem space of any application domain, which is using OntoREM to develop and manage their domain requirements, and consists of the classes 'ProblemSpaceActor', 'ProblemSpaceAspect', 'ProblemSpaceGeneralConcern', 'ProblemSpaceLifeCycle-Phase', 'ProblemSpaceNeed', 'ProblemSpaceObjective', and 'ProblemSpace-Workflow'. The latter main part of the Metamodel concerns the specification of the solution space of the application domain in question and consists of the classes 'OntoREMGoalHierarchy', with the sub-classes 'OntoREMGoal' and 'OntoREM-Softgoal', and 'OntoREMRequirement'. These concepts with their described relationships define 'templates' of goals, soft goals and requirements that are used when creating an instance of a goal, soft goal or requirement, and are linked to the relevant areas of the available domain ontologies (in particular to the identified domain needs). The requirement instances represent the main outcomes of OntoREM, i.e. the requirements specification for a given domain in a given context (solution space). Both the problem space domain ontology and the solution space domain ontology together as specified in the Metamodel serve as the template to build domain ontology including domain requirements for any domain that starts using OntoREM.

Figure 3 provides an overview of the OntoREM process, which is partly based on the requirements engineering process as described by Kotonya and Sommerville (1998) [12], as well as the supporting tooling environments. All stages of the OntoREM process are specified as iterative and concurrent 'workflows' consisting of 'activities' rather than just individual steps of the process in a sequential order. This is more suitable in light of the concurrent and iterative nature of OntoREM. As mentioned above, this is mainly due to the fact that all relevant stakeholders and domain experts, who need to be met and worked with, will not always be available at the convenience of the requirements engineer and also because, in case of unavailability of knowledge or conflicts, problems will need to be resolved by revisiting activities that had already been performed.

The workflows 'Initiate OntoREM', 'Identify Roles', 'Establish Domain Ontology' and 'Establish Goal Hierarchies' (blue) comprise the activities needed in order to prepare, customise and start using OntoREM for a given project/programme, and build domain ontology including high-level domain needs (problem space), as well as develop goal hierarchies (goals and soft goals) through to the identification of root goals and root soft goals from which requirements will be derived (first part of the solution space).

The workflows ‘Establish Requirements’ and ‘Validate Requirements’ (pale green) encompass the activities needed to elicit, analyse, document and validate requirements based on the available domain knowledge (i.e. the problem space and first part of the solution space), thereby completing the solution space of the domain ontology. The workflows ‘Track Process Status’, ‘Manage Configuration’, ‘Export Requirements’ and ‘Improve Continuously’ (dark green) contain activities needed to keep under control and guide the OntoREM process, to export intermediary results from the domain ontology and to continuously improve the entire methodology over time.

The OntoREM process is mainly supported by the following available tools: Protégé (including the Protégé-owl plug-in), MindManager, and DOORS. The tool OntoRAT (Ontology-driven Requirement Analysis Tool) [13] is currently under development within the framework of the OntoREM research project. OntoRAT is mainly used to trace and analyse the process status, analyse goals and requirements, and export them. MindManager [14] is used for mind mapping in order to elicit and capture domain knowledge and develop goals/soft goals with relevant stakeholders and domain experts.

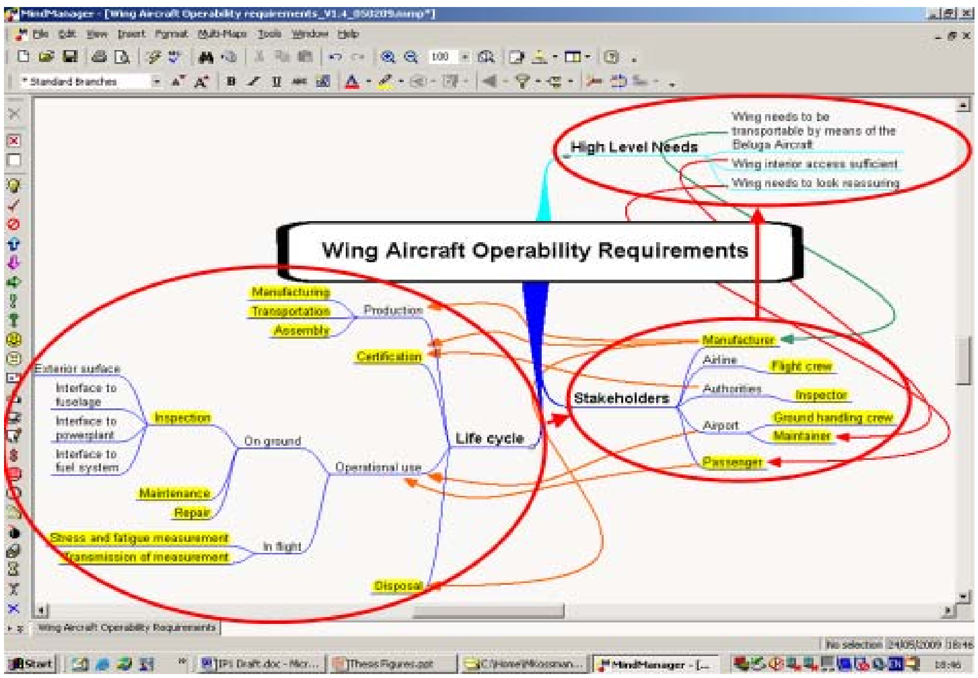


Figure 4. Developing Domain Ontology with Mindmaps

Protégé [11] is used to edit ontology, i.e. to create, extend, maintain, explore and re-use domain ontology. Finally, DOORS (IBM) [15] is used as the traditional requirements management tool with great strengths in terms of traceability (it has been used at Airbus since 2003).

Figure 4 shows an example of the use of a Mindmap (in the MindManager tool environment) in order to elicit the problem domain and identify high-level needs of the application domain in question. The outcome of this exercise is transferred into the Protégé tool environment to be used in subsequent process workflows. Mindmaps are used in the OntoREM approach because they assist in visualising the initially unstructured or less structured information with a visual model to structure and validate the information with the help of domain experts and relevant stakeholders. Figure 5 provides an example of how requirements that have been elicited, analysed and negotiated (using the Protégé tool environment) are transferred and documented in the requirements database (in the DOORS tool environment). In the Protégé tool environment, textual requirements are stored as instances of the concept ‘OntoREMRequirement’ using the generic requirement template. In the DOORS (IBM) tool, on the other hand, all available information about a requirement is stored in an object (within a requirements module that usually correspond to a requirements document) with all attribute information contained in DOORS attributes that are associated with that object and can be displayed as columns next to the requirement statement. Link information is contained in link modules that are associated with specific requirements modules.

The reason why both tool environments (i.e. both Protégé and DOORS) are needed is that both tools have been developed for different purposes and hence have different strengths and certain limitations: Protégé is needed to build and use ontology, which is of course essential for OntoREM as this methodology is explicitly driven by ontologies. However, for a number of reasons, Protégé is not at the moment very intuitive and user-friendly (as perceived by users without a background in the field of ontology). Users of Protégé within OntoREM need to be familiar with the underlying concept of ontology and understand the principles of the OWL language. In other words, only users that have familiarised themselves in detail with the concept of ontology will be able to use the tool efficiently. DOORS, on the other hand, is powerful in terms of traceability and flexibility to visualise and make use of requirements stored within the tool's database. DOORS also appears to be perceived as much more intuitive and user-friendly by requirement engineers. DOORS is useful in supporting the cascading of requirements and their evolution over time. Finally (and for the above reasons), the tool has been deeply embedded in many companies of the aerospace industry for some time. The above demonstrates that OntoREM makes use of the readily available tooling environments at the various stages of the RE process. Other tools and interfaces are being developed within the context of the OntoREM project, such as OntoRAT, if there were no existing tool solutions available to support specific aspects of the process.

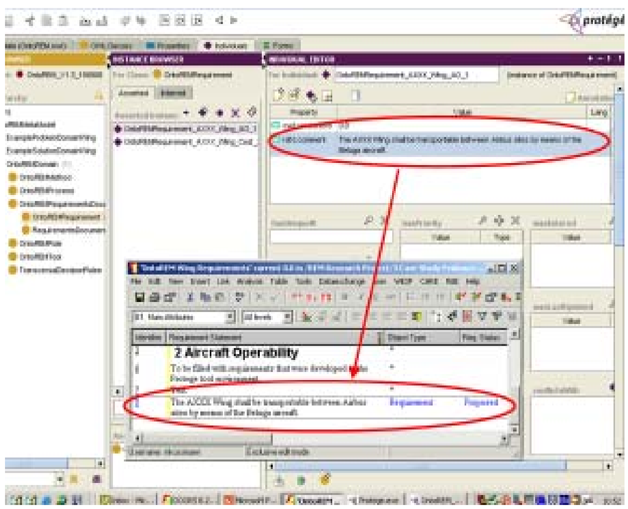


Figure 5. Exporting Requirements from Protégé to DOORS

4. First Applications of ONTOREM

OntoREM or specific parts of the methodology have already been applied in a number of small case studies in order to evolve and further improve certain aspects of the methodology. For instance, in an easyJet example (passenger satisfaction at airport), for the development of tool requirements to support RE within a current aircraft development programme (product family management), and in order to develop tooling interface requirements (exchange of requirements and associated V&V information).

Although not the main objective of the first applications of OntoREM, process measurements have already conducted to suggest early indications of the typical duration of individual workflows. However, there is not yet sufficient evidence available to make sound conclusions.

A major case study is planned to be conducted within Airbus in 2009 in order to apply the OntoREM approach to the wing-engineering domain and develop in a controlled environment a specific sub-set of wing requirements, for the later, critical evaluation of OntoREM as a methodology. Both the quality of the developed requirements and the efforts and time needed to develop them will be recorded and assessed in order to allow for a comparison with requirements that were developed in the context of previous aircraft programmes and also the processes that were applied to develop them.

As mentioned in the introduction of the present paper, the following aspects in RE appear to merit special attention in future research: (1) The combined consideration of products and services together in RE, (2) the focus on the systematic determination of the goals of the served system and obstacle analysis at goal level, (3) the use of domain ontologies (and semantic web technologies) above all in the problem space, and (4) the systematic inclusion of non-functional requirements (NFRs) [2]. While conducting the above Airbus case study, which aims to address all these issues, the requirements developed using the OntoREM approach will be focused on aircraft operability requirements within the wing domain for a number of reasons: *First*, non-functional requirements have been identified as being very challenging to develop. Aircraft operability requirements consist, to a large extent, of non-functional requirements, some of which can be expected to be specific to the wing domain. Others will be 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 require a focused approach, limiting investigation to a smaller area of particularly interesting and challenging requirements to be developed using OntoREM. *Third*, the data available from recent aircraft programmes will allow for a direct comparison with the aircraft operability requirements that will be developed during the wing case study. *Fourth*, while applying OntoREM using the wing aircraft operability requirements, a number of requirements elicitation methods can be used in the framework of OntoREM, for instance a goal-based approach [16, 17]. *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.

5. Conclusion

This paper has briefly reported on the application of ontology-driven requirements engineering methodology, namely OntoREM, with reference to the aerospace industry, with the objective to assess the extent to which this ontology-driven approach has the potential to develop better quality requirements in less time and at less cost, compared to traditional requirements engineering processes that were applied to develop requirements from previous aircraft development programmes. The OntoREM Metamodel has been introduced and briefly explained being the key process and task ontologies that guide the OntoREM related RE activities along with the supporting tooling environments. A number of small applications of specific aspects of the methodology have helped to evolve and further improve OntoREM, as well as deliver first process measurements. This will be followed by a major case study in 2009, which will be the basis for the critical evaluation of OntoREM. Subject to the outcomes of this critical evaluation, Airbus may conduct a larger scale pilot study within a new aircraft programme in order to decide on and prepare for a full scale deployment of OntoREM across the whole of Airbus as an integrated part of the overall product development process.

6. Acknowledgement

The authors would like to acknowledge the efforts of Younes Al-Hroub in developing the OntoRAT tool and thank him for all associated, fruitful discussions.

References

[1] Alexander, I. Stevens, R. (2002). Writing better requirements, Pearson, Harlow.

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

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

[4] Kossmann, M. Odeh, M. Gillies, A.Wong, R. (2008).Ontology-driven Requirements Engineering: Building the Meta Model, ICCTA'08, IEEE.

[5] Murman et al, E. (2002). Lean Enterprise Value, Palgrave, New York.

[6] Kossmann, M. (2006) Enhancing System Engineering (SE) deployment in large organisations by proactively managing service quality of SE training and support services, INCOSE.

[7] Gruber, T. R. (1992). A translation approach to portable ontology specifications, *Knowledge Acquisition*, 5 (2) 199-220.

[8] Saeki, M., Kaiya, H. (2006). Using Domain Ontology as Domain Knowledge for Requirements Elicitation, Proceedings of the 14th IEEE International Requirements Engineering Conference (RE'06).

[9] Gasevic, D., Djuric, D., Devedzic, V. (2006). Model Driven Architecture and Ontology Development, Springer, Berlin.

[10] Noy, N.F., McGuinness, D.L. (2000). Ontology Development 101: A Guide to Creating Your First Ontology, Stanford University.

[11] Stanford University, Protégé OWL: Getting Started, (http://protégé.stanford.edu/doc/owl/gettingstarted. html), 2006.

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

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

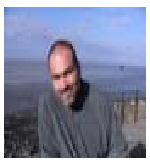
[14] www.mindjet.com, 27 October 2009.

[15] www.ibm.com, 03 December 2009.

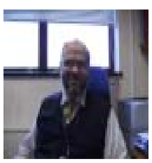
[17] van Lamsweerde, A. Letier, E (1998) .Integrating Obstacles in Goal-Driven Requirements Engineering, O-8186- 8368-6/98, IEEE

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

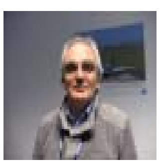
Biography



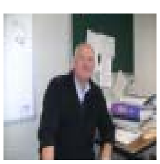
Mario Kossmann is currently completing his Ph.D. studies at the University of the West of England and works as a Systems Engineer for Airbus, having previously worked for Blohm & Voss as a Systems Engineer, Technical Manager and also Consultant in Services Marketing. He has served as an officer with 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). Currently, Mario is leading the research and application of Requirements Management 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 UWEBristol lead investigator on the OntoREM project with Airbus.



Stephen Watts is an aircraft configuration management specialist and qualified CMII Professional with a postgraduate Certificate in Management from the University of Lancaster. He has almost 40 years of industrial experience (many years of which in the aerospace and defence sectors) in a range of engineering fields such as conceptual design (electronics, electro-mechanical), work-package management and configuration management, both in specialist and leadership roles.



Andrew Gillies is senior lecturer at the University of the West of England with special interest in Systems Engineering and in particular Requirements Engineering.