Semantic vs. Syntactic Compositions in Aspect-Oriented Requirements Engineering: an Empirical Study

Ruzanna Chitchyan, Phil Greenwood, AmericoSampaio, Awais Rashid
Lancaster University, UK
(rouza, greenwop, a.sampaio, awais)
@comp.lancs.ac.uk

Alessandro Garcia
Pontifical Catholic University of Rio de Janeiro, Brazil
afgarcia@inf.puc-rio.br

Lyrene Fernandes da Silva
State University of Rio Grande do Norte, Brazil
lyrene@gmail.com

ABSTRACT

Most current aspect composition mechanisms rely on syntactic references to the base modules or wildcard mechanisms quantifying over such syntactic references in pointcut expressions. This leads to the well-known problem of pointcut fragility. Semantics-based composition mechanisms aim to alleviate such fragility by focusing on the meaning and intention of the composition hence avoiding strong syntactic dependencies on the base modules. However, to date, there are no empirical studies validating whether semantics-based composition mechanisms are indeed more expressive and less fragile compared to their syntax-based counterparts. In this paper we present a first study comparing semantics- and syntax-based composition mechanisms in aspect-oriented requirements engineering. In our empirical study the semantics-based compositions examined were found to be indeed more expressive and less fragile. The semantics-based compositions in the study also required one to reason about composition interdependencies early on hence potentially reducing the overhead of revisions arising from later trade-off analysis and stakeholder negotiations. However, this added to the overhead of specifying the compositions themselves. Furthermore, since the semantics-based compositions considered in the study were based on natural language analysis, they required initial effort investment into lexicon building as well as strongly depended on advanced tool support to expose the natural language semantics.

Categories and Subject Descriptors

D.2.1 [**Software Engineering**]: Requirements/Specifications—*Methodologies (e.g. object-oriented, structured)*; D.2.8 [**Software Engineering**]: Metrics—*Performance Measures*

General Terms

Experimentation, Measurement

AOSD'09, March 2–6, 2009, Charlottesville, Virginia, USA.

Keywords

aspect-oriented composition specification, aspect-oriented requirements engineering, evaluation, requirements metrics

1. INTRODUCTION

The majority of current aspect-oriented (AO) composition mechanisms rely on syntactic references to enable the aspectual and base artefacts to be composed. By syntactic references we mean use of specific naming conventions and structural references (e.g., to requirements ids) or quantification over such elements using wildcards. When performing refactoring or maintenance activities this often leads to the well-documented fragile pointcut problem [8, 9] whereby a structural change in the base modules may invalidate the aspect composition specifications. Further undesirable phenomena can also occur such as ripple-effects [8]. Additionally, when using syntactic references the compositions are always constrained by the syntax of the base artefacts. As a result the developer may never be able to fully express his/her true intentions [19].

The problems associated with syntax-based composition are not just limited to AO programming languages, such as the string-based name pattern matching used in AspectJ[1], but are also rife in approaches tackling analysis and design level aspects. For example, as demonstrated in [5] most composition mechanisms in aspect-oriented requirements engineering (AORE) rely on syntactic references, such as requirement ids and use case step numbers. This has a number of negative consequences (in addition to the already mentioned problem of pointcut fragility). Firstly, the requirements compositions have to be expressed in terms of the structure of the requirements rather than their semantics. As a result, the requirements engineer's (and stakeholder's) intentionality is lost in the mapping to a syntax-governed model. This complicates subsequent requirements analysis, for instance, by forcing the analyst to conduct trade-off analysis in terms of syntactic elements. Secondly, the requirements engineer has to know ahead where the compositions will be applied and has to prepare these points by assigning ids or names to them or using specific naming conventions (in the rest of this paper such elements are referred to as scaffolding). If these points are not readily available in the requirements structure, the existing structure has to be changed before an unexpected composition can be defined.

Semantics-based composition mechanisms, e.g., [5, 10, 11, 13], aim to address these expressiveness and fragility problems of syntax-based composition mechanisms. Chitchyan et al. [5], for instance, present a semantics-based composition mechanism for AORE which

[1] http://www.aspectj.org

utilises the semantics of the natural language as the basis for composition. Similarly, Knoll and Mezini [10] outline their solution for a programming language to support writing programmes directly in the way that people think. Such mechanisms aim to support specification of compositions that:

- Require less scaffolding by relying on the meaning of the relationships to be captured by the composition rather than the structure of the base modules or specific naming conventions;
- Are stable in the face of change, i.e., less fragile, and hence unaffected due to structural or syntactic changes in the base modules;
- Are able to directly capture the composition's intention, i.e., are more expressive, hence bridging the gap between the developer's intentions and the composition specification mechanism.

However, to date, no empirical study exists that demonstrates whether semantics-based compositions indeed require less scaffolding, are less fragile and more expressive compared to their syntax-based counterparts. In this paper we present a first empirical study comparing semantics- and syntax-based composition mechanisms for AORE. Since we are primarily interested in evaluating semantics- vs. syntax-based composition mechanisms in AORE, we have chosen two representative syntax-based approaches from contemporary AORE which provide good support for composition (some prominent AORE approaches, e.g., Theme/Doc [2] were not selected as they postpone the composition to the design stage). We have selected one approach from the goal-based category - the AO Requirements Models with V-graphs approach (AOVG) [16, 17], and one from the viewpoint-based category - the AORE with Arcade approach [14]. While the AOVG approach uses some semantic elements in the compositions, such as term dictionaries (along with string-based name matching of syntactic compositions), the Arcade approach uses a well defined but purely syntax-based composition. These two approaches are evaluated against the composition support provided by a purely semantics-based approach - that of the Requirements Description Language (RDL) [4, 5]. As such, the selected approaches offer a suitable selection of semantics- to syntax-based composition techniques for this explorative study: with a purely semantics-based approach in the RDL, a partially semantics- and syntax-based approach in AOVG, and a purely syntax-based approach in Arcade. The findings of our study can be summarised as follows:

- Semantics-based compositions in the RDL need less scaffolding and are more stable and expressive compared to syntax-based compositions in Arcade and AOVG. Specifically, the scaffolding required for semantics-based RDL compositions is often detached from the particular specification document, which enables them to capture new requirements that may come within the scope of a pointcut during maintenance and evolution. Furthermore, RDL compositions rely on abstraction as a referencing mechanism hence resulting in compositions that are closer to the developer's intentions and more meaningful during subsequent analysis and reasoning.
- Specification of a new composition using the semantics-based RDL approach may require review of existing compositions to check if these are semantically interdependent with the new one. While, on the one hand, this may encourage developers to consider composition dependencies early on rather than defer them to conflict/trade-off analysis later on (and reduce the subsequent revision and negotiation overheads), on the other hand, it potentially complicates the composition specification task.
- When used for the first time in an application domain, natural language-based semantic compositions (as utilised in the RDL) may require a significant initial effort investment to prepare domain-specific lexicons and/or ontologies (which can only be partially automated). The syntax-based compositions in Arcade and AOVG do not require such investment.
- Semantics-based composition in the RDL is strongly dependent on availability of advanced tool support to expose the relevant semantics for use in compositions - a natural language processing (NLP) tool is used to expose the grammatical semantics of the nature language.

```
<Concern name="Complaint Specification">
...
  <Requirement id="2">The
    <Object>complaint</Object>is
    <Relationshup type="Rest" semantics="Maintain">
         saved</Relationship> on the
    <Object>system</Object>
  </Requirement>
...
</Concern>
```

(a) RDL concern.

```
<Composition name="TransactionalityComposition">
  <Constraint operator="apply">subject="update" and
       (relationship="commit" or relationship="
      roll back")</constraint>
  <Base operator="after">relationship="save" or
      relationship="change"</Base>
  <Outcome operator="ensure"/>
</Composition>
```

(b) RDL Composition.

Figure 1: An RDL concern and composition specification.

The rest of the paper is structured as follows. Section 2 presents the AORE approaches and the metrics suite used for the evaluation. Section 3 presents the evaluation set up and methodology. The results of the evaluation are presented in Section 4. Section 5 discusses threats to the validity of the study and how these have been managed in the study setup and execution. Section 6 discusses related work co-relating our findings with other relevant studies. Section 7 concludes the paper.

2. ELEMENTS OF EVALUATION

This section provides an overview of the three approaches evaluated in our study. The approaches are evaluated using a metrics suite dedicated to evaluation of composition support. This metrics suite is also discussed.

2.1 Overview of AORE Approaches Used

2.1.1 RDL

The RDL is a symmetric approach, i.e., all concerns, whether aspects or base, are treated uniformly using the same abstraction, that of a concern. As shown in Fig. 1(a), this approach annotates the natural language requirements with additional information on

their grammatical and/or semantic properties. Grammatical properties are related to the grammatical functions of the words, the main ones being:

- Subject: the entity that performs the main action of the sentence, or its main theme;
- Verb: the main activity (e.g., save) or property of the sentence (e.g., "is safe");
- Object: the entities(s) most affected by the activity in the sentence, or with respect to which the activity is realised.

The semantic properties are related to grouping of words on the basis of synonymy (e.g., complaint, grievance) or type. For instance, verb types are based on the notion of a set of participatory roles engaged in the given activities (e.g., both "Administrator saved the complaint" and "Susan stored the apples" imply that someone (Administrator, or Susan) playing the Causer role puts into resting (save, store) some Resting Thing role (complaint, apples), etc. Such grammatical and semantic annotations of the requirements text in the RDL are provided via a general purpose NLP tool, Wmatrix[2].

As shown in Fig. 1(b), an RDL composition consists of 3 parts: Constraint, Base, and Outcome. Each of these parts has a semantic query (i.e., pointcut) expressed in terms of the natural language words and their properties. These queries select requirements (i.e., joinpoints) from all across the specification document without reference to any structural information, such as requirement id or string-based name matching. For instance, the query: relationship="save" or relationship="change" in the Base element of the composition in Fig. 1(b) will select the requirement shown in 1(a) stating that "Complaint is saved on the system"; the save verb will match the saved verb of this requirement. Similarly, if there were any other requirements either directly or via synonymy referring to save, they would also be selected by this pointcut.

For each composition Constraint query selects some requirements which crosscut the requirements selected by the Base element's query. The Outcome element may select some requirements which should be checked as post-conditions though in some cases (as in Fig. 1(b)) the outcome may have an empty query (if no post-condition needs to be checked). The details of the RDL are presented in [4, 5].

2.1.2 Aracade

In AORE with Arcade requirements are modularised into viewpoints and aspects, where aspects encapsulate requirements that crosscut the viewpoint decomposition. Each of the viewpoints and aspects has a unique name and encompasses a set of requirements and sub-requirements. Each requirement has a unique identification number within the scope of its enclosing viewpoint or aspect (Fig. 2(a)).The corresponding Arcade composition is shown in Fig. 2(b). It states that the statement expressed in requirement with id=1 specified in some Transactionality aspect (not shown in Fig. 2) should be provided for all the requirements (id= "all") and sub-requirements (children="include" statement in Fig. 2(b)) of the Employee, Citizen, and Complaint viewpoints. Fig. 2(b) demonstrates that, in an Arcade composition, requirements and aspects are referenced via their unique names and ids within their defining scopes.

Thus, here the requirements are prepared for composition by initially structuring them in such a way that any given requirements statement of interest for composition is given a separate id. This is the additional scaffolding needed for composition in this approach. The pointcuts are defined by enumerating the unique "string-name and requirement id" pairs and some wildcards, such as "all" - all of which are syntactic references. A change in the concern name or id of a requirement will invalidate the compositions - this fragility arises due to structure dependence.

[2]http://www.comp.lancs.ac.uk/ucrel/wmatrix/

```
1  <Viewpoint name="Complaint">
2    <Requirement id="4"> In the event of a complaint
         being made, it will be registered on the
         system and addresses by a specific
         department.
3      <Requirement id="4.1"> This department will be
           able to handle the complain in an
           appropriate manner and return a response
           when the complaint has been dealt with.</
           Requirement>
4      <Requirement id="4.2"> This response will be
           registereed on the system and available to
            be queried.</Requirement>
5    </Requirement>
6  </Viewpoint>
```

(a) Arcade Viewpoint.

```
1  <Composition>
2    <Requirement aspect="Transactionality" id="1">
3      <Constraint action="provide" operator="for">
4        <Requirement viewpoint="Employee" id="all"
             children="include"/>
5        <Requirement viewpoint="Citizen" id="all"
             children="include"/>
6        <Requirement viewpoint="Complaint" id="all"
             children="include"/>
7      </Constraint>
8  ...
9  </Composition>
```

(b) Arcade Composition.

Figure 2: An Arcade viewpoint and composition specification.

2.1.3 AOV-Graph

The AOVG [16, 17] approach proposes that aspects can be identified during goal-oriented requirements analysis, from the integrated Goal and Softgoal Interdependency Graphs (G/SIG), as goals/tasks which contribute to more than one other goal.

As in all goal-based approaches, each goal has a type and a topic. The type reflects the generic functional/non-functional requirement, while the topic captures the contextual information of the goal. AOVG focuses on representing crosscutting relationships and defines AspectJ-motivated constructs for pointcuts, advice and intertype declarations, as well as a construct for source which defines which goal/softgoal the advice and intertype declarations initially belong to. An example of this is shown in Fig. 3.

Composition is carried out by matching the types and topics of the goals/tasks. The matching is purely name-based. The types and topics must be manually prepared during SIG development - this is the scaffolding required for composition in this approach. The pointcut specification is based on the Type[topic] enumeration, and use of some wildcards (e.g., Type.* selects all goals/-tasks of the given type). As shown in Fig. 3, goals/tasks such as Specify[complaint], Register[health unit], are selected through their names by a pointcut called P15.2.1. Advice Commit and Roll back from the Transactionality goal (source) are to be applied around the goals/tasks selected by the P15.2.1 pointcut. This is an example of the use of syntactic elements in the AOVG compositions which lead to fragility.

```
Source: Transactionality
  pointcut P15.2.1: include(Specifiy[complaint]) and
       include(Update[complaint]) and include(Register[
       health unit]) and include(Register[speciality])
       and include(Register[employee]) and include(
       Register[disease]) and include(Register[symptom])
       and inlcude(Request[sanitary license]) and include
       (Update[state of the sanitary license])
  pointcut P15.2.2: include(Detect[persistence expception
       ] and include(Detect[robustness exception])
  advice around: P15.2.1
    Commit
    Roll back
  advice after: P15.2.2
    Roll back
```

Figure 3: Composition in AOVG

In addition, this approach may use dictionaries and topic parameterisation. Dictionaries describe the concepts used in a given project. Parameterised topics allow reference to a number of topics assigned to a given parameter. For instance, if a parameter data is defined as referring to complaint and certificate by using Register[data] pointcut, one would refer to both Register[complaint] and Register[certificate] goals. This kind of referencing shows the elements of semantic referencing used in AOVG.

2.2 Metrics Suite

In order to evaluate the composition mechanisms of the above discussed AORE approaches we needed an appropriate metrics suite. However, to the best of our knowledge, no such metrics suite for comparison of semantics- and syntax-based composition mechanisms exists. Consequently, we had to be develop such a suite before the evaluation could commence. In order to ensure that this metric suite had a sense of validity, it was necessary to examine existing metric suites to draw inspiration from these and base our suite on measures that have previously been accepted by the software engineering community. Furthermore, concepts that have been successfully assessed in later development stages were considered for their applicability in the requirements engineering stage and in the context of this study. The proposed metrics suite is summarised in Table 1. Next we discuss the metrics and the foundations on which they are based.

2.2.1 Scaffolding

Scaffolding refers to structural preparation of base modules (albeit without any direct references to the aspects) for composition, for instance, through use of specific naming conventions or breaking up compound requirements into primitive ones so that their identifiers may be referenced individually. To evaluate the scaffolding required by AORE approaches, we have proposed two metrics: number of scaffolding elements prepared and the mobility index of scaffolding elements.

The number of scaffolding elements prepared measures how many elements need to be introduced/modified to enable a given composition. The higher the number the more the preparation and effort required. For instance, in order to be able to write the composition shown in Fig. 2(b) for Arcade, the particular relevant requirement of the Transactionality concern has to be isolated and given an id=1.

It should be noted, that certain elements such as the verb group types or subject - verb - object grammatical functions of the RDL are used in composition definitions. They are, however, not counted in this metric as they do not require any additional preparation: they are part of the tool-supported environment and are always annotated in the same way and do not require any human involvement. However, the lexicon/dictionary entries are counted, even though they may be a part of the tool, as they also need manual updates if previously unused project-specific terms are introduced.

The mobility index indicates what proportion of the defined scaffolding elements possesses independent semantics and so are reusable. To evaluate this we use the "What vs. Where" principle: if the element can be moved from its present location and still be meaningful, it is considered mobile. For instance, the concern name is a mobile element, since (as shown by the NFR framework [6]) a named concern can be recognised in many different requirements documents. Thus, a reference to a Transactionality concern used in Figs. 1 - 3 carries a definite meaning of coordinated storage or discarding of some changes. Conversely, the requirement id is an immobile element. For instance, in Fig. 2(b), moving the "id=1" from the context that relates it to Transactionality, makes it devoid of its intended meaning. Wildcards are also immobile as they do not have any definite quantification semantics, but quantify over the elements of the specific requirements document. In general, the closer a mobility index is to 1, the larger the proportion of the scaffolding elements that are semantically motivated.

2.2.2 Stability

While the scaffolding metrics focus on preparatory changes that may be required to realise compositions, the stability metrics, adapted from [8], are used to determine the overall effort an approach requires to implement a change. This type of metric has been successfully used to assess the stability of implementation related software artefacts as demonstrated in [8]. In addition, all these measures are generally derived from the extensively applied coupling and cohesion [3, 20] metrics. For example, Chidamber and Kemerer's coupling metric measures the number of references from one component to other external components. Our stability metric measures the stability of this coupling when applied to compositions. Cohesiveness can be inferred from the scope of the changes that have to be made. For example, if changes are generally localised to a single concern or requirement one can infer that these elements are relatively cohesive. Of course, these stability metrics are not specific to RE approaches, but they provide a better sense of the effort involved to implement a particular change with a specific composition approach rather than measuring the time spent implementing a change due to the differing levels of tool support.

2.2.3 Expressiveness

Ostermann et. al. [13] introduce the notion of expressiveness in terms of abstraction, precision, and robustness which inspired our composition expressiveness metrics. We propose two metrics related to abstraction and precision: the number of information elements used in compositions and the reachability of an information element, and a third, remembrance, related to robustness.

The number of information elements used in compositions metric measures the abstractness of references. For instance, we can "reference" each item in the following set <0, 1, 2, 3, 4, 5, 6, 7, 8, 9> using its concrete name as: "zero, one, ... nine"; or we can describe them all in a more abstract way as "decimal radix". In this example the 1st referencing alternative uses 10 information items (zero,... nine), while the 2nd uses only one (decimal radix). Similarly, in the compositions, we can enumerate each item via its direct reference (e.g., by listing all goals and topics, such as Register[health unit], Register[speciality] shown in Fig. 3) or abstract over such references (e.g., by using the Register[data] pointcut where data is defined as a parameter for health unit, speciality). Of course, when using abstractions, one must ensure that they al-

Attribute	Metric	Description
Scaffolding	Number of scaffolding elements	Measures how many elements are introduced/modified in the requirements documents to enable a given composition.
	Mobility Index	The ratio of scaffolding elements that are context independent and can be reused.
Stability	Added/Changed/Removed Compositions	The number of compositions that are altered during a maintenance change.
	Added/Changed/Removed Concerns	The number of concerns that are altered during a maintenance change.
	Added/Changed/Removed Compositions	The number of compositions that are altered during a maintenance change.
	Added/Changed/Removed Requirements	The number of requirements that are altered during a maintenance change.
Expresiveness	Number of information items	Measures the abstractness of the compositions by counting the number of elements that make up the composition definition.
	Reachability	The ratio between the number of elements a composition identifies and the number of information items used in the composition.
	Remembrance	Measures the number of correct elements identified by existing compositions after a maintenance change.

Table 1: Metric suite summary.

ways refer to the same (possibly open) set of elements; thus, it is not acceptable to define data as a parameter for health unit and speciality in one composition, and, in the same specification, use data as a parameter that excludes speciality in another composition.

The metric for the reachability of an information element is calculated as the ratio of the number of references reached by a composition over the number of information elements used in a composition. This metric evaluates how many intended elements are selected by each information item in the composition. Note that this metric does not advocate wide use of wildcards, as these tend to over-generalise and so reach the unintended items. The higher the reachability indicator, the more abstract and expressive each information item is. For instance, one information item "decimal radix" reaches ten intended elements (0 to 9), and has a reachability of 10/1=10. In contrast, when each number was individually listed ten information elements were used, and reached ten intended items resulting in a reachability of 10/10=1.

Furthermore, it is useful to differentiate between crosscutting and base reachability in addition to calculating the average reachability. This distinction allows us to investigate if the pointcuts defined for selection of crosscutting elements (e.g., the queries in the Constraint part of the RDL in Fig. 1(b)) have any distinguishing properties compared to the pointcuts defined for selection of base elements (e.g., the query in the Base part of the RDL in Fig 1(b)).

The final metric used to measure expressiveness is Remembrance related to the robustness property of Ostermann et. al. [13]. This metric evaluates the number of references correctly picked up by existing pointcuts defined in a composition when the requirements are evolved. Essentially, it evaluates the "open-endedness" [12] of the composition, assessing how well a composition will accommodate the elements that could potentially be added or changed. A composition that can sufficiently abstractly define its intentions should be able to discriminate in favour of selecting newly added requirements which fit that intention. For example, if a new requirement related to "storing data in Oracle..." is added to a set of requirements, the RDL composition shown in Fig. 1(b) will demonstrate remembrance by including this newly added requirement into the joinpoints selected by its Base query, because the synonym of the save verb (store) appears in this new requirement. Wildcards too may demonstrate some remembrance (e.g., the composition of Fig. 2(b) will select a requirement added to the Complaint viewpoint due to "id=all"). A composition with concrete references to existing requirements will have a null remembrance. This metric is similar to the traditional precision and recall measures. However, the precision and recall metrics allow "incomplete" precision and recall (i.e., they allow some incorrect references to be included, or some correct ones to be omitted). Thus, they cannot be meaningfully used for syntactic compositions which use direct pointers to relevant elements. Instead our remembrance metric evaluates the number of references correctly picked up by existing pointcuts upon changes in the requirements.

3. EXPERIMENTAL SETTINGS

This section highlights the various decisions made to set up the study to evaluate whether semantics-based compositions deliver the perceived benefits of lower scaffolding, higher stability, and better expressiveness in comparison with syntax-based compositions[3].

3.1 Case-Study Selection

The first major decision was selecting the case-study that would be the target of our investigation. The system chosen is a typical web-based information system called Health Watcher (HW) [18]. HW is a public health monitoring and complaint registration system developed and presently used in Brazil. The system allows citizens to report complaints, and query information on diseases, health service units, and previously made complaints. This system was selected because it met a number of key criteria relevant to this study. Firstly, HW is a real and non-trivial system and so enables credible conclusions to be drawn. Secondly, HW is rich in both non-crosscutting and crosscutting concerns. This provides a variety of compositions hence enabling a broad investigation to be conducted. Thirdly, the HW system has been used in a variety of other empirical studies [8, 15]. This facilitates co-relation of the results of this study with previous studies (Section 6). Finally, the original requirements are represented as use-cases, which are publicly available. Furthermore, the RDL-, AO goal- and AO viewpoint-based decompositions used in this study were derived from the original use-cases document prior to the inception of the present study. This goes some way to reduce the bias that could have been induced due to the specific objectives and focus of this study. Each derived document was developed by a specialist in the given approach. Thus, 3 specialists were involved in document preparation, change realisation, and data collection. Three additional participants took part in experiment preparation and data interpretation.

3.2 Study Setup

There were a total of 6 participants involved in the study. One participant, *the study designer*, was dedicated to the design of the study and alignment between the artefacts from the three approaches.

[3]complete study materials are available from http://www.comp.lancs.ac.uk/ greenwop/aosd09Evaluation

The study designer was an expert in HealthWatcher but had no vested interest in any of the approaches. For each AORE approach, *the tester* was a specialist in the approach (in case of RDL and AOVG, the developers of the approach, and in case of Arcade a senior RA developing tooling for it). The three testers were involved in document preparation, change realisation and data collection. Two additional participants took part in experiment preparation and data interpretation. The scenarios were executed once for each approach by its specialist tester. Repetition of the scenarios with same testers would not have provided further insightful results about the experiment due to absence of any run-time factors (e.g., network latency, etc.). On the contrary, such repetition would have introduced a bias due to the familiarity of the specialist tester with HealthWatcher due to the previous iterations.

3.3 Study Phases

The study was divided into three key phases: (1) the alignment of each requirement document, (2) applying a series of maintenance changes to each requirement document, and (3) the assessment of the changes made in phase 2 using the metrics detailed in Section 2.2.

3.3.1 Alignment Procedure

The initial set of requirement documents derived from HW's use-cases had to be aligned to ensure that all the requirement documents were largely equivalent to each other. First, the requirements documents for RDL, AOVG, and Arcade were compared against the original use-case document. This was to ensure that all the main requirements listed in the use-case document were covered in the derived requirements documents. Next the level of detail preserved in each of the derived documents was analysed. This was to address the contrasting level of detail that was present in each document and the varying degree of direct correspondence to the original use-case specification.

The next phase in the alignment process involved making adjustments to the derived documents. This mainly involved excluding certain requirements from the documentation and establishing name correspondence between concerns and requirements of different documents.

During alignment the study designer discussed changes to documents with each testers individually and obtained his/her agreement. This ensured that the study designer did not misinterpret any of the document elements specific to an AORE approach. If the study designer were to prepare the documents for each AORE approach, this would require him/her to be proficient in each approach at the same level as the testers. In practice, this would be hard to achieve without introducing bias.

After performing these adjustments we were able to establish a high level of equivalence between the concerns and requirements of the three derived documents and the original use cases. This was verified by ensuring that similar high level concerns and requirements were covered in all documents and identifying the goals/subgoals/tasks from AOVG which corresponded to the concerns/requirements of the other documents. It must be noted that we did not intend for each approach to represent the same elements as concerns, only that all the relevant content from the original use-case document was equally represented in each of the derived documents. It is quite natural that the concern-level elements differ in each representation as each approach utilises its own perspective on modularising concerns. Evaluation of "good vs. bad" modularisation of these approaches is not the subject of the present study. Instead we accept the modularisation structure of each given document and focus on assessing its compositional properties.

Description
Scenario 1: Add the Transactionality concern with its respective requirements and compositions to the HealthWatcher requirements document. This is a perfective change intended to ensure consistency preservation for the data in the database.
Scenario 2: Add the information about Repeating Communication Attempts to the HealthWatcher requirements document. This is a perfective change aimed at improving system usability.
Scenario 3: Update the requirements document with a new requirement which ensures that a complaint can be updated only if the complaint status has not been set to CLOSED. This is a corrective change intended to enforce a previously known but not enforced issue.
Scenario 4: Add a concern which allows the user to request a sanitary licence certificate via the HealthWatcher system. This is an adaptive change, intended to extend the services provided by the system to meet newly emerging requirements.
Scenario 5: Remove a requirement about the need to use a secure protocol with the HealthWatcher system. This is a (hypothetical) adaptive change intended to remove a requirement which has become redundant.

Table 2: Summaries of the change scenarios applied.

3.3.2 Change Scenarios

The second phase of the study involved applying a series of five maintenance scenarios to each of the derived requirements documents. The scenarios were explicitly designed to assess the scaffolding, stability, and expressiveness of the three AORE approaches. However, it was also necessary for each scenario to have an element of realism and so the original HW developers were consulted when drawing up the scenarios to determine whether the proposed changes were valid. The scenarios are summarised in Table 2.

Participants were instructed to realise the scenarios independently using best practices for their particular approach. More specifically, the participants were instructed to realise the scenarios one by one and use the output from the previously completed scenario as an input for the next one, progressively evolving the initial requirements document through the whole set of scenarios.

Once each of the maintenance scenarios had been applied, it was necessary to again re-apply the alignment procedures outlined earlier. Furthermore, it was necessary to harmonise each approach with regards to the metrics suite to ensure the metrics were accurately calculated for each approach. For instance, it was necessary to decide what constituted a concern in AOVG. Upon document comparison and discussion with the AOVG author, it was decided that 3rd level goals in the AOVG approach mapped naturally to concerns and any sub-goals mapped to requirements.

4. RESULTS AND ANALYSIS

Having presented the case study and the metrics suite used for the study, we now present the data and its analysis.

4.1 Scaffolding should be based on Semantically Motivated Mobile Elements

As shown in Fig. 4, the RDL approach requires consistently fewer scaffolding elements to be added (except for scenario 2) while AOVG requires the highest. In scenario 2 the figure for the RDL is slightly higher as it introduces a relatively large set of terms into the lexicon, in particular definitions for specific error types. This scenario also illustrates the problem of initial lexicon development for RDL compositions. Though this issue is not a direct focus of this study, which focuses on evolution and change support, it is a prerequisite for successful use of the RDL for specific domains. The

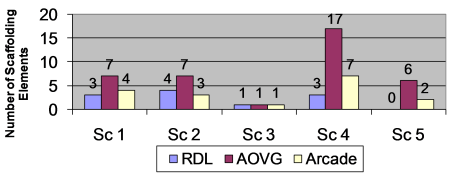

Figure 4: Number of scaffolding elements per scenario.

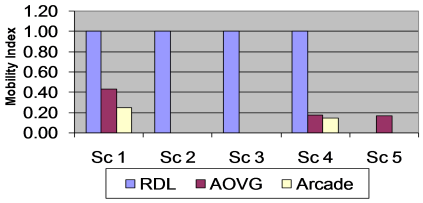

Figure 5: Mobility index of scaffolding elements per scenario.

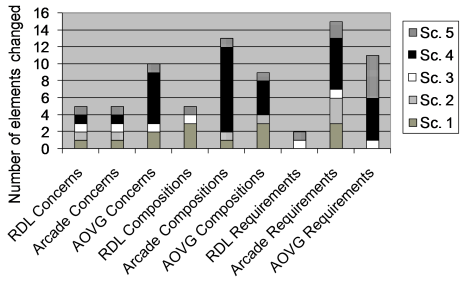

Figure 6: Total number of elements affected due to change.

first-time development of the lexicon for a given domain can be a substantial task, as it requires (semi-automatic) processing of a set of domain-related documents and identification of domain-specific synonymous concepts and their relationships. Such an effort investment is often justified by repeated reuse of the lexicon or assisted long-term maintenance of a specific application. The higher indicator for AOVG in scenario 4 is due to extensive changes to the named requirements.

The mobility index for each approach is illustrated in Fig. 5. As all the RDL scaffolding elements are semantically motivated, the index for all scenarios where RDL adds scaffolding elements is 1.

In scenarios 2 and 3 both AOVG and Arcade have a mobility index of 0, indicating that none of these scaffolding elements has an independent meaning outside of the specific setting of the given concern/document. In scenarios 1 and 4 both AOVG and Arcade have used named sub-concerns and AOVG has also used a dictionary entry, all of which are considered to be reusable elements resulting in a mobility index well below 1. In addition, AOVG has defined a sub-concern for scenario 5 hence a mobility index of 0.17 compared to a mobility index of 0 for Arcade which employs semantically unmeaningful requirements ids as scaffolding elements for this scenario. Note that the RDL has a mobility index of null as no scaffolding had to be used for scenario 5 (this should not be confused with poor mobility, but indicates absence of relevant data). This does not imply that the scaffolding elements are immobile.

In summary, this analysis leads us to conclude that all the considered approaches require some scaffolding. However, some types of the scaffolding elements have self-contained meaning, and so can be reused in other locations. For instance, a lexicon entry for a specialised term can be reused in other requirements documents of a similar domain, whereas a requirement id, when removed from its location completely loses its meaning.

While it is desirable to minimise all types of scaffolding elements, we consider that semantically motivated mobile elements are necessary for the meaningful interpretation of the requirements and their relationships, and so should be maintained. The immobile elements, on the other hand, are unnecessary. For instance, the lexicon entries used in the RDL are essential to convey the specific terms used in a project domain, or to demonstrate the relations between such elements. Conversely, requirement ids have no inherent contribution to the domain description; they will often be changed due to addition/removal of new requirements, thus this is only artificial and transient information about the requirements and should not be expected to be known to the analysts or domain experts. While such information could be produced and maintained by tools for internal processing, ideally it should remain hidden from the human users. The metrics for the number of scaffolding elements introduced, and the mobility index are complementary, as a higher number of elements does not always indicate excessive scaffolding.

4.2 Stability Analysis

The total number of elements affected by the changes of all 5 scenarios is shown in Fig. 6. Next we discuss in turn the metrics for concern, composition and requirement stability.

4.2.1 Can Semantics-based RDL Compositions Localise Changes?

From Fig. 6 we observe that the RDL and Arcade approaches have equal number of affected concerns: one per scenario, while AOVG has twice that number of affected concerns. At a glance this may suggest that RDL and Arcade are compatible in concern stability measure. However, these approaches differ in the nature of effects: RDL had added 3 new concerns and modified 2, while Arcade had modified 5 concerns - one per scenario.

AOVG performs worst for this metric, with 5 added concerns and 5 modified. Here the largest set of affected concerns belongs to scenario 4 (3 added and 3 modified concerns). These modifications are due to the need to rename and re-structure concerns and their requirements to ensure the correctness of compositions. The number of additions is somewhat dependent on the previous concern granularity choice made during the alignment procedure set out in Section 3 (i.e,. that 3rd level goals in the AOVG approach mapped to concerns). However, for this metric, if higher-level goals (i.e. 2nd level goals) were mapped to concerns instead, then this number of additions would be lower. A cut-off point had to be selected and we found that generally 3rd level goals performed well as concerns throughout the study.

Thus, these results are encouraging for semantics-based composition, as RDL used the least invasive changes, by directly modifying only 2 concerns. Yet, Arcade - the purely syntactic approach - also performed comparatively well in localisation of changes to concerns. Further detailed analysis is performed in subsequent sub-sections to accurately determine the effects of semantics-based composition on stability.

4.2.2 Semantics-based RDL Compositions are More Stable

The added RDL and Arcade compositions are focused on adding relationships between concerns/requirements. The changed compositions for these two approaches focus on adding/removing references to existing or additional relationships. AOVG, on the other hand, also creates new requirements via its intertype declarations.

As shown in Fig. 6, the RDL approach has the smallest number of affected compositions (2 compositions added and 2 changed). However, in scenario 1 this approach has 3 compositions, comparable to 3 compositions by AOVG and more than the 1 by Arcade. This is because when adding a transactionality relationship (i.e., composition), the closely semantically related relationships on data storage and data consistency checks are also reviewed. Such semantically motivated composition review may complicate the composition specification task for the RDL. On the other hand, such related relationship review is left for later conflict/trade-off analysis in AOVG and Arcade. This later exploration of composition interdependencies may lead to overheads in terms of revision and negotiation subsequent to the trade-off analysis. Conversely, scenarios 2, 4, and 5 for the RDL do not require any new composition definition or change, as the existing semantic relationships accommodate the changes.

For Arcade scenario 4 creates new requirements that cannot be accessed via existing id= "all" wildcard based references, as selective participation of joinpoints is needed. This causes a large number of compositions to be changed (10 in total).

Scenario 4 also has a noticeable effect on AOVG compositions, causing not only addition of new joinpoints, but also removal of joinpoints from existing compositions due to re-named requirements. This effect on AOVG compositions is explained by the need to review the goal naming conventions due to addition of a large number of new goals.

4.2.3 Syntactic Compositions in Arcade and AOVG Affect Requirement Locations

The RDL has a relatively small number of affected requirements: total of 2 (see Fig. 6). A requirement is added to check the "CLOSED" state of complaint required by scenario 2 and the secure protocol requirement is removed for scenario 5. The rest of the changes were realised through concern addition. On the other hand, Arcade has quite a large number of affected requirements as it used requirements and not full concern addition for every scenario (13 additions, 1 removal, 1 change).

A significant number of requirement modifications occur also in AOVG (1 added, 7 removed, and 3 changed). AOVG removes the highest number of requirements in scenario 5 where secure protocol requirement (represented by Cryptography sub-concern and a number of its requirements) is removed. Here modifications mainly relate to requirement re-naming, as the adopted naming convention had to be updated.

Thus, we observe that the semantics-based RDL approach uses invasive changes, such as directly modifying requirements in the concerns only when this is explicitly dictated by the nature of change, relying on purely additive changes otherwise (e.g., adding full concerns). This is because in the RDL the relevant compositions do not expect the related requirements to be in a specific module. The syntactic compositions in Arcade and AOVG, on the other hand, force more invasive changes (like inserting requirements into a specific concern, as done by Arcade) since their wildcard-based compositions (e.g., viewpoint = "Complain" id= "all") will not be applicable if a relevant requirement is located elsewhere, e.g., added via a new concern.

	No. of info. items per scenario	**No. of crosscutting info. items per scenario**	**No. of base inf. items per scenario**
RDL	2.6	2.6	0
AOVG	7.5	3.5	5.3
Arcade	34.5	2.6	36.2

Table 3: Standard deviation values.

4.2.4 Types of Changes

All three approaches are similar in the addition/removal of concerns and compositions. In addition, Arcade has used id re-numbering in concerns followed by the corresponding id updates in compositions. This is because its id-based referencing in composition specifications becomes invalid when id-structure in the requirements document is changed due to addition/removal/move of requirements.

The AOVG approach has used re-naming, and merging changes along with id-replacement in compositions. This was necessary for larger changes, for instance, as in scenario 4, when a large number of new goals and tasks were added. It became apparent that the previously used naming conventions were inadequate and so needed to be reviewed.

The RDL approach, on the other hand, was able to accommodate all changes without any additional types of change, as it has no id or name-based structural dependencies.

4.2.5 Summary

As shown in Fig. 6, the RDL approach has the lowest number of concerns, compositions, and requirements affected when realising all change scenarios. This is due to the RDL composition mechanism being decoupled from the structure of the requirements document: since this structure is not relied upon, any changes made to it do not propagate to the compositions.

AOVG is the next best from the stability perspective. Since this approach uses pre-defined naming conventions and composition patterns, the identities of the elements referenced in the compositions are relatively stable. Problems arise when naming patterns need to be reviewed. Of course, the naming pattern definition and preservation effort, as part of requirements preparation, should also be kept in mind.

Finally, the pure syntactic approach - Arcade - trails with the least stable compositions. Here any change to the requirements structure must be validated in compositions to check if the wildcard matches are still correct after the changes and if the id-based referencing needs to be updated.

4.3 Expressiveness Analysis

4.3.1 Compositions should Rely on Abstractions

From Fig. 7, we can see that the RDL has the lowest number of (combined base and crosscutting) data items defined per composition. The Arcade approach, on the other hand, has the highest number of information items used. This is not surprising, as the RDL composition mechanism relies mostly on word grammatical functions, types and word groups. AOVG relies mainly on named requirements/concerns and occasional dictionary entries or wildcards. Arcade, on the other hand, relies only on concern name and requirement id pairs, with significant use of wildcards. Thus, on average, RDL relies on some form of abstraction, AOVG on named elements, and Arcade on named element and id pairs.

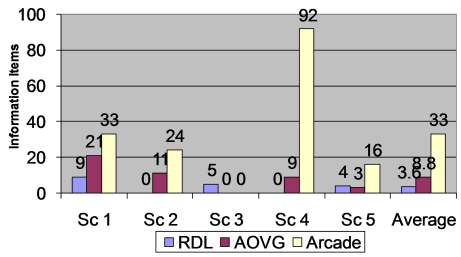

Figure 7: Number of data items defined per scenario.

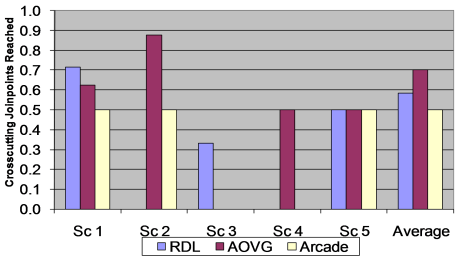

Figure 8: Number of data items defined per scenario.

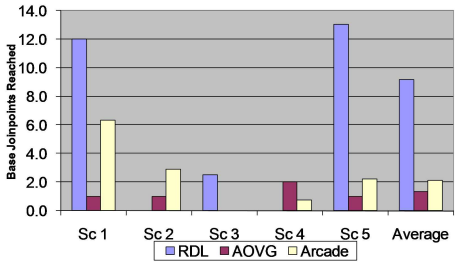

Figure 9: Base element reachability.

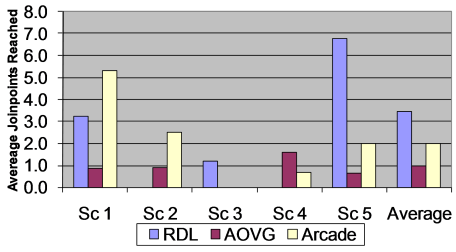

Figure 10: Average element reachability.

Looking at the data on standard deviation for this metric shown in Table 3, the RDL compositions in this experiment also have the lowest standard deviation - 2.6, compared to that of 7.5 for AOVG and 34.5 for the Arcade. Such a large deviation value for Arcade is explained by its frequent use of wildcards: in scenario 4 it was not possible to use a wildcard, and individual listing of concern-requirement id pairs had to be used. If this scenario was left out of the analysis, the standard deviation for Arcade would drop from 34.5 to 8.5. Indeed, scenario 4 for Arcade demonstrates that untamed use of wildcards can actually deject the abstracting power of a composition mechanism rather than foster it.

When comparing the standard deviations of the number of crosscutting information items with that of the base information items it emerges that for the AOVG and Arcade approaches the standard deviation of base items is significantly higher than that of the crosscutting items. This is again a consequence of the use of wildcards (as mentioned above).

4.3.2 *Reachability of Semantic Compositions in RDL Depends on Reachability of Lexicon Entries*

When analysing the reachability metrics for crosscutting (Fig. 8) and base (Fig. 9) concerns it is clear that the crosscutting item reachability is noticeably lower for all three approaches, than the base item reachability. This is due to items for crosscutting elements in each approach targeting a smaller set of specific requirements which have a broad (i.e., crosscutting) influence on a larger set of other (i.e., base) elements. Since it is natural that there are fewer crosscutting elements, each information item reaches a smaller set for crosscutting elements than for base.

Arcade describes each crosscutting element via a “concern and requirement id” causing all its crosscutting item reachability to be 0.5 with null standard deviation. Similarly, AOVG specifies its crosscutting elements as a source (i.e., concern of origin) and requirement name. However, unlike in Arcade, a source may contain more than one crosscutting requirement name. Consequently, the larger the number of crosscutting requirements per source, the closer this metric is to 1, making AOVG reachability vary between 0.5 and 1. The RDL values for this metric are more varied than for the other two approaches, changing from 0.3 to 0.7, yet its standard deviation is equal to that of the AOVG approach with the value of 0.2.

Thus, as shown in Fig. 8, on average all three approaches would normally use approximately 2 information items for crosscutting element definitions to reach a joinpoint. This indicator has a very low standard deviation for all three approaches which implies that the above conclusion can be made with high certainty.

The values for the base item reachability (shown in Fig. 9) are much higher for all three approaches. This is particularly noticeable for the RDL with average base item reachability value of 9.2. This is due to the rich lexicon-based referencing model of the RDL compositions. It should be noted that the standard deviation for this RDL indicator is quite high (5.8) for two reasons. First, the lexicon-based referencing can be broad (e.g., if multiple synonyms are defined) or narrow (e.g., if a term is used with no or few synonyms) depending on the number of entries in the lexicon. Second, the base set will be narrower if the semantic query is restricted to the elements of one concern (as for scenario 3).

The mode value for base item reachability for AOVG elements is 1.0. This indicates that the relevant base requirements are all enumerated by name, except for scenario 4 where “Specify.*” wildcard based reference was used. The value of base item reachability for Arcade varies from 0.7 for scenario 4 where no wildcard is used, to 6.3 in scenario 1.

Fig. 10 visualises the combined crosscutting and base element reachability for each approach. This metric is not completely accurate as several indicators include some “crosscutting-only” or “base-only” parts of compositions. These parts were added/changed when existing compositions were modified. Thus the added/changed parts were counted in the indicators, while stable ones were not. This could skew the objective characteristics of the compositions, though it reflects the objective nature of change. In accordance with this indicator the RDL, having used no wildcards, has the highest

Approach	Sc. 1	Sc. 2	Sc. 3	Sc. 4	Sc. 5	Total
RDL	3	6	2	19	2	32
AOVG	0	2	0	4	0	6
Arcade	0	0	9	0	0	9

Table 4: Remembrance values.

average reachability . Arcade has the next highest reachability due to its broad use of wildcards. Finally, AOVG has an average reachability of 1, indicating that an average item in an AOVG requirement composition is a named requirement or a named pointcut.

4.3.3 Semantic Compositions in RDL have Memory

The ability of compositions to pick out correct references in the face of change is evaluated by the remembrance metric, the results of which are presented in Table 4.

One immediately notices that the RDL provides remembrance-based references for each scenario. Moreover, in scenarios 2 and 4 no additional composition definition was needed, as the existing compositions were already sufficient. This was achieved without use of any wildcards.

The Arcade approach demonstrates some remembrance only in one scenario. The newly added requirements are incorporated into existing compositions because in a number of places generalisations such as viewpoint id=”all” are used. Notably, such generalisations will include all newly added viewpoints or requirements of a certain concern into their selection set even if the newly added viewpoints/requirements are not related to the composition intention. Consequently, in several cases the wildcard-based composition definitions had to be updated with an explicit “exclude” clause making the wildcard-based remembrance unreliable.

In case of the AOVG approach (scenarios 2, 4) the requirements were named in accordance with a convention adopted for this case study, to ensure the correct use of composition remembrance. Here, knowing that some compositions use “Register.*”, “Show.*” and “Request.*” wildcards, the requirements engineer intentionally named the corresponding requirements she wanted to include into these compositions. Even then, certain changes were still necessary to achieve correct compositions.

4.4 Summary

From the above discussion we conclude that the semantics-based approach uses, on average, the smallest number of information elements per composition with the lowest standard deviation. Arcade - the purely syntactic approach - uses both the highest average number of information elements per composition, and has the highest standard deviation. We explain this fact by the higher level of abstraction of elements used in semantic compositions: no id or named requirement references are used; grammar and semantics of natural language are exploited instead for quantification. Arcade, on the other hand, needs to either enumerate each element used in a composition via its “concern name-id” pair, or use wildcard-based quantification. The wildcard-based quantification, however, is not always possible to define (as occurred in scenario 4). Moreover, it is rather fragile, as shown, for instance, in scenario 5 for Arcade and scenario 4 for AOVG (e.g., Fig. 6).

Furthermore, semantics-based elements in the RDL demonstrate a higher average reachability than those of the syntax-based alternatives, although RDL also demonstrates a higher standard deviation. This is explained by the scoping and lexicon definition characteristics of the composition reference mechanism. If a lexicon entry for an element is narrowly defined, the element will have relatively narrow reachability, bordering, in the worst case, with the named-requirement like referencing mechanism of AOVG where only the joinpoints that exactly match the string of the given word are selected. However, such a narrow entry definition is rather unlikely. Normally a lexicon entry will be defined more broadly, and the broader its definition the wider the set of intended joinpoints that will be reached.

In addition, we have also seen (Table 4) that the RDL’s semantics-based compositions are better suited for preserving and enforcing the intention of the composition in a changing environment than the syntactic compositions of AOVG and Arcade. In other words the RDL compositions have memory of their intentions and are able to enforce these intentions when changes occur.

5. THREATS TO VALIDITY

In this section we consider a number of such threats to our experiment and our solutions that would minimise their effects on the study results.

5.1 Threats Arising from Chosen Artefacts and Participants

There is a threat that the original requirements documents and the change scenarios applied favour one approach over another. Furthermore, it could be the case that we purposefully selected weak representatives of semantic, syntactic or hybrid approaches to bias the results. To minimise these threats we have selected a pre-existing industrial case-study (HealthWatcher) which has been previously validated in other studies [8]. The changed scenarios applied were based on changes applied in a previous maintenance study to assess the stability of AO designs. We were able to re-apply these changes, in consultation with the original developers of HealthWatcher, to the requirements documents. Therefore, the requirements documents and maintenance changes have not been influenced by the AORE approaches examined in this study. Finally, each of the examined AORE approaches have been extensively peer-reviewed and validated through various publications [4, 5, 14, 15, 16, 17]. Therefore, we can be sure that these approaches are strong candidates for such a study. However, it is possible that future studies using different AORE approaches or different case-studies may uncover different results. Generalisation of the results can only be achieved by conducting more studies. In future works we plan to not only compare other AORE and non-AORE approaches but to extend the study by examining case-studies from other domains.

Another threat to the validity of the study is the participants selected for the study. A number of participants were necessary for this study to be conducted, including: the study designer, experts for each AORE technique, and data interpreters. To ensure the study was not influenced by any of the AORE approaches, the study designer created the study independently. To guarantee that each set of artefacts were of the best possible quality, the creators of the selected AORE approaches were recruited as testers. Furthermore, no restrictions were placed on the practices that the testers could use to achieve these results, thus ensuring that the best possible practice for each AORE approach was employed. While this best “known person and practice” approach provides some objectivity for comparable quality of artefacts, it also poses a question as to how well the approaches would have faired if an average user were evaluated instead. This however, is a subject for a different study.

Finally, the study designer also performed the alignment of the documents of various approaches. The designer is an expert in the HealthWatcher system and so has an excellent understanding of the requirements. However, he did not have an in depth understanding

of any of the AORE approaches analysed. It is important to point out that the purpose of this alignment process was not to change the fundamental output of each of the AORE approaches. Instead, the purpose was to align the HealthWatcher related concepts and terminology used in each set of artefacts to ensure that the artefacts could be compared. This did not require an in depth understanding of each AORE approach but needed a thorough understanding of HealthWatcher domain. Yet, there still is a threat that his individual perspective could have inadvertently instilled some elements that could have benefited one approach over the others.

5.2 Threats due to Metrics and Procedures

The area of semantics-based composition in RE (and in AOSD as a whole) is very young. Consequently no previous studies have been conducted or metrics developed. As a result, there are numerous threats to their validity. For instance, the metrics could be engineered to favour one approach over the others, or, even, not to be practically informative at all. To provide some level of trust to these metrics, we have based them on previously validated ones. When no suitable established metric was available for adaptation for our purposes, as was the case with our scaffolding metric, we relied upon our own experiences with requirements engineering and the observations of other experienced personnel [4, 5]. We believe that our metrics suite itself is a valuable and new contribution to AORE. Nevertheless, we also acknowledge that further validation of this metrics suite must form an integral part of our future work.

As mentioned previously, we employed a best "known person and practice" to employ the maintenance changes. We are of the view that this allowed us to use the strength of each approach, rather than force a change method which could be unsuitable for a given technique. Furthermore, the different decomposition mechanisms could be considered a contributing factor to the results we observed. We have attempted to minimise these effects by the mapping and alignment of different approaches discussed in Section 3.3.1.

Yet, the alignment and harmonisation procedures themselves could have biased some results. For instance, as discussed before, the 3rd level goals of AOVG were chosen to be equivalent to concerns of RDL and Arcade. However, as noted in Section 4.2.1, if a different level goals were chosen to represent concerns (though 3rd level fit best to the given case study), a different set of data would be collected for AOVG approach. Removal of such threats could only be achieved upon repeated application of the developed procedures to a number of case studies and experiments.

5.3 Miscellaneous Threats

A variety of other factors that threaten the validity of the study also need to be considered. The individual details of each AORE approach could influence the results; however, as these details are intrinsic to the AORE approach being studied it is undesirable to consider them in isolation. Furthermore, it is beyond the scope of this study to consider the individual details of each AORE approach; we are solely interested in comparing the different types of AORE approaches. The tool support offered by each approach may also influence the results. However, by allowing the expert testers to use the best possible practices any tools that were available have been used. No doubt, the development of new and improved tools for all three AORE approaches would alleviate some of the problems highlighted. We hope that the results from this study will inform and influence the development of such tools.

In this study we have considered that concern names and lexicon/ontology entries preserve their semantics across multiple applications (at least in the same domain). In our experience lexicon entries dedicated to such concerns as security, error handling, and persistence are stable and applicable across a number of domains and applications. However, each project also has its own application specific lexicon, and such lexicon entries may vary from project to project. Even within the same project entries obtained from different documents (written by different authors) may have different semantics. Further evaluation of the stability of the types of lexicon entries is necessary to provide a deeper understanding of the mobility index indicator.

6. RELATED WORK

A number of previous studies have assessed AO approaches at a variety of development stages. In this section we discuss a number of related studies and co-relate the conclusions drawn from these studies with the results of our study.

The first study examined is directly relevant to ours as it specifically focuses on comparing AO requirements engineering approaches [15]. The study proposes common process and naming schemes for AORE approaches to enable assessment of the effort expended and the quality of the requirement artefacts. The quality is measured in terms of precision and recall, whereas effort is measured in terms of time spent creating the requirement artefacts. One of the significant findings of this study was the large amount of effort that was spent on producing the compositions necessary to apply the crosscutting concerns. In fact, the composition specification task was specifically highlighted as an effort bottle-neck in AORE approaches. This finding is significant for our study, especially when one considers that all the approaches examined in this previous study relied upon syntactic composition mechanisms. Obviously, it would be necessary to re-conduct this experiment to include a semantics-based AORE approach to determine whether the effort is also reduced. However, the findings of our study, such as the reduction in scaffolding and the improved remembrance, suggest that semantics-based approaches may mitigate some of the overhead associated with specifying and modifying aspect compositions.

A similar study to ours was conducted, again using the HealthWatcher system, whereby a similar set of maintenance changes were applied to HW's implementation [8]. The purpose of this study was to determine how well AO implementations could maintain a stable design compared to an OO implementation. From this study, a number of conclusions were drawn that are relevant when discussing semantics- and syntax-based composition. Firstly, the results highlighted the need for semantics-based pointcuts due to the abundance of fragile pointcuts that are present in the implementation of HW. The results of our study have highlighted the benefits of semantics-based composition and confirm that semantics-based pointcuts can address the fragile pointcut problem through semantically motivated mobile elements, reliance on abstractions, and higher remembrance to remain resilient in the face of change. Besides, a smaller number of changes have to be made to RDL's compositions compared to that of the syntax-based approaches. Secondly, the design stability study introduced the notion of wide and deep ripple-effects that relate to different types of unanticipated changes that occur. The results from the design stability study show that ripple-effects in AO tend to go deeper in that they affect more seemingly unrelated artefacts, whereas OO ripple-effects tend to go wider in that they affect related artefacts much more extensively. We have observed that the RDL is able to localise the changes more efficiently than the syntax-based approaches and so reduce the deep ripple-effects. However, it is difficult to directly attribute this benefit to semantics-based approaches in general as the modularity of the examined approaches has not been assessed in our study.

The dependency of aspects on the syntactic structure is also demonstrated in the study performed by [7]. The purpose of this study was

to determine how AO can cope when performing extensive restructuring. This study again illustrated the reliance of syntax-based aspect compositions on the underlying structure and the problems which this causes. For example, a number of relocation, renaming and redistribution activities were performed that negatively affected the aspects that were dependent on the modified artefacts. Some of the changes performed in our study were of a similar nature (whereby concerns/requirements had to be renamed or restructured in certain approaches). However, the semantics-based compositions were not influenced by the restructuring changes and no concern renaming was necessary.

The final study examined [1] involves assessing the evolution of a program's lexicon compared to the evolution of the program's structure. The study found that the program's lexicon is more stable than the program's structure and that changes to the lexicon are rare. This conclusion, in some ways, goes against the findings of our study. The majority of the changes that are made to the RDL artefacts occur within the lexicon. However, the fact that the RDL does not have any reliance on the structure of the base elements naturally causes more changes to occur in the lexicon than would otherwise be expected. Furthermore, the previous study was performed at the implementation level. It should be expected that the lexicon is more stable at this stage of development. Due to the requirements engineering phase occurring earlier in the development life-cycle, where the problem domain is still being understood hence definitions may change as more information is elicited causing the lexicon to be altered. It should be noted that it is only domain specific entities in the lexicon that are altered during the maintenance changes performed in our study. Well-understood generic crosscutting lexicon entities, such as security and persistence, did not require modification at any point in the study.

7. CONCLUSION

In this paper, we have presented a first empirical study analysing the potential benefits of a purely semantics-based composition mechanism for AORE (RDL) compared to a purely syntax-based one (Arcade) and a hybrid approach (AOVG). Our study confirms the benefits intuitively perceived, i.e. semantics-based compositions in the RDL are less fragile and more expressive than syntax-based ones in Arcade and AOVG. We have also uncovered some interesting challenges for semantics-based compositions as realised in the RDL, including the need for advanced tool support, effort involved in initial lexicon development and the need to explore composition interdependencies when introducing a new composition.

Our study also uncovers some key insights about composition mechanisms in general. Firstly, it shows that the extent of scaffolding itself is not a major hurdle. In fact, it is the nature of scaffolding that one needs to consider when developing or using a composition mechanism. Scaffolding elements with location-independent meaning are more mobile and can be used in other concerns or applications in the same domain. Secondly, our study indicates that wildcard-based quantification mechanisms can, in fact, reduce the abstraction power of a composition mechanism rather than foster it. Thirdly, our study highlights the need for compositions to exhibit memory about their original intention, i.e. when a change occurs relevant new joinpoints are selected and irrelevant ones discarded without any change to the composition or the need to restructure the base elements to provide hooks. These results, though derived specifically in the context of our specific AORE study, represent generic underpinnings for aspect composition mechanisms. We hope that future studies, both by ourselves and others, will aim to validate these findings with a view to developing fundamental guidelines for the design of aspect composition mechanisms.

8. ACKNOWLEDGEMENTS

This work was partially supported by the European Commission grant IST-215412 - Dynamic Variability in complex, Adaptive systems (DiVA), and the European Commission grant IST-33710 - Aspect-Oriented, Model-Driven Product Line Engineering (AMPLE).

9. REFERENCES

[1] G. Antoniol, Y.-G. Guéhéneuc, E. Merlo, and P. Tonella. Mining the lexicon used by programmers during sofware evolution. In *ICSM*, pages 14–23, 2007.

[2] E. Baniassad and S. Clarke. Theme: An approach for aspect-oriented analysis and design, 2004.

[3] S. R. Chidamber and C. F. Kemerer. A metrics suite for object oriented design. *IEEE Trans. Softw. Eng.*, 20(6):476–493, 1994.

[4] R. Chitchyan, A. Rashid, M. Pinto, and L. Fuentes. Compass: composition-centric mapping of aspectual requirements to architecture. In *Transactions on Aspect-Oriented Software Development IV*, LNCS, pages 3–53. Springer, feb 2007.

[5] R. Chitchyan, A. Rashid, P. Rayson, and R. Waters. Semantics-based composition for aspect-oriented requirements engineering. In *AOSD '07*, pages 36–48, New York, NY, USA, 2007. ACM.

[6] L. Chung, B. A. Nixon, E. Yu, and J. Mylopoulos. *Non-Functional Requirements in Software Engineering*. Springer, October 1999.

[7] C. Gibbs, C. R. Liu, and Y. Coady. Sustainable system infrastructure and big bang evolution: Can aspects keep pace? In *ECOOP*, pages 241–261, 2005.

[8] P. Greenwood, T. T. Bartolomei, E. Figueiredo, M. Dósea, A. F. Garcia, N. Cacho, C. Sant'Anna, S. Soares, P. Borba, U. Kulesza, and A. Rashid. On the impact of aspectual decompositions on design stability: An empirical study. In *ECOOP*, pages 176–200, 2007.

[9] A. Kellens, K. Mens, J. Brichau, and K. Gybels. Managing the evolution of aspect-oriented software with model-based pointcuts. In *ECOOP*, pages 501–525, 2006.

[10] R. Knöll and M. Mezini. Pegasus: first steps toward a naturalistic programming language. In *OOPSLA '06*, pages 542–559, New York, NY, USA, 2006. ACM.

[11] C. V. Lopes, P. Dourish, D. H. Lorenz, and K. J. Lieberherr. Beyond aop: toward naturalistic programming. In *OOPSLA Companion*, pages 198–207, 2003.

[12] I. Nagy, L. Bergmans, and M. Aksit. Composing aspects at shared join points. In *NODe 2005, GSEM 2005*, volume p-69 of *LNI69*, pages 19–38, 2005.

[13] K. Ostermann, M. Mezini, and C. Bockisch. Expressive pointcuts for increased modularity. In *ECOOP*, pages 214–240, 2005.

[14] A. Rashid, A. Moreira, and J. Araújo. Modularisation and composition of aspectual requirements. In *AOSD '03*, pages 11–20, New York, NY, USA, 2003. ACM.

[15] A. Sampaio, P. Greenwood, A. F. Garcia, and A. Rashid. A comparative study of aspect-oriented requirements engineering approaches. In *ESEM '07*, pages 166–175, Washington, DC, USA, 2007. IEEE Computer Society.

[16] L. Silva. *A Guided Strategy the Modeling Aspect-Oriented Requirements (in Portuguese)*. PhD, Rio de Janeiro, Brazil: Catholic University of Rio de Janeiro (PUC-Rio), 2006.

[17] L. Silva, T. Batista, A. Garcia, A. Medeiros, and L. Minora. On the symbiosis of aspect-oriented requirements and architectural descriptions. pages 75–93. 2007.

[18] S. Soares, E. Laureano, and P. Borba. Implementing distribution and persistence aspects with aspectj. *SIGPLAN Not.*, 37(11):174–190, 2002.

[19] D. Stein, S. Hanenberg, and R. Unland. Expressing different conceptual models of join point selections in aspect-oriented design. In *AOSD '06*, pages 15–26, New York, NY, USA, 2006. ACM.

[20] W. P. Stevens, G. J. Myers, and L. L. Constantine. Structured design. *IBM Systems Journal*, 13(2):115–139, 1974.