Aspect-Oriented Requirements Engineering: a Roadmap

Awais Rashid
Computing Department
Lancaster University, Lancaster, LA1 4WA, UK
(01524)510316
awais@comp.lancs.ac.uk

Ruzanna Chitchyan
Computing Department
Lancaster University, Lancaster, LA1 4WA, UK
(01524)510499
rouza@comp.lancs.ac.uk

ABSTRACT
It has been five years since a Vision Paper at the Requirements Engineering Conference in 2002 laid out an initial vision for Aspect-Oriented Requirements Engineering (AORE). This vision included objectives such as offering better means to identify and manage conflicts arising due to tangled representations and identifying the mapping and influence of crosscutting requirements on architecture, design and implementation artefacts. Since then a number of AORE techniques have been proposed in literature. In this paper, we review these techniques and discuss whether and how effectively the original AORE vision from RE 2002 has been realised so far. We also discuss how the original vision needs to be extended given our improved understanding of AORE challenges and present a roadmap for the next five years as a challenge to both existing AORE researchers and those planning to pursue work in this area.

Categories and Subject Descriptors
D.2.1 [**Requirements/Specifications**]:*Languages, Methodologies (e.g., object-oriented, structured), Tools.*

General Terms
Documentation, Design

Keywords
Aspect-oriented requirements engineering; requirements analysis; requirements composition; crosscutting concerns

1. INTRODUCTION
The term Aspect-Oriented Requirements Engineering (AORE) can be traced back to the paper by Grundy at the IEEE International Symposium on Requirements Engineering in 1999 [9]. Grundy aimed to address specific issues of traditional component requirements engineering such as classification of component services per systemic areas (or aspects) of application, lack of sufficient detail per service, and ability to address a given service at the required level of detail at run-time. Aspects, therefore, were systemic characteristics for which components provided or required services, that helped to identify, categorise, and reason about the component requirements. Broader work in this area, however, was kick started by a Vision Paper[1] by Rashid, Sawyer, Moreira and Araujo at the Requirements Engineering Conference in 2002 [22]. This paper proposed a general model for AORE (demonstrated through a specific instantiation based on a viewpoints-based requirements engineering approach [21]) and laid out an initial vision for this area. Coinciding with the first workshop on Early Aspects[2], the paper also proposed an initial characterisation of the potential key contributions of AORE. Five years have passed since this original vision paper at RE 2002 that galvanised a community of researchers focusing on this topic. A number of AORE techniques, e.g., [1, 2, 3, 7, 8, 10, 11, 12, 14, 17, 18, 19, 23, 26, 27, 28], to note a few, have been proposed. In this paper, we review the AORE research space and aim to answer two key questions:

1. Given the breadth of AORE techniques available today, to what extent the original vision from RE 2002 has been realised? What are the key challenges from that vision that still remain fully or partially unaddressed?
2. Given our improved understanding of AORE challenges over the past five years, does the original vision need to be extended with new challenges that may have emerged. If yes, what are these additional challenges?

We highlight that while AORE approaches have extensively studied the issues of requirements tangling, identification of crosscutting requirements and their analysis and subsequent mapping onto the solution space, they lack strong empirical evaluation and only partially address issues relating to traceability and consistency of aspect specifications. We also discuss that research in AORE has uncovered additional challenges pertaining to composition fragility, formality of composition semantics, cognitive underpinnings of aspects and incorporation into existing software engineering processes, which must be met if AORE is to be utilised in day-to-day requirements engineering practices. Our ultimate goal is to lay out a set of research challenges as a roadmap for the next five

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
EA@ICSE'08, May 12, 2008, Leipzig, Germany.
Copyright 2008 ACM 978-1-60558-032-6/08/05...$5.00.

[1] A specific category of "vision papers" solicited for the Requirements Engineering conference in 2002.

[2] A term initially used for AORE but has since become synonymous for tools, techniques and methods utilising aspect-oriented concepts during both requirements engineering and architecture design.

years for both existing AORE researchers and those new to the area. This is not to suggest that the discussed challenges will be completely resolved within the next five years, but that we hope to see a body of work that *starts* to address these issues.

The rest of this paper is structured as follows. Section 2 summarises the AORE vision from the paper at RE 2002. Section 3 reflects on existing AORE techniques in the light of this vision and discusses whether and how effectively it has been realised. Section 4 highlights additional challenges for AORE techniques while Section 5 concludes the paper by presenting the roadmap for the next five years as a challenge to AORE researchers specifically but also to RE and AOSD researchers in general.

2. THE RE 2002 VISION

AORE techniques aim to provide modular and compositional reasoning support for crosscutting requirements, i.e., broadly-scoped functional and non-functional concerns that would otherwise be scattered across and tangled with other requirements [20, 21]. Note that it is not our aim to motivate the need for AORE in this paper. That case has been argued by several papers over the last five years. Here we only summarise the initial vision for AORE from the RE 2002 paper.

That paper noted two key objectives for AORE techniques:

- "Providing improved support for separation of crosscutting functional and non-functional properties during requirements engineering hence offering a better means to identify and manage conflicts arising due to tangled representations;
- Identifying the mapping and influence of requirements-level aspects on artefacts at later development stages hence establishing critical trade-offs before the architecture is derived."

The paper also noted a number of potential benefits of AORE:

- Ensuring consistency of stakeholders' concerns with global requirements and constraints.
- Tackling tangled requirements representations that are difficult to understand and maintain.
- Better support for modularisation hence reducing development, maintenance and evolution costs.
- Promoting traceability of broadly-scoped requirements and constraints throughout system development, maintenance and evolution.
- Support for development of systems resilient to unanticipated changes hence meeting the adaptability needs of volatile do-mains such as banking, telecommunications and e-commerce.

We next review the state-of-the-art in AORE based on the two objectives and the potential benefits noted above to discuss the extent to which the objectives have been met and the benefits demonstrated.

3. STATE-OF-THE-ART IN AORE

For our state-of-the-art analysis we have focused on a set of representative techniques from the AORE space. Our decision on whether to include a specific technique was motivated by the level of maturity of a technique. As such we have chosen to focus on techniques that have been reported in major conferences and journals as well as text books as these are usually validated in real-world scenarios. As such we have not included the significant number of workshop papers with promising ideas but a lack of validation. The techniques included in our analysis are the following:

- Scenario Modelling with Aspects [1, 27];
- Theme/Doc [2, 8];
- The AOSD-Europe Requirements Description Language (RDL) [7];
- Aspect-Oriented Use Case Approach [11];
- Modelling of Volatile Requirements with Aspects [17, 18];
- Multi-Dimensional Separation of Concerns (MDSoC) for Re-quirements [19];
- AORE with ARCADE [21];
- Cosmos [26];
- AO Goal Modelling Approach [28];
- Haley, Laney and Nuseibeh's work on Deriving Security Requirements [10].

Due to a lack of empirical evaluation for AORE techniques (as we note in our analysis below), we used our extensive qualitative comparison of AORE and other contemporary requirements engineering techniques from [6] as a starting point. We characterised each technique for its support of the two objectives and the five potential benefits on a scale of: no support, low, medium, medium to high and high support. One may argue about the subjectivity of our approximations. However, (a) our approximations are conservative rather than optimistic, (b) they are rooted in the extensive qualitative survey that we conducted and presented in [6] and (c) draw upon the results of an initial empirical study of a subset of these techniques that we have undertaken on a real-world system [24].

The results of our analysis are shown in the charts in Figs. 1 and 2 which show area plots of the coverage provided by various AORE techniques. One point that is immediately obvious, at a glance, from Figs. 1 and 2 is that no AORE technique has holistic support for all the objectives and benefits.

3.1 Support for Objectives from RE 2002 Vision

We now discuss the extent of support for the two objectives from the RE 2002 vision in current AORE techniques (as visualised in the area plots in Fig. 1).

Separation of crosscutting requirements for improved trade-off analysis

Various AORE works note this as a key contribution of AORE to the broader RE space. So it is not surprising that all techniques have extensive support for separation of crosscutting requirements. Approaches such as the AOSD-Europe RDL and AORE with ARCADE have dedicated and scalable tool support in the form of EA-Miner [23] that utilises corpus-based semantic

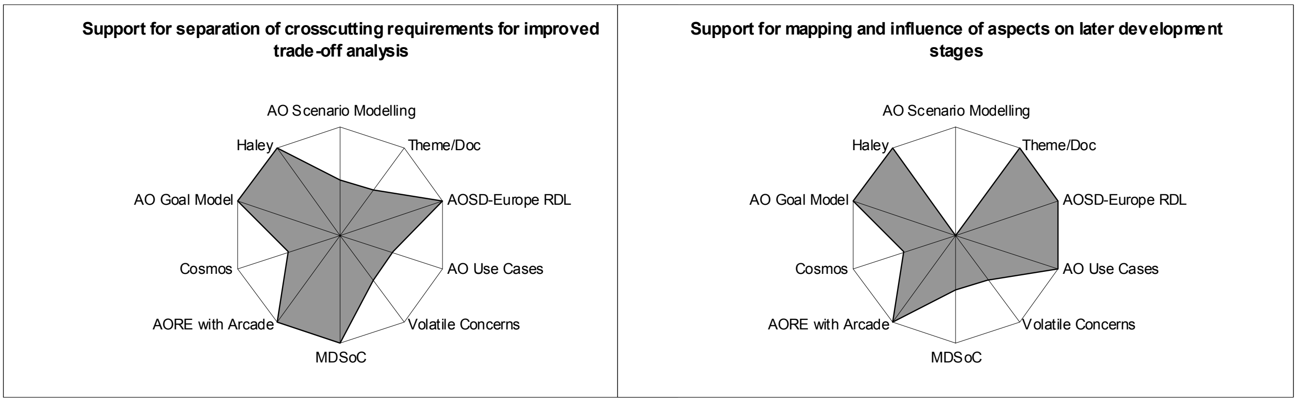

Fig. 1: Extent of support for the RE 2002 objectives in AORE techniques

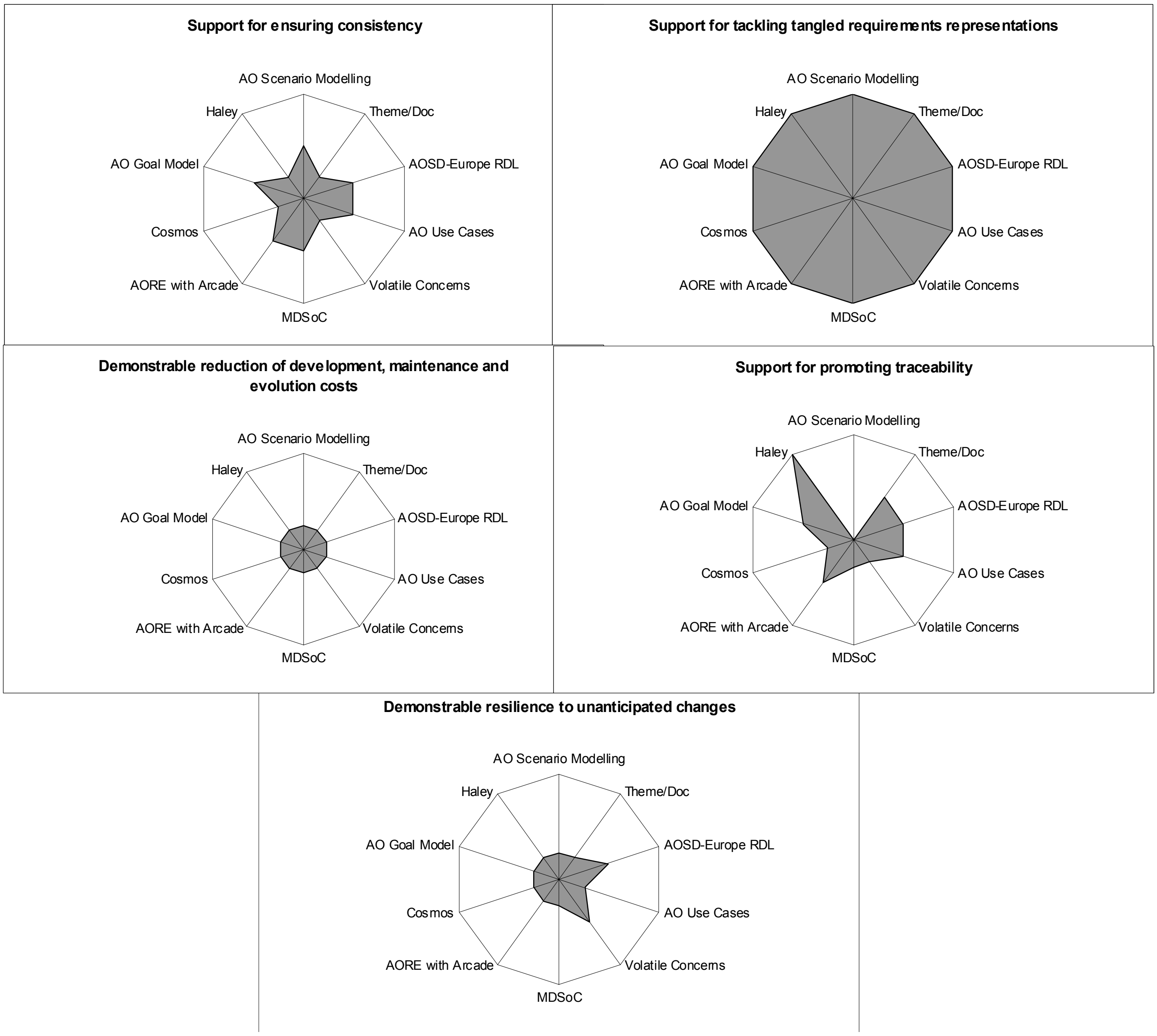

Fig. 2: Extent to which the RE 2002 vision benefits are supported or demonstrated by current AORE techniques

analysis of natural language requirements to categorise crosscutting concerns and structure the requirements specification. Similarly, Theme/Doc can be complemented by Latent Semantic Analysis (LSA) to provide similar support for aspect identification [14]. On the other hand, not all techniques then utilise this separation to reason about the mutual trade-offs between the systemic properties isolated by aspects. While approaches such as AORE with ARCADE, the MDSoC approach and the RDL (with its supporting tool MRAT) use fuzzy logic and temporal semantics as a basis to identify and reason about trade-offs, others such as Theme/Doc and AO Use Cases defer such reasoning to design and implementation and those such as AO Scenario Modelling do not provide any support at all.

Mapping and influence on later development stages

Most AORE techniques are motivated by the need to understand how aspects first manifest themselves during software development, relating the aspects to stakeholder viewpoints and understanding how the aspects are refined or impact key architectural decisions. So naturally most AORE techniques have some support for identifying such mapping and influence. The most extensive support exists in the cases of AORE with ARCADE, Theme/Doc and the RDL. AORE with ARCADE has been extended with the PROBE framework [12] to derive proof obligations that can form the basis of test cases or formal validation of the resulting implementation. In Theme/Doc the aspect themes have a direct correspondence in a Theme/UML design. A set of pattern-based guidelines exist for deriving an initial architecture from an RDL specification for further elaboration [5]. Though this initial work in this area is promising, the relationship between requirements-level aspects and their solution space counterparts needs to be studied in further depth, however, to fully understand the often intricate and subtle interplay between them [5].

3.2 Demonstration of Benefits from RE 2002 Vision

We now discuss the extent to which the potential benefits from the RE 2002 vision have been demonstrably materialised in current AORE techniques (as visualised in the area plots in Fig. 2).

Ensuring consistency

Perhaps it is not surprising that most AORE techniques have good support for the two key objectives from the RE 2002 vision. However, things are not as straight forward when it comes to demonstration of the five potential benefits. Most AORE techniques lack sufficient support for ensuring consistency of the separated aspects with other requirements and even fewer have any formal underpinning for such consistency checking. AO Scenario Modelling uses state transition semantics. AORE with ARCADE and its associated PROBE framework use first-order temporal logic. However, even in these approaches, the syntactic references utilised in the composition specifications introduce fragility in such specifications hence compromising the resilience to changes [7]. While the RDL and its supporting tool MRAT utilise semantics-based composition specifications and temporal ordering semantics extracted from the semantics of the natural language, no formal semantics for the composition are available.

Tackling tangled requirements representations

Since AORE's main motivation is to address tangling of systemic concerns with other requirements, all AORE techniques are motivated by various tangling scenarios and provide extensive support to address such tangling.

Reducing development, maintenance and evolution costs

AORE promises improved modularity of requirements specifications. Intuitively one can argue that such modularity would reduce development, maintenance and evolution costs. However, without strong empirical evidence and/or industrial sized case studies such claims cannot be effectively demonstrated. Across the whole AORE space we find such empirical or industrial evaluation almost non-existent. The only exceptions that we are aware of are the requirements-level aspect mining tools and techniques, e.g., EA-Miner and LSA for Theme/Doc, which have reported non-trivial case studies and scalability tests [14, 25], the extensive analysis of security requirements by Haley, Laney and Nuseibeh and the comparative empirical study of development efforts and precision and recall of a subset of techniques included in our analysis [24].

Promoting traceability

Traceability is a general problem in software engineering. In the early days of AORE one of the attractions behind the notion of an aspect at the requirements-level was the potential homogeneity (in terms of direct mapping to architecture, design and implementation) it could offer in an aspect-oriented development context. However, this vision was quickly shown to be an over-simplification by works, such as [12, 19, 21], that argued that requirements-level aspects do not always have a direct one-to-one mapping to aspects that a designer or implementer may utilise in a system's architecture, design and implementation. Instead some requirements-level aspects can be realised in the solution space using existing modularity mechanisms while others become drivers for key architectural choices and decisions hence becoming implicit to the system implementation and there may be others that are substituted by their refinements in architecture and design.

Nevertheless, since most AORE techniques provide support for mapping aspectual requirements onto architecture, design and implementation (as we discussed above), there is inherent support for traceability. For instance, Theme/Doc themes can be traced to implementation via Theme/UML which has mapping guidelines to derive AO and non-AO implementations [8]. The AO goal modelling approach facilitates tracing of high-level system and stakeholder goals to their operationalisations. Systemic aspects crosscutting stakeholders' viewpoints in an ARCADE specification can be traced to suitable proof obligations about the implementation. However, all these approaches mainly talk about forward traceability. Backward traceability and round tripping have received little attention. Furthermore, infrastructure support for traceability is lacking. It is not clear whether aspects introduce additional traceability issues that would need to be tackled by such infrastructure and, if so, how complex these are and whether they outweigh the benefits of inherent traceability that AORE techniques provide.

Resilience to unanticipated changes

Like development, maintenance and evolution costs, one can argue about the resilience to change of a requirements specification in any technique that aims to improve modularity. However, without appropriate empirical evidence this claim remains just a claim especially (as we discussed above) since the mapping between aspectual requirements and their solution space counter parts is not always one-to-one. The refinement relationships can be complex and even multi-dimensional in nature [5, 7, 19]. The only AORE technique specifically targeting volatile requirements, i.e. [17, 18] discusses a non-trivial case study but no empirical data from that study is available. Once again we find a serious lack of empirical evidence for this perceived benefit of AORE.

4. BEYOND THE RE 2002 VISION

Over the past five years several facets of the RE 2002 vision have been explored in significant detail. However, there are other challenges that have come to the fore that research in AORE must address if it is to make a transition from a research idea to deployment in day-to-day development practices. We now motivate and discuss four key challenges in this context.

Cognitive models underpinning the notion of an aspect

Just like in the early days of object-orientation, the question of "what is an aspect" has been frequently asked since Kiczales et al. introduced the notion of an aspect [13]. Mostly an aspect is described as a concern that cannot be modularly represented in a chosen decomposition and is hence scattered across and tangled with other stakeholder concerns. In other words, an aspect is defined in terms of the problem it should solve. Specific problems of real-time requirements [21], security specifications [10] and business rules [18], for instance, have been used to motivate the need for aspects during requirements engineering. However, if an aspect is a concept inherent to the complexity [4] of the variety of requirements analysis problems we aim to solve with its help, then we should be able to define it in terms of a concept and not just the problem that it solves. We have clear conceptual counterparts for objects, use cases, etc. The same should be the case for aspects.

The problem is that it is not yet clear how developers build mental models of crosscutting concerns for ultimate translation into an AORE specification. These mental models have a basis in a variety of sources, including but not limited to:

- natural language descriptions of requirements;
- other sources of requirements, e.g., visual representations of business processes, etc.;
- developers' own domain knowledge and problem analysis expertise.

The process of identifying software features that are appropri-ate for "aspectisation", and for distinguishing them from those that are not, requires thorough exploration of the cognitive models that underpin such aspectisation.

Software and business process improvement implications

The success of any software engineering technique significantly depends on how it integrates with existing software development and business practices. For AORE (and, in fact, AOSD as a whole) the implications for software process improvement (SPI) and business process improvement (BPI) have not been explored to date. One needs to understand: what are the SPI and BPI implications of introducing AORE into an organisation; does it provide real gains in terms of productivity and software quality and reduction in development costs; how resource intensive is such a transition to utilise AORE and whether its benefits outweigh the costs of such transition. Only when these issues have been studied in detail can AORE make a transition from research into day-to-day development practices.

Composition semantics for requirements-level aspects

One of the major perceived benefits of AORE is that it attempts to shift the focus of requirements-level reasoning from just modular reasoning to both modular and compositional reasoning [7, 20]. The composition specifications in AORE elaborate the intricate dependencies and interactions between crosscutting concerns and other requirements in the system as well as those amongst the crosscutting concerns themselves and encourage the requirements engineer to reason about these dependencies and interactions in a systematic fashion[3]. However, despite composition being central to AORE, most AORE techniques do not provide any formal composition semantics. The exception is the temporal logic specification of composition operators in PROBE [12] and the state machine semantics in AO Scenario Modelling [1, 27]. This is in contrast with other contemporary RE techniques, e.g., KAOS [15, 16], where a formal underpinning facilitates sound reasoning about the specification. This, of course, relates to the consistency issue that we mentioned above. However, ensuring consistency of an aspect with the requirements that it crosscuts is one thing and ensuring that the composition and the resulting analysis is sound is another. Suitable composition semantics can address both these issues for AORE techniques. Ideally there has to be a simple formal reference model for composition in AORE that techniques can extend as needed hence providing a vehicle for the comparative analysis of the composition capabilities of AORE techniques.

Composition fragility in AORE

Like their programming-and design-level counter-parts, AORE techniques, to date, have heavily relied on syntactic references in composition specifications. In a recent paper about the AOSD-Europe RDL [7], we have highlighted the fragility of such syntactic compositions as well as the drawbacks of reasoning about dependencies and interactions amongst concerns in terms of the syntax of a technique and then transforming that to the semantics of the problem being analysed. We proposed one specific technique for specifying aspect compositions in terms of the semantics inherent to the natural language hence allowing the requirements engineers to reason about semantic dependencies and interactions directly instead of making the significant and arduous jump from syntactic to semantic reasoning. We see our approach as a stepping stone towards more semantics-based composition specifications, not just in AORE but in AOSD in general. More extensive exploration of this space is needed if we are to avoid the accidental complexity [4] that syntactic composition

[3] Goal-oriented techniques aim to support similar compositional reasoning in terms of positive and negative contributions amongst softgoals. AORE techniques often borrow this notion of positive and negative contributions once the composition specification is available and the results of the composition reveal the various dependencies and interactions.

models introduce and instead focus on semantics-based reasoning about aspect dependencies and interactions.

5. ROADMAP FOR THE NEXT FIVE YEARS: A CHALLENGE

To conclude, we present a roadmap for AORE research for the next five years based on the above discussion. We note that these are not absolute markers but we would like to see AORE researchers take these on as a challenge and seriously attempt to address them over the next five years as this would yield significant improvement in the state-of-the-art. We also note that we are not suggesting that the other topics from the RE 2002 challenge have been fully ad-dressed and that no further research there should take place. On the contrary, we encourage the continuation of the extensive work on those topics but set additional challenges which also need immediate attention from this community of researchers. The five challenges we set the community are the following:

1. Provide empirical validation of the potential contributions of AORE clarifying its benefits in comparison with other con-temporary RE techniques and its impact in terms of productiv-ity, quality and costs as well as its SPI and BPI implications.
2. Complement the inherent support for aspect mappings with suitable traceability infrastructure from state-of-the-art in re-quirements traceability to thoroughly address aspect traceabil-ity to subsequent AO or non-AO designs and implementations. Furthermore, do not only address forward traceability of requirements-level aspects to architecture, de-sign and implementation but also backward traceability to the aspectual requirements that are sources of solution space ele-ments.
3. Clarify the cognitive models underpinning an AORE-based requirements analysis.
4. Provide a formal underpinning for AORE composition speci-fications for both consistency checking purposes and sound-ness of the analysis.
5. Address composition fragility arising from syntactic composi-tion specifications.

We hope that AORE researchers specifically, and AOSD and RE researchers in general, will take up these challenges to im-prove upon our current understanding of AORE and its implica-tions for software engineering.

ACKNOWLEDGMENTS

This work is supported by EC FP6 Grant AOSD-Europe (IST-2-004349) and UK EPSRC Grant MULDRE (EP/C003330/1). The authors thank Tom Ormerod and Linden Ball in the Psychology Dept. at Lancaster University for helpful discussions.

6. REFERENCES

[1] J. Araujo, J. Whittle, D.-K. Kim, "Modeling and Composing Scenario-Based Requirements with Aspects", Int'l Conf. Requirements Engg. (RE), 2004, IEEE CS, pp. 58-67.

[2] E. Baniassad, S. Clarke, "Theme: An Approach for Aspect-Oriented Analysis and Design", Int'l Conf. Software Engg. (ICSE), 2004, IEEE CS, pp. 158-167.

[3] E. Baniassad, P. Clements, J. Araujo, A. Moreira, A. Rashid, B. Teki-nerdogan, "Discovering Early Aspects", IEEE Software, 23(1), pp. 61-69, 2006.

[4] F. P. Brookes, "No Silver Bullet - Essence and Accident", in The Mythical Man-Month: Addison-Wesley, 1995, pp. 177-203.

[5] R. Chitchyan, M. Pinto, A. Rashid, L. Fuentes, "COMPASS: Composi-tion-Centric Mapping of Aspectual Requirements to Architecture", Transactions on Aspect-Oriented Software Development: Special Is-sue on Early Aspects, (Accepted to Appear).

[6] R. Chitchyan, A. Rashid (eds.), "Survey of Aspect-Oriented Analysis and Design", AOSD-Europe Project Deliverable: AOSD-Europe-ULANC-9 (available from: http://www.aosd-europe.net) 2005.

[7] R. Chitchyan, A. Rashid, P. Rayson, R. W. Waters, "Semantics-based Composition for Aspect-Oriented Requirements Engineering", Int'l Conf. Aspect-Oriented Software Development (AOSD), 2007, ACM.

[8] S. Clarke, E. Baniassad, Aspect-Oriented Analysis and Design: The Theme Approach: Addison-Wesley, 2005.

[9] J. Grundy, "Aspect-Oriented Requirements Engineering for Compo-nent-based Software Systems", 4th IEEE Int'l Symp. on RE, 1999, IEEE CS, pp. 84-91.

[10] C. Haley, R. Laney, B. Nuseibeh, "Deriving Security Requirements from Crosscutting Threat Descriptions", Int'l Conf. Aspect-Oriented Software Development (AOSD), ACM, 2004.

[11] I. Jacobson, P.-W. Ng, Aspect-Oriented Software Development with Use Cases: Addison-Wesley, 2004.

[12] S. Katz, A. Rashid, "From Aspectual Requirements to Proof Obliga-tions for Aspect-Oriented Systems", Int'l Conf. Requirements Engg., 2004, IEEE CS, pp. 48-57.

[13] G. Kiczales, et al., "Aspect-Oriented Programming", European Conf. on Object-Oriented Programming (ECOOP), 1997, Springer, LNCS 1241, pp. 220-242.

[14] L. K. Kit, K. C. Man, E. Baniassad, "Isolating and Relating Concerns in Requirements using Latent Semantic Analysis", ACM Conf. on Object-Oriented Programming, Languages, Systems and Applica-tions (OOPSLA), 2006, ACM.

[15] A. Lamsweerde, "Goal-Oriented Requirements Engineering: A Guided Tour", 5th Int'l Symp. Requirements Engg., 2001, IEEE CS, pp. 249-261.

[16] E. Letier, A. Lamsweerde, "Deriving Operational Software Specifica-tions from System Goals", ACM SIGSOFT Conf. Foundations of Software Engineering (FSE), 2002, ACM, pp. 119-128.

[17] A. Moreira, J. Araujo, "Handling Unanticipated Requirements Change with Aspects", Int'l Conf. Software Engg. and Knowledge Engg. (SEKE), 2004, pp. 411-415.

[18] A. Moreira, J. Araujo, J. Whittle, "Modeling Volatile Concerns as Aspects", Conference on Advanced Information Systems Engineer-ing (CAiSE), 2006.

[19] A. Moreira, A. Rashid, J. Araujo, "Multi-Dimensional Separation of Concerns in Requirements Engineering", Int'l Conf. on Requirements Engg. (RE), 2005, IEEE CS, pp. 285-296.

[20] A. Rashid, A. Moreira, "Domain Models are NOT Aspect Free", Proceedings of MoDELS/UML, 2006, Springer, LNCS 4199, pp. 155-169.

[21] A. Rashid, A. Moreira, J. Araujo, "Modularisation and Composition of Aspectual Requirements", 2nd Int'l Conf. Aspect-Oriented Soft-ware Development, 2003, ACM, pp. 11-20.

[22] A. Rashid, P. Sawyer, A. Moreira, J. Araujo, "Early Aspects: A Model for Aspect-Oriented Requirements Engineering", IEEE Joint Int'l Conf. on Requirements Engineering, 2002, IEEE CS, pp. 199-202.

[23] A. Sampaio, R. Chitchyan, A. Rashid, P. Rayson, "EA-Miner: a Tool for Automating Aspect-Oriented Requirements Identification", Int'l Conf. Automated Software Engineering (ASE), 2005, ACM, pp. 353-355.

[24] A. Sampaio, P. Greenwood, A. Garcia, A. Rashid, "A Comparative Study of Aspect-Oriented Requirements Engineering Approaches", Int'l Conf. Empirical Software Engineering and Metrics (ESEM), IEEE CS, 2007.

[25] A. Sampaio, A. Rashid, R. Chitchyan, P. Rayson, "EA-Miner: To-wards Automation in Aspect-Oriented Requirements Engineering", Transactions on Aspect-Oriented Software Development: Special Is-sue on Early Aspects, (Accepted to Appear).

[26] S. M. Sutton and I. Rouvellou, "Modeling of Software Concerns in Cosmos", International Conference on Aspect-Oriented Software Development, 2002, ACM, pp. 127-133.

[27] J. Whittle and J. Araujo, "Scenario Modelling with Aspects", IEE Proceedings - Software Special Issue, 151(4), pp. 157-172, 2004.

[28] Y. Yu, J. Leite, and J. Mylopoulos, "From Goals to Aspects: Discov-ering Aspects from Requirements Goal Models", Int'l Conf. on Re-quirements Engg. (RE), 2004, IEEE CS, pp. 38-47.