A Tool Suite for Aspect-Oriented Requirements Engineering

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

Américo Sampaio
Computing Department
Lancaster University
Lancaster, LA1 4WA, UK
a.sampaio@comp.lancs.ac.uk

Awais Rashid, Paul Rayson
Computing Department
Lancaster University
Lancaster, LA1 4WA, UK
{awais;paul}@comp.lancs.ac.uk

ABSTRACT
Aspect-Oriented Requirements Engineering (AORE) supports identification of crosscutting, aspectual requirements as well as analysis of their influence on other requirements of the system. Identifying and analyzing aspectual requirements manually is very resource intensive due to their broadly scoped nature and the large volumes and ambiguity of input information from the stakeholders. In this paper we present a tool suite to support AORE in a scalable fashion. The tools support identification of aspectual requirements and their influences on other requirements, conflict detection and resolution between aspectual requirements, as well as requirements representation and requirements document structuring. A number of case studies, including two in an industrial setting, demonstrate the scalability and efficiency of the tool suite. They also show that its output is comparable to that of a requirements engineer carrying out the same tasks manually.

Categories and Subject Descriptors
D.2.1 [Requirements/Specifications]: *tools*; D.2.9 [Management]: *productivity*.

General Terms
Management.

Keywords
aspect-oriented requirements engineering, concern identification, requirements analysis, requirements composition, aspect-oriented software development.

1. INTRODUCTION
Requirements Engineering (RE) is primarily concerned with information collection and structuring tasks which can be categorized into Requirements Gathering and Requirements Analysis. Every activity (e.g., interviews, ethnographic studies, etc.) in Requirements Gathering results in production of additional (often textual) documents. The wealth of these documents needs to be analyzed during Requirements Analysis. Individual requirements are identified and structured into concerns (e.g., use cases, viewpoints, goals, etc.). The concerns are then analyzed for dependencies and trade-offs. The outcome of RE is usually a specification document (signed off by the stakeholders) for use in development of the architecture, design, implementation and system acceptance tests.

Aspect-Oriented Requirements Engineering (AORE) [4, 5, 9, 15] has emerged as a new way to modularize and reason about crosscutting concerns during requirements engineering. AORE extends the notion of separation of concerns in RE (e.g., viewpoints, use cases, goals, etc.) with that of requirements-level aspects. Such aspects modularize requirements that affect and constrain other requirements. Requirements pertaining to these concerns are often (fully or partially) scattered in the statements of other requirements. By explicitly modularizing crosscutting concerns at the requirements-level, AORE makes it possible to reason about such concerns in isolation from early on in the software lifecycle. By providing support for composition of requirements-level aspects with base concerns in the system, AORE facilitates analysis and understanding of the influences and constraints exerted by the former on the latter. The composition also makes it possible to identify potential trade-offs among requirements-level aspects, i.e., when two or more aspects influence or constrain the same set of requirements. These trade-offs can then be resolved early on, for instance, by weakening the requirements of one aspect with reference to the other after negotiations with the stakeholders. This early understanding of aspect trade-offs plays a significant role in shaping the system architecture [4, 5].

Identification of aspectual requirements is, however, a non-trivial task. Firstly, as is the case for identifying relevant concerns using any RE technique, one often has to mine for aspects in large volumes of input documents. Documents, such as interview transcripts, are frequently imprecise, full of apparent contradictions and missing essential information. Secondly, parts of aspectual concerns can often be scattered across a document or even across documents making their identification difficult. This is further compounded by factors such as the occurrence of similar, often incomplete requirements in several places, mutual influence of requirements, difference of language (user vocabulary) used to express same or similar requirements and implicit requirement implication. Furthermore, once identified, the aspects need to be examined for such issues as dependencies, conflicts, their resolution and then structured (along with other requirements) into a Requirements Specification document.

Undertaking such identification and analysis tasks manually is often very time-consuming and costly. For instance, for an analyst with an average reading speed, it would take 1.5-2 minutes (at the rate of 250-350 words per minute) to read the problem description in Figure 1. Identifying key concepts, concerns and crosscutting relationships requires even more time and effort. In our lab studies,

EA'06, May 21, 2006, Shanghai, China.

it took novice aspect-oriented developers on average 6 person hours to identify the initial set of viewpoints and the aspects crosscutting them in this small problem description. An experienced aspect-oriented developer, on the other hand, required 10 minutes for the purpose. When we extrapolate this data to manual processing of large documents, the average personnel effort required is substantial. Structuring requirements, analyzing the influences of aspects on other concerns, analyzing mutual trade-offs among aspects as well as resolving such trade-offs is also resource intensive, if undertaken manually. Therefore, like other RE techniques, AORE requires effective and scalable tool support so that its benefits may be fully exploited in analyzing large scale problems.

In this paper we describe a tool suite for AORE, which employs Natural Language Processing (NLP) to support the requirements engineer in identification of both crosscutting (aspectual) and non-crosscutting concerns. Tools in the suite also support structuring of the requirements into the concerns identified. Requirements-level aspect composition and analysis of aspect influences and trade-offs are also supported.

Auction System Problem Description

An online auction system is commissioned that will be used for buying and selling of goods over the Internet in auctions. The owners of the system want to promote it as a secure and credible marketplace for which the system must be protected and must guarantee that all bids placed in auctions are solvent.

All potential users of the system must first enrol with the system; once enrolled they have to log on to the system for each session. Then the customers are able to browse the auctions available on the system. In order to be authorised to buy or sell through the system a customer needs to set up a credit account with the system and invest a minimum amount in it. This account will be used as security on each and every bid. A customer can increase his/her credit by asking the system to debit a certain amount from his/her credit card. If a customer wants to close his/her auction system account, he/she can ask the system to transfer his/her credit back to his/her credit card. The system owners will charge a small amount for initial credit setup.

A customer can act as a seller and a buyer. A seller must initiate an auction by inputting into the system details on the goods to auction with the minimum bid price and reserve price for the them, the start period of the auction, and the duration of the auction, e.g., 30 days. The seller has the right to cancel the auction as long as the auction has not already started.

Customers that wish to follow an auction must first join the auction by logging in. It is only possible to join an active auction. Once a customer has joined the auction, he/she may make a bid, or post a message on the auction's bulletin board. A bid is validated by the system. The bid is valid if it is over the minimum bid increment and if the customer's credit with the system is at least as high as the sum of all pending bids. Buyers can place bids until the auction closes, and simultaneously place bids across as many auctions as they please. When auction closes, the system calculates whether the highest bid meets the reserve price given by the seller. If the reserve price is met, the system deposits the highest bid price, minus the commission taken for the system owners for the auction service, from the buyer's auction account into the seller's auction account.

The auction system is highly concurrent - clients bidding against each other in parallel, and a client placing bids at different auctions and increasing his/her credit in parallel.

Figure 1: Auction Problem Description

The contributions of the tool suite are fourfold:

- Efficient initial requirements structuring from all kinds of input documents – for large documents we have observed significant reduction in time spent, at times to two orders of magnitude, on aspect identification and subsequent structuring of the requirements document;
- Effective identification of composition relationships in which crosscutting concerns participate;
- Composition and subsequent analysis of influences of crosscutting concerns on other system requirements;
- Support for identification of mutual trade-off points amongst crosscutting concerns for negotiations with the stakeholders.

In section 2 we present a general process for AORE. Our tool suite supporting the process is discussed in section 3 with the help of the example problem description of the online auction system adapted from [10] (see Figure 1). Section 4 evaluates our tool suite based on our experiences with analysis of larger problem descriptions and industrial case studies and Section 5 concludes.

2. AORE PROCESS

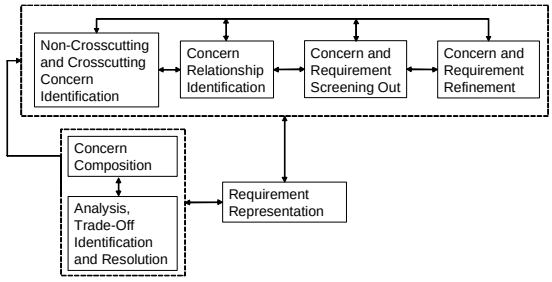


Figure 2: A General Process for AORE

The AORE process supported by our tool suite is shown in Figure 2. This is a general AORE process synthesized from the various AORE processes proposed in literature to date, e.g., [4, 5, 9, 15]. All the tasks in the process can be carried out iteratively and not necessarily in the presented order.

Once a set of initial requirement gathering documents is available (e.g., interview transcripts, ethnographic studies, etc.) the process commences with the *concern identification* step (Figure 2). Here the various base concerns (e.g., viewpoints, use cases, etc.) as well as the crosscutting concerns, i.e. the aspects cutting across the base concerns, are identified.

Concern identification is followed by the *identification of relationships among concerns*. This is necessary in order to understand how concerns relate to each other, e.g., "extends" relationships among use cases. More importantly, however, it is essential to facilitate reasoning about the crosscutting influences and constraints imposed by aspects on other concerns.

Having identified the concerns and their relationships, we can decide which concerns, relationships and requirements are pertinent to the intended software system, and if there are repetitions in the identified list. This step is termed *screening out* in Figure 2.

The remaining pertinent concerns, their requirements and relationships are then represented in a chosen format. This may be text, graph, or another format. The representation selection is dictated by the specific AORE approach used for the analysis (e.g., [4] structures requirements into viewpoints and aspects, [9] structures requirements into base and crosscutting themes, etc.). This step is necessary for production of the final requirements specification document.

However, during concern representation one may identify a need for refinement. New concerns or requirements may arise. Similarly, new or alternative relationships may be identified. Thus, the requirements representation and the previous identification steps are linked.

Finally, the requirements represented in the selected notation need to be analyzed for the influences exerted by the aspects as well as for potential conflicts among aspects. These analyses are facilitated by the *concern composition* and *analysis, trade-off identification and resolution* steps. The results of the composition are analyzed, conflicts identified and resolved in consultation with the stakeholders.

3. TOOL SUITE FOR AORE

Our tool suite, depicted in Figure 3, consists of several integrated tools with each consecutive one working on the output received from the previously applied tool. The tools support the requirements engineer in carrying out the various tasks in the AORE process in Figure 2.

The tools in our suite play two types of roles: *information generator* and *information consumer*. The information generators analyze the input documents (transcripts of stakeholder interviews, ethnographic studies, etc.) and complement them with linguistic, semantic, and statistical information and annotations. The information consumer tools use the annotations and additional information for multiple types of analysis. This facilitates reuse of once generated general information for multiple purposes in multiple tools and helps in establishing clear information provider-consumer links between tools.

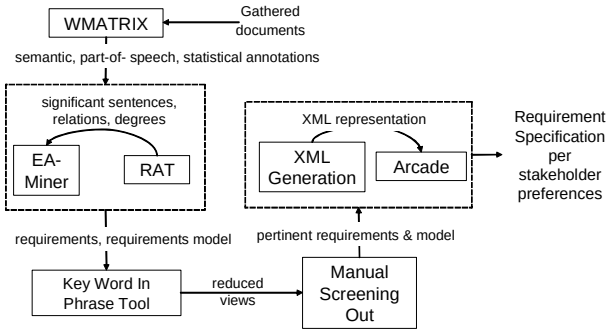


Figure 3: Tool Suite for AORE

3.1 Information Generator Tools

We start by submitting the gathered information documents to WMATRIX [1] – a collection of corpus-based natural language processing tools. WMATRIX is web-based, with a straightforward input/output web interface, and is our main information generator. It uses a combination of part-of-speech and semantic tagging, frequency analysis and concordances (i.e. words in context) to identify domain concepts of potential significance. Part-of-speech analysis automates extraction of nouns and verbs (as well as other word classes) from the text with a high degree of precision (98%). Each word from the text is assigned a part-of-speech tag. Then semantic analysis is used to group related words and multi-word expressions into conceptual categories even when many different word forms are used in the documents. In WMATRIX corpus if a word has several meanings, it is assigned to several semantic categories. When an input text is analyzed, the most likely category for each word is selected, taking into account the context where the word occurs.

For instance, from the Figure 1 description, the words customer(s), buying, selling and bid are grouped into the semantic field of Business: Selling. The tool outputs such semantic categories along with their overuse/under-use statistics, indicating the key concepts addressed by the input text.

The "keyness" of words used is calculated by comparing the frequency of usage of each word to that of the usage of the same word in the reference corpora – the British National Corpus. If a set of words in the given text is used significantly more frequently than in the reference "standard" corpus, such words are likely to indicate the importance of the concepts in the given text. The statistically significant set of words, thus, obtained facilitate identification of the first cut of the user requirements. A sample of such data for the auction example (as displayed by WMATRIX) is presented in Figure 4.

From the sample in Figure 4, we can see that the requirements are mainly about Selling (LL 357.93) with related terms of auction, buying, selling, customers, and bids; On-line system, i.e. the mental-object which the text describes (LL 75.34), and Money (LL 21.07) with related terms such as credit, credit card, and invest. The list of the concepts for each semantic category in the document can be viewed from the List hyperlink, and their occurrence in the document from the Concordance hyperlink.

However, the semantic categories of WMATRIX (based on [1]) are general language categories that do not take into consideration the specifics of requirements and their relationships – from a perspective of facilitating requirements understanding and analysis of their relationships. In order to incorporate these additional semantics, we have augmented WMATIRX categories with additional sub-categorisation for *Relationship*, *Temporal and Conditional Sequencing*, and *Degree*. Figure 5 depicts the sub-categories additionally demarked as belonging to the *Relationship* category. Similar classifications have been derived for *Temporal and Conditional Sequencing*, and *Degree* categories (Figure 6).

		Item	O1	%1	O2	%2	LL	Semantic Group
List	Concordance	I2.2	61	14.32	2738	0.28 +	357.93	Business: Selling
List	Concordance	X4.2	21	4.93	3108	0.32 +	75.34	Mental object:- Means, method
List	Concordance	I1.1	9	2.11	2654	0.27 +	21.07	Money: Affluence

KEY:
O1 is observed frequency
O2 is observed frequency for normata
%1 and %2 values show relative frequencies in the texts.
+ indicates overuse in O1 relative to O2,
- indicates underuse in O1 relative to O2
The table is **sorted on log-likelihood (LL) value** to show key items at the top.

Figure 4: Data Generated by WMATRIX

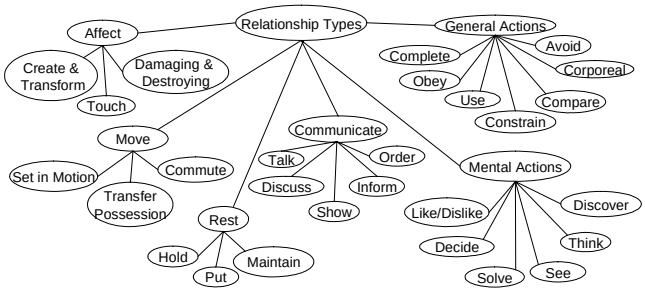


Figure 5: Semantic Categories as Relationships

The category for *Relationship* contains the subset of words which can be classed as related to *effects of actions*. We use these sub-categories to analyze the effects of requirements on each other. Examples of words marked to belong to the *Relationship* sub-category in our Auction example, for instance, are *have to*, *initiate*, *make* etc. These words are noted as it is likely that *having* to do something or *initiating* mentioned in one requirement occur with regard to some other requirement.

Each relationship category suggests a general type of relationship, thus facilitating the understanding of potential influences of the requirements on each other. For instance the *initiate* in our example belongs to the Move category's Set in Motion sub-category (Figure 5). If further analyzed, this verb is classified as belonging to a finer-

grained Beginning/Ending sub-category (not depicted in Figure 5), indicating that there is a Beginning/Ending type relationship between *seller* and *auction*. The relationships also suggest the composition actions for requirements composition, as discussed in Section 3.2.4.

The Relationship categories have been identified by drawing on the other linguistics work on semantic verb groups [2, 3] and generalizing the semantic categories of WMATRIX lexicon itself. This linguistics perspective has been complemented with action categories from practical case-study investigations as we have worked to align the categories with the actions already identified in our previous work on requirements composition [4, 5].

An effective word can belong to more than one category, for instance, *make* belongs to both *Create & Transform* and *Order* categories. In such cases both possible groups are indicated by WMATRIX and the final selection is left for the requirements engineer to make.

The *Temporal and Conditional Sequencing* category groups the words that indicate the temporal and conditional dependencies between requirements. This category is separated from the general *Relationship* category because of the specific ordering of the requirements required by its semantics. The words belonging to this category in the auction example are, for instance, *first*, *once*, *then* (see 2nd paragraph in Figure 1), etc. These categories are depicted in Figure 6 (a).

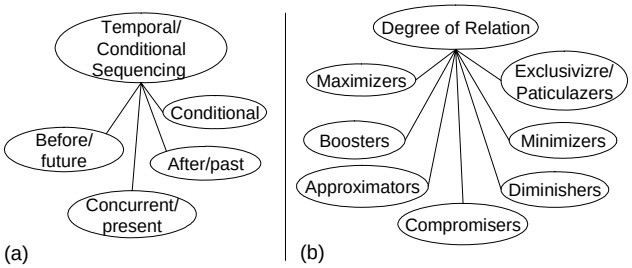


Figure 6: (a) Temporal and Conditional Sequencing Categories; (b) Degree Categories

The *Degree* category (cf. Figure 6 (b)) groups the words that assist in evaluating the strength of the relationship between the requirements or degree of some desired or unwanted property. Examples of such words in the auction case study are: *at least, highly.*

In addition to annotating the words with their semantic categories and generating statistical information, WMATRIX also demarcates (with <s></s> tags) and numbers each sentence as a candidate for later requirement extraction. If desired, a requirement-unit boundary may be assigned to a text of different granularity, e.g., a paragraph, etc., but our experience suggests that a sentence is most appropriate in the majority of cases.

3.2 Information consumer tools

Our information consumer tool set includes the Requirements Analysis tool (RAT); Early Aspect Mining (EA-Miner) tool [6, 7]; Key Word In Phrase (KWIP) tool; and the Aspectual Requirements Composition and Decision Support (ARCADE) tool [4]. These tools utilize the wealth of information produced by WMATRIX to assist the requirements engineer in the tasks in Figure 2.

3.2.1 Concern and Relationship Identification

The task of concern identification deals with both aspectual and non-aspectual concerns. This is carried out by the EA-Miner tool [6, 7]. EA-Miner utilizes information produced by WMATRIX and complements it with its own lexicon for crosscutting concerns and requirement document representation models (e.g., viewpoints model, use cases model, etc.).

For identification of non-functional crosscutting concerns (which are often strong candidates for aspects), the EA-Miner lexicon builds on top of (and extends) non-functional requirement trees of the NFR framework [8]. Non-functional aspects are identified by assigning semantically close words to the sub-groups of each NFR category. For instance (as shown in Figure 7), in our auction example, EA-Miner lexicon helps to identify the words *authorised* and *logging* as semantically related to the *security* concern and suggests *security* as a non-functional aspect. For identification of functional crosscutting concerns, EA-Miner uses a Theme/Doc-like [9] strategy, detecting the repeated occurrences of action words, which may suggest presence of a functional aspect. Thus, EA-Miner also alerts the requirements engineer (not shown in Figure 7) that the *bid* functionality is mentioned in many requirements, which may indicate some crosscutting association of *bidding*.

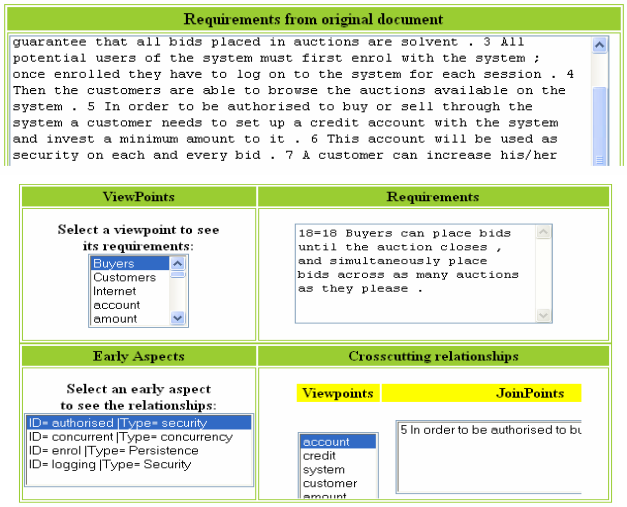


Figure 7: EA-Miner tool

EA-Miner also facilitates the production of the aspect model and requirements document representation with chosen structures. For instance (cf. Figure 7), EA-Miner helps to automatically identify viewpoints (e.g., buyers, customers, etc.), using the WMATRIX part-of-speech annotation; all nouns are identified as potential viewpoints. Since the list of nouns for large documents can be very long, EA-Miner applies several reduction strategies:

1. lemmatization to recognize words with the same root (e.g. *customer* and *customers*) and treating these as one viewpoint;
2. dictionaries of synonyms to amalgamate words with the same meaning (e.g. *client* and *customer*);
3. WMATRIX usage statistics to consider only significantly overused nouns as potential viewpoints.

EA-Miner then provides the list of requirements related to each viewpoint and its related aspects. Such representation is well suited for use in ARCADE. Note that EA-Miner is not limited to viewpoint-based structuring of requirements. It can also be applied to develop use cases style requirements documentation by identifying possible use cases from the action verbs (e.g., sell, buy, bid, etc.) and relating corresponding requirements to them. Similar reduction strategies to those outlined above are also usable for other kinds of representations.

The Requirement Analysis tool (RAT) uses the statistical information produced by WMATRIX to identify the requirements containing the statistically significant words used in the input text on the grounds that requirements related to significant concepts are likely to be significant. For instance, one requirement identified in the auction example is: "Then the *customers* are able to browse the *auctions* available on the *system*", as it contains several words of the highly significant *Selling* and *Mental Object* categories. In fact, due to the nature of our example document, all the sentences in this small text have significant words, and are suggested as potential requirements. This, though, is very unlikely when large documents (e.g., interview transcripts, etc.) are used.

RAT uses the *Relationship*, *Temporal and Conditional Sequencing*, and *Degree* sub-group annotations to determine the dependencies between and the ordering of the identified initial requirements. The initial requirements can later on be refined, e.g., merged, separated into two or more requirements, dropped, or re-worded, etc. For instance, in our auction example, the tool suggests to consider the *have to* effective word from the *Relationship* sub-category and *first*, *once* words from the *Sequencing* sub-category in the following requirement: "All potential users of the system must *first* enroll with the system; *once* enrolled they *have to* log on to the system for each session".

Having considered these words, we may refine the initial requirement into two requirements: (1) "All potential users of the system must enroll with the system" and (2) "The users have to log on to the system for each session". An explicit temporal ordering of these requirements is defined in the Specification document when specifying the composition relationships between concerns encapsulating these requirements.

The additional elaboration of requirements by RAT can be used to reduce the set of requirements to only those containing significant elements.

3.2.2 Screening Out

The KWIP tool is applied to support the manual screening out of redundant concerns and requirements. KWIP extends the standard WMATRIX concordance technique applied by varying the amount of context around a key word based on a set of heuristic filters. For instance, a filter is applied to avoid splitting words or clauses when displaying concordances by reducing or extending the amount of context on display. This allows the user to view the main phrases or parts of the requirement sentences containing the words identified by WMATRIX in key domain concepts, thus reducing the amount of information to consider. It is also very useful in determining the value of the requirement relationships indicated earlier by the effective words of the *Relationship* group. Hence, the initial requirement structures produced by the EA-Miner and RAT tools are reduced to key-phrase representation by KWIP for manual screening out, while a link to the full documents is also provided in case more detail is required.

3.2.3 Refinement

Refinement occurs continuously throughout the Identification, Screening, Representation and Composition tasks, when concerns or requirements are merged, divided into sub-concerns or sub-requirements respectively, new concerns, requirements or relationships are identified, or existing ones deemed unnecessary. For instance, a requirement splitting example was discussed earlier in Section 3.2.1 with respect to the *have to* effective word.

This identification, screening and refinement cycle (cf. Figure 2) leads to a relatively complete set of concerns (in this case aspects and viewpoints) and their associated requirements along with their corresponding relationships and ordering, if any. The aspects and viewpoints form input to ARCADE, which uses the viewpoints as a base to observe and analyze aspect influences and trade-offs. Note that, as mentioned earlier, ARCADE is one possible tool to be used for analyzing the influence and trade-offs of aspectual requirements. Other tools, e.g., Theme/Doc [9] can be used instead with appropriate representations from EA-Miner.

3.2.4 Concern Representation and Composition

ARCADE uses XML representations, governed by an XML schema, to specify viewpoints, aspects and associated composition rules. The composition rules are derived from the relationships and temporal and conditional sequencing identified by RAT. The semi-structured representation makes it easier to analyze the otherwise purely textual requirements based on composition rules.

One can either manually convert the viewpoint and aspect definitions into XML or use the built in XML generator from ARCADE. The generator also provides a wizard to specify the composition rules. A snapshot of the wizard is shown in Figure 8(a). Here we can see the Security aspect and the Buyer viewpoint. The requirements engineer can select which specific Security requirements influence specific requirements in the Buyer viewpoint. A follow up screen (not shown) allows the requirements engineer to choose the relevant relationships, temporal sequencing, etc. to formulate the composition rules. An ARCADE composition rule comprises of:

- *action* which specifies the type of influence/relationship (e.g., enforce, provide, etc.) directed from the aspect towards the viewpoint;
- *operator* which specifies the previously identified temporal and constraint orderings between aspect and viewpoint requirements (e.g., *first* maps to a *before* operator, *then* to an *after* operator, etc.);
- *outcome* which specifies the expected outcome of the composition (e.g., whether another requirement needs to be satisfied or just the specified constraint needs to be fulfilled).

Figure 8(b) shows a composition rule generated by the wizard. The rule states that *Security requirement 3* (All potential users of the system must enroll with the system) must be *enforced before Buyer requirement 10* (A customer can act as a seller and a buyer). The *outcome* of the composition should be that this *constraint is fulfilled* (which implies that no additional requirements need to be satisfied as a result of the composition).

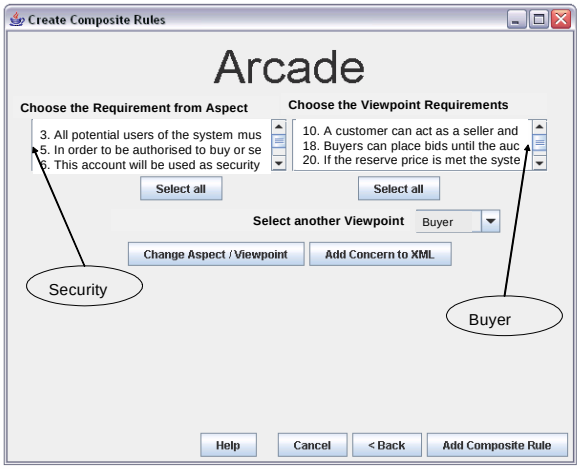
3.2.5 Analysis of Influences, Trade-offs and Trade-off Resolution

The ARCADE composition mechanism projects the aspects on the viewpoints hence making it possible for the requirements engineer to clearly see how an aspect potentially influences or constrains specific viewpoint requirements. The tool also includes a trade-off analyzer component which identifies overlapping between aspects with reference to the viewpoint requirements they influence. The requirements engineer is alerted to these potential trade-off points. S/he decides whether the overlapping aspects strengthen each other, i.e., positively contribute to each other or not. If the mutual contribution is negative, then stakeholders are requested to attach fuzzy importance values to aspects. Fuzzy values are useful in this

regard as stakeholders often refer to their priorities as very important, not so important, etc. Fuzzy logic-based analysis helps map these priorities to weakening of one aspect's requirements with reference to another's to resolve the conflicts. In cases where the conflicting requirements are given equal priorities, the stakeholders must negotiate amongst themselves and agree on an alternative prioritization.

As depicted in Figure 3, the output from ARCADE is the Requirements Specification based on a systematic treatment of crosscutting (as well as non-crosscutting) concerns.

(a)



(b)

```
<Composition>
    <Requirement aspect="Security" id="3">
        <Constrain action="enforce" operator="before">
            <Requirement viewpoint ="Buyer" id="10"/>
        </Constrain>
        <Outcome action="fullfilled"/>
</Requirement>
....
```

Figure 8: ARCADE Composition (a): Wizard; (b): Rule

4. DISCUSSION AND EVALUATION

Earlier in the paper, we highlighted the significant manual effort required for identification and analysis of aspects in AORE. In order to assess the usefulness and scalability of our tool suite we focused on the most time consuming of these activities: the identification of the crosscutting (aspects) and non-crosscutting (in our example viewpoints) concerns and structuring of the requirements according to these concerns. We compared results of the tool-based approach with the manual analysis for three problem descriptions of varying sizes and document structures.

The problems used were the *auction system* described in Figure 1 (443 words, 1 page); the light control system [10] (3671 words, 11 pages), and a library system used in Lancaster University (6504 words, 29 pages).

The execution of both methods (tool-based and manual) was carried out independently by different requirements engineers (with similar level of expertise and knowledge of AO) in order to avoid biased results. We collected data on the time required in each case and on the *quality of the output* measured as the number of correctly identified concerns (i.e., viewpoints, aspects).

In this evaluation the tool-based analysis consistently outperformed the manual one. For the auction system description it was approximately 20 times faster (4 minutes 45 seconds vs. 90 minutes). For the larger documents the performance of the tool suite was even better: approx. 125 and 120 times faster for the light control and library systems respectively.

In terms of correctness of viewpoints identified, the output of the two analyses was the same for the auction system description. However, the tool-based analysis was more accurate (i.e. corresponding with analysis of a human expert) for the two larger problem descriptions.

For aspect identification both results were comparable for the light control system. For the auction system, the manual analysis missed one aspect (persistence). For the library system the manual analysis identified 7 out of the 8 aspects, while the tool-based analysis identified only 4. This is due to the fact that vocabulary for such aspects as *standards and protocols* (mentioned in the library system) was not part of the EA-Miner lexicon. Relevant vocabulary has now been added to the lexicon.

As more analyses are performed in other case studies, and the tool lexicon is populated with more aspect-related vocabulary, the tool's performance on correctness criterion will improve further.

There are also interesting observations in terms of false positives, i.e., abstractions (viewpoints and aspects) redundantly identified as correct. This became noticeable in evaluation of the two larger problem descriptions. The rate of false positives (number of false positives/total number of abstractions identified) for viewpoints was approximately 0.1 for the tool suite and zero for the manual analysis. On the other hand, for the tool suite, the rate of false positive aspects was zero compared to 0.3 for the manual analyst. The subsequent interviews with the analysts helped explain these contrasts. For viewpoint identification, the tool considers all nouns as potential candidates. However, usually subjects in a sentence tend to make stronger candidates for the purpose. As viewpoints are well understood, the analyst working manually placed a stronger emphasis on subjects as viewpoints. Conversely, aspects at requirements are not as well-understood abstractions. Thus, the manual analyst preferred to err on the side of caution identifying more aspects than needed. In contrast, the analyst working with the tools was more clearly guided with regards to potential aspects hence resulting in no false positives. This demonstrates the value of the tool suite in helping with the adoption of aspect-oriented requirements engineering techniques, especially when extending existing viewpoints- or use case-based approaches with the notion of aspects.

4.1 Experiences in industrial case studies

Subsequent to the above evaluation of the tool suite, we are using it to assist with case studies of re-engineering existing software systems to an aspect-oriented architecture. One industrial setting involves a large, structured (use case based) requirements specification of an air traffic control system (multiple documents comprising several hundred pages) while the other is a satellite telemetry simulator. In the latter case, there is no extensive requirements specification available. The only available documentation is user manuals and a description of the final functionality of the system. Due to confidentiality reasons, we are unable to go into any details of either system. However, the tool suite has helped us greatly to gain initial insights into potential candidate viewpoints and aspects to guide the re-engineering. This is particularly useful for the second system where no extensive requirements specification is available – the tools have helped us to understand the various concerns of the stakeholders from other documentation. In both cases it took us only a few hours to obtain a first cut of our aspects and viewpoints. We could then specify this initial set in ARCADE for a rough analysis of potential trade-off points for discussion with the stakeholders. This has facilitated

understanding the priorities of the stakeholders, before undertaking an extensive re-structuring of the requirements specification and re-architecting the systems.

5. RELATED WORK

The use of NLP in automation of some tasks in RE has been discussed in [11-13], though none of these approaches addresses crosscutting concern identification. Circe [11] and COLOR-X [12] are based on parsing an initial set of structured natural language requirements into an intermediate model to further generate specific models (ER, DFD, OO design, etc). In [13] the Abstfinder tool is used for identification of abstractions (i.e. concepts such as "booking a flight") in requirements elicitation documents. Unlike WMATRIX, both Circe [11] and COLOR-X [12] approaches expect that the software engineer will provide the list of some key terms which can then be used to build models. Besides, none of the above automation approaches consider the broad influence of crosscutting requirements.

The requirements engineering method PREView [14], with its supporting tool [16], recognizes the crosscutting influence of some (organizational) concerns and supports their analysis. However, the tool does not automate the activities of concern/viewpoint identification, structuring, and relating concerns to viewpoints – all these done manually.

The NFR approach [8] (supported with the NFR-Assistant tool [17]) acknowledges the broad influence of non-functional requirements on the other requirements by providing positive or negative contributions. However, this tool, and approach, assist neither with automation of concern identification nor with requirements structuring from arbitrary textual input.

The work in [15] also addresses composition, providing a supporting language and goal-based visualization. While our tool suite is not yet complemented with a graph-based visualization tool, our composition is enriched with the semantics of concern relationships as well as their degree of relatedness.

Theme/Doc [9] provides a tool for semi-automatic identification of crosscutting behaviors from requirements specification. However, here the developer has to read through input documentation and manually identify a list of action words and their related entities as input to the Theme/Doc tool.

Thus, NFR, PREView, goals and aspects, and Theme/Doc all require manual concern/softgoal/action word identification and structuring, even if some initial concern decomposition structures (such as softgoal trees, and Theme/Doc visualization) are available to support analysis after identification. This implies that the analysts need to read through available documentation (e.g. manuals, standards, etc.) and analyze user input (e.g., interviews, etc.) before the concerns can be identified and requirements structured. With our tools no such initial commitment is required.

6. CONCLUSION

As AOSD techniques move towards maturity, it becomes increasingly important to treat aspects systematically throughout the software lifecycle. AORE provides software engineers the ability to identify and analyze aspects from the very early stages of system development. However, like other RE techniques, AORE needs to be supported by effective and scalable tools if its benefits are to be fully realized. Our tool suite provides such a set of tools supporting the requirements engineer in not only aspect identification but also determining compositional and temporal relationships between aspectual and non-aspectual requirements. It also supports analysis of aspect influences and trade-offs hence providing opportunities for early trade-off resolution. This, in turn, can lead to more informed architecture and design choices that are well aligned with the stakeholders' intentions.

7. ACKNOWLEDGMENTS

This work is supported by EC grant IST-2-004349: European Network of Excellence *AOSD-Europe*, 2004-2008. The authors thank Christopher Atkinson implementing the ARCADE GUI.

8. REFERENCES

[1] *UCREL Semantic Analysis System*, Lancaster University http://www.comp.lancs.ac.uk/ucrel/usas/, 2005.

[2] B. Levin, *English verb classes and alternations: a preliminary investigation*: Univ. of Chicago Press, 1993.

[3] R. M. W. Dixon, *A Semantic Approach to English Grammar*, 2 ed. Oxford: Oxford University Press, 2005.

[4] A. Rashid et al., "Modularisation and Composition of Aspectual Requirements," AOSD 2003, ACM, pp. 11-20.

[5] A. Moreira et al., "Multi-Dimensional Separation of Concerns in Requirements Engineering" RE 2005, pp. 285-296.

[6] A. Sampaio et al., "EA-Miner: a Tool for Automating Aspect-Oriented Requirements Identification," ASE, 2005.

[7] A. Sampaio et. al., "Mining Aspects in Requirements," Early Aspects, Workshop (at AOSD), Chicago, USA, 2005.

[8] L. Chung et al, *Non-Functional Requirements in Software Engineering*: Kluwer Academic Publishers, 2000.

[9] E. Baniassad, S. Clarke, "Theme: An Approach for Aspect-Oriented Analysis and Design," ICSE'04, IEEE, pp 158-167

[10] *Light Control Case Study: Problem Description*, http://wwwagss.informatik.uni-kl.de/Veroeffentl/jucs2000.pdf , Univ. of Kaiserslautern, Germany, May 2005.

[11] V. Ambriola and V. Gervasi, "Processing natural language requirements," Proc. ASE 1997, pp. 36-45.

[12] F. M. Burg, *Linguistic Instruments in Requirements Engineering*: IOS Press, 1997.

[13] L. Goldin and D. Berry, "AbstFinder: A Prototype Natural Language Text Abstraction Finder for Use in Requirements Elicitation," *ASE*, vol. 4(4), 1997, pp. 375-412.

[14] I. Sommerville "Viewpoints for requirements elicitation: a practical approach," ICRE'98, pp. 74-81.

[15] Y. Yu et al. "From Goals to Aspects: Discovering Aspects from Requirements Goal Models," Proc. RE 2004, IEEE CS, pp. 38-47.

[16] *JPreview Tool*, Computing Department, Lancaster Univ., http://www.comp.lancs.ac.uk/computing/research/cseg/projects/deada/JPreview.html, 2005.

[17] Q. Tran and L. Chung, "NFR-Assistant: Tool Support for Achieving Quality," Application-Specific Systems and Soft. Eng. & Tech.,1999, IEEE Computer Society, pp. 284.