Abstract — It is well-known that effective requirements analysis plays a crucial role in the quality of software systems. However, the scattered and tangled nature of certain system's concerns can hinder the proper understanding and treatment of import requirements. A key goal of prominent Aspect-Oriented Requirement Engineering (AORE) techniques, such as EA-Miner and Theme/Doc, is to support the automatic identification of crosscutting concerns at textual requirements documents. However, it is still unknown whether and which of these approaches produce accurate results in large text documents and according to the software engineers' expectations. In this context, this paper presents an analysis regarding the accuracy of the aforementioned AORE approaches when processing requirements of two industry software systems. Around 300 pages of requirements descriptions in these systems were the target of our investigation. In general, EA-Miner suffered more than Theme/Doc from the incompleteness and inconsistencies of requirements documents. In addition, other factors can differently influence each approach's accuracy, such as: the participation of requirements engineers, and the level of details provided in the requirements document.

Keywords – requirements analysis; requirements documents; modularity; early aspects.

I. INTRODUCTION

The size of textual requirements documents in industry systems typically ranges from dozens to hundreds of pages [34][35]. Hence, software engineers spend a considerable amount of their time on analyzing the requirements specification [34][35]. A major source of complexity in requirements analysis is the identification and reasoning about crosscutting concerns [7]. These concerns are usually related to non-functional requirements, but they can also be related to functional ones [1][3]. In both cases, it is far from trivial to identify them in an ad hoc fashion due their scattered and tangled nature, which hinders in turn the proper understanding and treatment of requirements. On the other hand, their identification is essential as these concerns are often: (i) required to be analyzed in isolation [2][4][6][18], (ii) conflicting with other concerns [2][20], and (iii) candidates to be modularized in the architectural design [36].

Over recent years, several researchers [1][3][4][6] in the field of Aspect-Oriented Requirements Engineering (AORE) have developed techniques and tools for supporting the representation and analysis of crosscutting concerns. They also offer new ways of modularising systemic requirements in units called early aspects [16]. More importantly, these approaches provide automated support for identifying crosscutting concerns in textual documents, which is essential to reduce the effort of software developers on analyzing requirements documents [1][14][16][17][32].

In fact, the AORE proponents claim that their approaches lead to effective detection of crosscutting concerns at an early stage of software development [2][19][21]. However, the fact is that there is not much work on measuring the actual effectiveness of AORE techniques in large systems. In particular, there is limited information about the accuracy required to identify crosscutting concerns in industry requirements documents. So far, all studies were usually performed by the own proponents of these techniques. Even worse, the evaluations also tend to be applied in the context of the same small systems (Section VI). As a result, it is still unknown whether these approaches produce accurate results according to the expectations of software engineers in large-scale real systems.

In order to address this gap, our study is aimed at analyzing the accuracy of prominent AORE approaches on the identification of crosscutting concerns in two industry systems. We selected and compared two representative AORE approaches, namely EA-Miner [19][21] and Theme/Doc [2] (Section II), as they provide tool support robust enough to be independently applied in large text documents. These techniques are well cited in the literature, while still being relevant nowadays [37][38]. For instance, they are consistently serving as the basis for the development of recently-proposed approaches, such as (but not limited to) EA-Analyzer [37] and EA-Tracer [38].

In our study, a number of empirical procedures were applied to promote a fair and equitable evaluation of both techniques (Section III). It was assigned different analysts who had to apply the corresponding tools using two different systems. We quantify the accuracy of the gathered crosscutting concerns with the traditional precision and recall metrics. The results of the analysis activities were compared with results obtained from senior requirements engineers. We also quantified the
degree of automation of each approach by measuring the time spent on each activity of the analysis process. Both systems used in our evaluation are from the banking domain—a client complaint system and an ATM record management system. Their requirements are specified using natural language descriptions and tables. The requirements document of the client complaint system consists of around 230 pages, whereas the ATM system document has approximately 65 pages. As far as the gathered data is concerned, interesting results were observed (Section IV):

- EA-Miner yielded better results than Theme/Doc in the identification of concepts in requirements models. The superiority of EA-Miner was due to its characteristics of: (i) part-of-speech and the semantic tagging, and (ii) the automatic generation of requirements documents in different formats.

- The size and level of details in the requirements document may influence on the performance and the accuracy of the AORE approach. This finding suggests that the AORE approaches should be chosen taking into consideration these two properties of the requirements document (Section IV.B). The time spent on the concepts identification is more sensitive to the requirements completeness with EA-Miner than when using Theme/Doc. This sensitiveness seems to become more evident as the size of requirements documents grows. Therefore, this result helps the developers to decide which approach should be used based on the completeness degree of a requirements document (Section IV.A).

- Proper actions of requirements engineers can help to significantly improve the accuracy of the AORE approach when detecting functional concerns. This finding is supported by the fact that Theme/Doc yielded better accuracy than EA-Miner when engineers had to manually determine the themes and simultaneously start the concern identification activity (Section IV.B).

The rest of this paper is organized as follow. We also report some threats to validity (Section V) of the conducted study, and compare it with related work (Section VI). Section VII draws some conclusions and points out directions for future work.

II. BACKGROUND

AORE approaches are mainly aimed at supporting requirements engineers to reveal crosscutting concerns. AORE approaches can be classified into two main groups: asymmetric and symmetric approaches. In asymmetric approaches there is an explicit distinction between crosscutting and non-crosscutting concerns. In symmetric approaches all concerns are treated in the same way [7] In this paper, we compare two AORE tool-based approaches: the Theme/Doc [2] and the EA-Miner approaches [19][20]. The latter is a representative of asymmetric approaches, while the former is considered a symmetric approach (both crosscutting and non-crosscutting concerns are themes). The choice of these approaches was driven by their maturity and availability of relevant expertise and robust tool support, thereby enabling us to conduct empirical analysis of AORE. In addition, both approaches can operate over textual descriptions of requirements based on the use of natural language. Nonetheless, our experimental methodology defined here (Sections III e IV) can be applied in further empirical studies to compare other AORE methods.

Next we present a brief overview of the AORE approaches used in our exploratory study. A more detailed description of these AORE approaches can be found in [2][19][20]. Before discussing them, we introduce a common terminology that enables us to contrast their particularities.

A. Terminology

The AORE approaches have the same final goal: support the identification of crosscutting concerns. However, the terminology used by them is different. Therefore, in our work, we adopted the unifying terminology proposed by Sampaio et al. [22], which is revisited in this section. A fundamental difference between AORE and traditional requirements engineering approaches is the way modularization units are structured. In AORE approaches the concept of early aspects is treated differently, i.e. they are explicitly modularized [1][16][33]. An early aspect is as an abstraction that modularizes the requirements of a crosscutting concern.

Examples of crosscutting concerns in requirements documents can be of non-functional (e.g., Security, Performance, and Logging) or functional nature (e.g. Add Item to Shopping Cart and User Sign-in). These two kinds of crosscutting concerns are called non-functional and functional early aspects [21], respectively. In both cases, an aspect influences or constrains functional concerns of the requirements specification [4][17].

In AORE approaches, a set of previous tasks is required before the actual identification of crosscutting concerns can be accomplished. However, evaluating and comparing AORE approaches are daunting tasks, due to the variation spectrum that separates the different approaches. Hence, it is necessary to commit to a common process scheme in order to compare them [21]. This strategy is also required to allow one to collect measures that can be further used for comparative analysis of these AORE approaches. In our work, we also adopted the common process scheme proposed by Sampaio et al. [22], described as follows.

The common process scheme denotes a set of activities that are common across several AORE approaches. The process scheme does not address all requirements engineering activities. Rather, it focuses mainly on three major tasks required to support the identification of crosscutting concerns in requirements documents. These tasks are described as follows [22]:

- Identification of requirements model concepts: this is subdivided into three other activities, namely the identification of viewpoints, identification of early aspects, and identification of crosscutting relationships.
- Structure requirements specification: this is subdivided into three sub-activities: gathering viewpoints requirements, gathering early aspect requirements, and specifying composition rules.
• Conflict resolution and validation: encompasses the sub-
activities of building a contribution table to represent
mutual influences of early aspects (e.g., security contributes
negatively to performance), to assigning weights so as to
to quantitatively assess the degree of conflicts, to resolve
conflicts, and finally to support decision making.

In the following subsections, we characterize both
Theme/Doc and EA-Miner approaches in terms of the three
tasks described above. The use of this single terminology will
enable us to contrast the accuracy of these approaches, while
also assessing the time spent on each of these major AORE
tasks (Section IV).

B. Theme/Doc approach

Theme/Doc is part of the Theme approach [1] and it is
centered on the notion of a theme. A theme is a collection of
structures and behaviours that represent a system feature [1]. In
the requirements document, two kinds of themes can be
observed: base themes and crosscutting themes. Base themes
are cohesive subsets of requirements that represent a
stakeholder’s functional concerns (Section II.A). Crosscutting
themes are similar to base themes, but they overlay the
functionality of the base themes. According to Baniassad and
Clarke crosscutting themes are considered as aspects in
requirements documents. Therefore, a crosscutting theme can
either represent a functional or non-functional early aspect
(Section II.A). In reality, Theme/Doc was created to address
functional early aspects, but adapted to consider non-functional
aspects.

Themes may be related to each other, and such
relationships may cause overlaps in the themes. There are two
main ways in which themes can relate [1]: (i) via sharing
concepts: where different themes have elements that represent
the same core concepts in the system domain, and (ii) via
crosscutting relationships, where behavior in one theme is
triggered by behavior in other themes. The main contribution
of Theme/Doc is a set of heuristics for the analysis of
requirements documents. A key goal of these heuristics is the
identification of crosscutting themes.

In this work we focus on the requirements analysis
activities, supported by the Theme/Doc, as our interest is to
investigate their accuracy on supporting the identification of
early aspects in requirements specifications.

Identification of model concepts. This activity relies on two
main Theme/Doc components.

• Identify themes: identify behavior described in the
requirements that could potentially be a theme. In order to
get a list of potential themes, one must be able to identify
key concerns from the requirement specification text. The
main approach to identify themes consists of classifying all
verbs, or action words, written in the requirements as
potential themes, then analyze which terms are real themes
and to group verbs to form bigger actual themes. Themes
can be directly mapped to use cases.

• Determine theme responsibilities: Once themes were
identified, we must decide which themes are aspects, and
which ones are base. In the Theme/Doc approach an aspect
is a theme whose description within a requirement is
tangled with the description of another theme. Requirements that are linked to more than one theme
(shared requirements) are the places where, more likely,
this intertwining occurs.

Structure the requirements specification. Once the themes
are indentified, there are four operations that can be performed
over themes to restructure their requirements: split them up in
case they are too general; add new ones, if anything is missing;
delete a theme that seems unhelpful or irrelevant; group themes
that seem similar, i.e. combining them into one larger theme.

Conflict Resolution. In the Theme/Doc approach, the
component responsible for this process is the plan for design
activity: once the themes were identified and the overlap of the
relationships between them is minimized. We use the
individual-theme view to analyze each theme individually and
organize them to move forward to design.

C. EA-Miner approach

The Early-Aspects Mining Tool (EA-Miner) provides the
following activities with automated support:

• Mining requirements-level early aspects: Identifying early
aspects (functional and non-functional ones) from a variety
of early stage requirements specifications.

• Structuring different AORE models: Structuring the
identified early aspects and other concepts, such as
viewpoints [33] into a specific AORE technique (e.g.,
viewpoint-based AORE [8] or use case based AORE [9]).

The foundation of EA-Miner’s automation is the use of
natural language processing (NLP) to reason about the
semantics revealed by the natural language analysis. The goal
is to identify the concepts and build the AORE models,
features provided by the WMATRIX NLP tool suite [18],
which have been shown to be effective in early phases of
requirements engineering [25][3][11]. WMATRIX implements
NLP techniques such as frequency analysis, part-of-speech
(POS) and semantic tagging (SEM) that provide relevant
information about the properties and semantics of a text in
natural language. EA-Miner utilizes WMATRIX to pre-process
a requirements document provided as input. WMATRIX
returns another file, which consists of the same content as the
input file, but tagged with POS and SEM tags. The main
activities of this approach are presented next.

Identification of model concepts. In the EA-Miner approach,
the activities responsible for the identification of model
concepts are:

• Viewpoint Identification, Presentation and Structuring: For
each word contained in the parsing unit, the parser
identifies the viewpoints (and their related requirements).
The rule used for identifying the viewpoints is based on the
part-of-speech (POS) attributed to the word. Words whose
POS represents nouns are identified as Viewpoints
candidates. For example, in toll collection systems,
viewpoint candidates are driver, gizmo, vehicle owner and
toll gate. The problems of identifying words with the same
root (e.g., driver and drivers; gate and gates) and synonyms (freeway and motorways) as different concepts can be corrected by applying filters.

- Early Aspect Identification, Presentation and Structuring: To identify non-functional early aspects (e.g., performance, security), the following steps take place: 1) The tool looks for the semantic (SEM) tag of each word in the sentences of the parsed input document and searches a lexicon file to check if there is one entry of the same word with the same SEM tag. If a match is found, the requirements sentences in which the word appears are added to the list of early aspect candidates. 2) EA-Miner searches for words, that were not found in the previous step, and whose semantic category belongs to semantic classes to which several NFRs belong. At the end of this step the list of candidates identified is added to the previous list.

- Crosscutting Relationship Identification: Each requirements sentence is understood as a join point and the collections of requirements are compared based on a set intersection operation. If the resulting set is non-empty then there is a crosscutting relationship between the viewpoint and the early aspect and the join points are the resulting set.

Structuring the requirements specification. Despite the fact that the former activities help to structure the requirements specification; there is an activity responsible for this task, called Gathering viewpoints and early aspects requirements. In this activity the user is able to add/remove any of the identified concepts (e.g., viewpoints, early aspects, requirements) that s/he considers relevant/irrelevant, respectively. A filter that helps to minimize the number of candidates is setting a threshold to get only the N most significant concepts (e.g., viewpoints) based on the frequencies returned by WMATRIX.

Conflict resolution: The EA-Miner approach does not support this activity of the AORE process; however the EA-Miner’s output can be easily used as input by other tools (e.g., Arcade [28]) that perform conflict identification.

III. STUDY SETTINGS

We decided to perform an exploratory investigation to analyze different AORE approaches with respect to the accuracy of their outcomes. These analyses were performed in two real software systems (Section III.A). A number of preparation and execution steps were carried out in order to conduct our exploratory study (Section III.B).

A. The Target Industry Systems

The first major decision that had to be made in our study was the selection of the target applications. The chosen systems are real and critical applications deployed in a bank located in the Latin America. The software development process in the company is certified as CMMI Level 3 by the SEI, i.e. Software Engineering Institute at Carnegie Mellon University. Each development team has around 30 professionals involved, and the specification of each requirement can receive the contribution of up to five team members.

The systems were selected because they met a number of relevant criteria for our intended evaluation. First, they are real and complex banking systems comprising a wide range of functional and non-functional requirements. Their explicit specification also allowed us to analyze the AORE approaches (Section II) when dealing with large systems, which may demand the use of these approaches. Second, we have access to the documentation of these systems, which consist of 65-page and 230-page requirements descriptions. They also encompass a rich set of widely-scoped crosscutting requirements, including Performance, Persistence and Correctness. Third, the available requirements documents are mostly governed by natural language descriptions, but they also include some complementary use cases, high-level process diagrams and window interface descriptions. Finally, we could count on the help of systems domain experts to contribute to the evaluation of the obtained results.

The biggest system is a web-based information application that allows customers to register and monitor complaints regarding banking operations. The second system is an application for controlling ATMs. For simplification, we will respectively refer to them as “complaint system” and “ATM system” in the following sections. Listings 1 and 2 present examples of basic requirements in both documents, which are

"The bank user needs to track all transactions made from all the ATM machines around the country. A consolidation statement must be generated from each and every ATM. The statement must contain the balance of all bills inside the ATM machine, the number of forgotten and locked cards, the ATM’s identification, and the identification of the people responsible for it. This consolidation statement will be filled out using an Excel template which can be uploaded to the system by the person in charge of the ATM machine. In order to evaluate the record, a set of states must be created since the beginning of the process until its end. During the evaluation process, the bank’s online system must to be compared to the consolidation statement system, so that irregularities can be detected while performing normal transactions (e.g. withdrawing money) which will improve customer care service.”

Listing 2. Example of requirements description in the ATM system.

1 Due to confidentiality agreement, the identity of the organization cannot be disclosed.
relevant to understand the main purposes of these systems. For reasons of confidentiality we cannot disclose some confidential information of this study; for instance, we removed the organization name from the examples and other details that might reveal their identity.

B. Study Phases and Assessment Procedures

After the selection of the target applications, the study encompassed five major phases: (1) the removal of any kind of formatting standards in the requirements documents, as demanded by the AORE approaches (Section II), (2) the execution of the tasks by using both AORE approaches, (3) the evaluation of the results by senior requirements engineers, (4) the time analysis regarding the tasks conducted by the engineers in the phase 2 and (5) the accuracy analysis of the results provided by each applied AORE approach.

Removing the formatting standards. The structure of the requirements documents used in our study are governed by the company formatting rules. The specifications follow a number of formatting rules related to the sections’ structure, font styles, headings, tables, and graphs. However, both approaches, Theme/Doc and EA-Miner, require a plain text document as input, i.e. a text document without any kind of style. In fact, the lack of formatting standards in the requirements specification documents is a well-known hindrance to the application of AORE approaches.

Consequently, a preprocessing of the requirements documents was necessary. To ensure adherence to the input format required by tooling support in each approach, the following processing actions were performed. First, we removed all the stylistic elements related to headings, subheadings, numbering, bullets and indexes of the original requirements documents. Second, all tables were transformed to plain text; in the transformation, we tried to keep the same semantics governing the relationship between elements in the different columns. Third, all the graphics were interpreted and translated to into plain text. Hence, the graphics were removed from the requirements specification. Finally, mathematical formulae were decoded and rewritten into plain text.

Using the AORE approaches. Our exploratory analysis involved the participation of two requirements engineers, which have similar experience in AORE approaches. Each engineer was responsible for using one of the AORE approaches described in Section II in order to perform the same task. The final goal of their tasks was to restructure the original documentation into a new specification, in accordance to the AORE approach designated to them. It is important to highlight that all the engineers involved in this stage were familiar with AORE approaches and, in particular, they were experts on the approach that they had to use in this study. Also, none of the engineers was previously familiar with the banking systems, enabling us to rule out any influence of previous knowledge on the study results. The engineers just had access to the systems requirements specification previously mentioned in Section II.A. Finally, there was no communication between the engineers and they were not aware of each other’s progress in order to preserve the validity of the results.

Analyzing the correctness data. In order to evaluate the accuracy of the results provided by the used approaches, we asked for the contribution of senior requirements engineers, who are experts in the systems domain. In particular, we asked the senior engineers to provide the lists of (1) functional concerns, (2) non-functional aspects, and (3) functional aspects that they deemed to be the correct ones. These lists are the output of the used approaches (Section II). Considering that each senior engineer analyzed the systems in an independent way (i.e. without being aware about the existence of other specialists), the lists provided by a senior engineer were not the same of those contemplated by the other specialists. In order to circumvent this situation, all the involved senior engineers gathered together, so that each explained their point of view and then they converged to a consensus. The outcome of the discussions was used to create new lists of concepts, whose elements were confirmed by the majority of the specialists. Therefore, the resulting set of lists was considered as the official oracle in our study.

Collecting the timing data. In the context of our investigation, the time-effectiveness variable is related to the tasks conducted by each engineer in the three processes of the common AORE process: identification, structuring and conflict resolution, which were introduced in Section II.A. During the identification process we considered the time to identify the themes themselves, themes responsibilities for the Theme/Doc approach. Regarding EA-Miner approach it was considered the time elapsed in identifying viewpoints, concerns and crosscutting relationships. During structuring process it was taken into consideration the time to gather requirements into themes for the Theme/Doc approach. In the context of the EA-Miner it was measured the time to gather viewpoints and early aspects requirements. Finally, the time invested in the conflict resolution process was only considered for the Theme/Doc approach because the EA-Miner approach does not support this activity as part of its common process. In particular, it was taken into consideration the time to plan the design activity of the Theme/Doc approach.

It is important to notice that all the engineers were monitored by a third person (i.e. the experimenter) when working on the tasks in order to guarantee a correct time measurement and do not disturb the engineers’ work. The engineers were asked to announce to the experimenter every time they logged the time, so that the experimenter could reset its personal timer. The experimenter was instructed to be strictly honest, and cautious to avoid counting elapsed time while the engineers’ work was accidentally interrupted by other activities, not related to the exploratory study at hand.

Analyzing the accuracy of the AORE approaches. We used two metrics, precision and recall, to analyze to what extent the both AORE approaches Theme/Doc and EA-Miner supported the correct identification of the functional concerns, non-functional early aspects and functional early aspects. The investigation of these measures was based on both set of lists: (1) detected elements using the AORE approaches, and (2) provided elements by senior engineers. In particular, the lists provided by the senior engineers were useful to analyze which concepts were not identified by the existing approaches.
Therefore, the precision and recall were defined in the context of this investigation as follows.

\[
\text{Precision} = \frac{|\{\text{detected} - \text{elements}\} \cap \{\text{provided} - \text{elements}\}|}{|\{\text{detected} - \text{elements}\}|}
\]

\[
\text{Recall} = \frac{|\{\text{detected} - \text{elements}\} \cap \{\text{provided} - \text{elements}\}|}{|\{\text{provided} - \text{elements}\}|}
\]

IV. STUDY RESULTS

This section reports the findings related to the observed behavior on the use of both AORE approaches in the target systems. We focus on presenting and discussing the most interesting data and findings. However, all the study material is available at a complementary website [36]. As previously mentioned (Section II), all AORE approaches require previous steps to be accomplished before the actual identification of crosscutting concerns is carried out. Hence, Section IV.A presents a comparative discussion of both EA-Miner and Theme/Doc approaches in terms of their performance (i.e. time spent) when performing each of these steps required to process the requirements documents. Section IV.B discusses the findings related to the accuracy of the approaches to identify the functional concerns, non-functional aspects and functional aspects.

The performance evaluation (Section IV.A) is also important to enable us to understand what is the effort required to apply both approaches. As a result, it also supports us on conducting a wider analysis of tradeoffs governing effort and accuracy. It might be the case that a technique provides slightly-higher accuracy on revealing crosscutting concerns, but it demands much more effort beforehand. In this case, the application of this technique might not be worthwhile when comparing with the other in terms of both criteria.

A. Comparative Performance Analysis

There was a significant difference on the time spent on the use of each approach needed to process the requirements document. We have analyzed the time of each approach required in the following activities (Table I): (i) identification of Requirements Model Concepts (RMC); (ii) structure requirements specification; and (iii) conflict resolution. The results collected for each activity are summarized in Table I. The EA-Miner approach does not provide support for conflict resolution (Section II). The “Total” row in this table represents the total time each approach required to perform analytical activities on the requirements document. A discussion of the results collected by using each approach is presented in the following paragraphs in terms of each major AORE task.

Identification of requirements model concepts. As shown in Table I, EA-miner presented significantly better results than Theme/Doc in this activity for the complaint system. The better results in the complaint system were due to the part-of-speech (POS) and the semantic tagging (SEM) provided by the WMATRIX (Section II). These characteristics facilitated the automatic viewpoint identification in the EA-Miner approach. On the other hand, in the Theme/Doc approach the requirements engineers are in charge of reading the whole requirements document to identify potential functional units (i.e. the themes). This activity is essential to support the identification of crosscutting concerns afterwards.

Therefore, from the perspective of the complaint system, this characteristic could be considered a disadvantage of the Theme/Doc approach, since a valuable time was invested in visually identifying and confirming the themes. In particular, the choice of initial entries for key concerns by requirements engineers can play a decisive role, as Theme/Doc requires a series of iterative theme refinement sessions in order to amass a comprehensive set of themes. After identifying the responsibilities for the themes, the engineer is able to identify the functional early aspects. The Theme/Doc approach does not prescribe any particular method to identify non-functional aspects. Furthermore, the EA-Miner approach demanded less time than Theme/Doc in the identification of crosscutting relationships. This happened because EA-Miner relies on a fully automated feature with this purpose. By contrast, the Theme/Doc approach requires a manual evaluation of the themes in order to identify the central theme when a

<table>
<thead>
<tr>
<th>EA-Miner Approach Task</th>
<th>Time (in minutes)</th>
<th>Theme/Doc Approach Task</th>
<th>Time (in minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification of RMC</td>
<td>Identification of viewpoints 25</td>
<td>Identification of themes</td>
<td>220</td>
</tr>
<tr>
<td></td>
<td>Identification of concerns 20</td>
<td>Identification of theme responsibilities</td>
<td>160</td>
</tr>
<tr>
<td></td>
<td>Identification of crosscutting relationships 20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Structure requirements specification</td>
<td>Gather viewpoints and early aspect requirements 5</td>
<td>Gather requirements into themes</td>
<td>320</td>
</tr>
<tr>
<td>Conflict Resolution</td>
<td>Not supported</td>
<td>Accomplish the planning for the project phase</td>
<td>60</td>
</tr>
<tr>
<td>Total Time (in minutes)</td>
<td>70</td>
<td></td>
<td>760</td>
</tr>
</tbody>
</table>

ATM System

<table>
<thead>
<tr>
<th>EA-Miner Approach Task</th>
<th>Time (in minutes)</th>
<th>Theme/Doc Approach Task</th>
<th>Time (in minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification of RMC</td>
<td>Identification of viewpoints 34</td>
<td>Identification of themes</td>
<td>64</td>
</tr>
<tr>
<td></td>
<td>Identification of concerns 32</td>
<td>Identification of theme responsibilities</td>
<td>56</td>
</tr>
<tr>
<td></td>
<td>Identification of crosscutting relationships 36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Structure requirements specification</td>
<td>Gather viewpoints and early aspect requirements 18</td>
<td>Gather requirements into themes</td>
<td>54</td>
</tr>
<tr>
<td>Conflict Resolution</td>
<td>Not supported</td>
<td>Accomplish the planning for the project phase</td>
<td>50</td>
</tr>
<tr>
<td>Total Time (in minutes)</td>
<td>140</td>
<td></td>
<td>214</td>
</tr>
</tbody>
</table>

RMC = Requirement Model Concept
requirement is shared by two or more themes. The requirements engineer needs to invest a large effort to complete this task. However, we also observed that the time spent in this Theme/Doc activity helped the subjects to acquire a deep knowledge about the problem domain, which can be used in later steps. These steps include the conflict resolution activity and the confirmation of crosscutting concern candidates.

When we consider the second system, the difference was not significant in this AORE task: Theme/Doc required two minutes less than the time spent when using EA-Miner in the ATM case. Therefore, we considered both approaches had a similar performance for the ATM system. Interestingly, we observed that the superiority of EA-Miner was not confirmed in this second case for two main factors: (i) the time spent on the concepts identification is more sensitive to the requirements completeness with EA-Miner than when using Theme/Doc, and (ii) the larger is the requirements description, the better is the performance of EA-Miner compared to Theme/Doc. As far as the former factor is concerned, if the requirements documents do not fully provide full-fledged descriptions of the functional requirements, EA-Miner does not perform well on the identification of functional aspects (Section IV.B). On the other hand, the identification of non-functional aspects is not visibly influenced by the completeness factor, as EA-Miner relies on the use of NFR knowledge base to identify potential concerns in the requirements document (Section IV.B).

**Structuring of Requirements Specification.** During the structure requirements specification activity the EA-Miner approach also presented better results than Theme/Doc. In the EA-Miner approach, the user has the option of generating part of the requirements specification document in different formats (e.g., XML file or DOC file) automatically. By contrast, in the Theme/Doc approach the user has to evaluate the requirements specification, and only then perform the task manually. This is the main reason why the Theme/Doc required a lot of time in this stage and, consequently, presented worse results than EA-Miner.

**Conflict Resolution.** The Theme/Doc approach allows the user to define a hierarchical structure of crosscutting themes. In this structure, the base themes as well as the crosscutting themes are differentiated, thus enabling the identification of crosscutting concerns between them. Conflict Resolution is resolved by the Theme/UML. The EA-Miner approach does not provide any automated support for conflict identification among aspects, but it generates an RDL (Requirements Description Language) file that can serve as input to an additional tool, designed for the detection and resolution of conflicts among early aspects.

### Table II. Analysis of the Data Accuracy

<table>
<thead>
<tr>
<th></th>
<th>EA-Miner Approach</th>
<th>Theme Approach</th>
<th>Correct Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DE</td>
<td>CE</td>
<td>PE</td>
</tr>
<tr>
<td>Complaint System</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Functional Concerns</td>
<td>45</td>
<td>14</td>
<td>22</td>
</tr>
<tr>
<td>Non-functional Aspects</td>
<td>7</td>
<td>5</td>
<td>11</td>
</tr>
<tr>
<td>Functional Aspects</td>
<td>9</td>
<td>7</td>
<td>11</td>
</tr>
<tr>
<td>ATM System</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Functional Concerns</td>
<td>17</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>Non-functional aspects</td>
<td>7</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Functional Aspects</td>
<td>19</td>
<td>12</td>
<td>15</td>
</tr>
</tbody>
</table>

DE = Detected Elements; CE = Correct Elements; PE = Provided Elements

### Table III. Automatically-detected Elements using the Analyzed Approaches in the Complaint System

<table>
<thead>
<tr>
<th></th>
<th>EA-Miner Approach</th>
<th>Theme Approach</th>
<th>Correct Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Functional Aspects</td>
<td>Form, Account, Create, Find, Associated, Record, Send, Claims, Client</td>
<td>Form, History, Find, Assign, Log, Validate, Authentication</td>
<td>Login, Authentication, History, Form, Record, Send, Claims, Client, Log, Find, Assign</td>
</tr>
</tbody>
</table>
B. Accuracy of the Analyzed Approaches

The performance investigation in the previous section revealed that EA-Miner usually requires less time than Theme/Doc to support the analysis of crosscutting concerns in the requirements document. This section complements the previous analysis by investigating which approach presents the most accurate results. The motivation for this analysis is to understand to what extent the automatic process supported by the analyzed approaches can provide developers with accurate results.

Table II presents the accuracy measured for both systems. The meaning of the acronyms is given below the table. We distinguish the accuracy measures in terms of functional concerns, non-functional aspects, and functional aspects. Moreover, in order to illustrate how the results were collected, Table III illustrates the sets of concerns and aspects generated by both approaches, EA-Miner and Theme/Doc, in the context of the ATM system. The same data for the complaint system is available at the study website [36].

As we can observe, some concerns and aspects were just identified by EA-Miner while others can be revealed by using Theme/Doc. These differences highlight how each approach places different emphasis on certain types of concerns, leading to different accuracy rates. We discuss the data and our findings in detail as follows.

Identification of Functional Concerns. Our results have shown that the participation of requirements engineers can help to increase considerably the accuracy of the obtained results. This finding is supported in EA-Miner by the fact that around 33% of all automatically-detected candidates for functional concerns were confirmed by the requirements engineers. They were confirmed after the engineers applied the adequate filters and performed manual refinements. In the context of the Theme/Doc approach, the finding was also confirmed. However, this approach was more effective than EA-Miner during the identification activity in both target systems, reaching around 50% of precision. We suspect this superiority is due to the fact that the requirements engineers have to manually determine the themes and simultaneously starts the concern identification activity. However, the time consumed by this task makes the Theme/Doc approach considerably slower than the EA-Miner one (Section V.A) and, consequently, affects its performance. Therefore, our results suggested that there is not an optimal approach to identify the functional concerns.

Identification of Non-Functional Early Aspects. The accuracy results presented in Table II suggest that there is no major difference between the EA-Miner and the Theme/Doc approaches regarding the identification of non-functional early aspects. While the EA-Miner precision results (71%) did not vary from the complaint system to the ATM, the Theme/Doc presented an increase of both measures: precision (from 50% to 80%) and recall (from 40% to 73%). In particular, we observed that the accuracy of the both approaches when identifying the non-functional early aspects largely depends on the level of details provided in the requirements document. This means that certain non-functional early aspects will be identified by the used approaches if they are explicitly specified in the requirements document. For instance, the "FaultTolerance" non-functional aspect was only identified by EA-Miner in the complaint system. As the system registers the user complaints, its requirements document explicitly specifies the alternative behaviors in case of fault occurrences in the system. Consequently, the approach was able to identify the Fault-Tolerance as a non-functional aspect. However, this did not occur in the ATM system because there are pre-established fault policies for this kind of system. Therefore, it was not necessary to specify these policies in the requirements document. The same reason motivated that the "Confidentiality" non-functional early aspect was not identified by the EA-Miner approach in the complaint system.

Furthermore, we observe that the size of the requirements document may influence the accuracy of the used approaches when detecting non-functional early aspects. In particular, the bigger the document, the less accurate the results. As we can see in Table II, both analyzed approaches presented less precision and recall rates when analyzing the requirements of the complaint system. It is important to remember that the requirements of the complaint system are specified in 230 pages, while the requirements document of the ATM system contains 65 pages. The high number of pages in the requirements document of the complaint system lead both approaches to automatically identify redundant non-functional early aspects, impacting on their accuracy. Therefore, our analysis revealed that more accurate results could be achieved by decomposing the requirements document into smaller ones.

However, if we consider both time spent and accuracy measures, the EA-Miner approach outperformed the Theme/Doc approach. EA-Miner achieved similar (in the complaint system) or better (in the ATM system) accuracy, but required less time in general. The reason for this global superiority is the following: in the concerns identification activity, the EA-Miner approach uses the NFR knowledge base to identify potential non-functional aspects in the requirements document. During this activity the requirements engineers can apply filters and make use of selected guidelines to improve identification and accuracy. Examples of these guidelines are filtering viewpoints by their relative frequency, grouping viewpoints synonyms and with same root form. The Theme/Doc approach allows specifying synonyms in a semi-automatic way, based on theme clustering algorithms. However, this feature is much more sensitive to the problem at hand, the expertise of those involved, and their knowledge of the problem domain.

Identification of Functional Early Aspects. It is important to notice that the Theme/Doc approach, in addition to helping identify functional early aspects, also helps alert the development of possible conflicts between early aspects (e.g. form and history), as the result of the constant interaction with the requirements engineer. On the other hand, the EA-Miner approach is based on the fan-in [12] analysis over the requirements to quickly recognize functional early aspects. However, it greatly relies on the requirements engineers domain knowledge in order to determine the fan-in threshold used to exclude the crosscutting concerns.
V. Threats to Validity

Some imperfections of our study are identified. This section presents the possible threats to validity and the mitigations considered.

Construct Validity. The first construct validity that affects our study is the introduction of possible errors during the re-processing stage of the requirements specification document. Since the process was performed manually, the resulting document may not correspond to what it was originally intended to be. Hence, this might lead to inconsistencies in the results. We have minimized this threat, to the best of our abilities, by establishing a few guidelines. Other possible construct validity is use of an incomplete oracle list. However, we have mitigated this threat by considering an oracle list that resulting from the senior engineers discussions.

Conclusion Validity. We have two issues that threaten the conclusion validity of our study: the number of evaluated requirements and assessed approaches. We tried to minimize the first threat by using a complex requirements document. It comprises 230-page description and encompasses a rich set of widely-scoped crosscutting requirements (Section III.B). Regarding the second threat, our analysis was concerned with the use of two of the most well-known approaches.

Internal Validity. Threats to internal validity in our study could be related to the level of engineers’ experience. However, we try to mitigate this threat by involving in our study engineers with a deep knowledge about the AORE approaches they were responsible for using. Also, both involved engineers have a comparable level of experience in using this kind of approaches (Section III.B). The second threat could be associated with the engineers’ enthusiasm and availability to participate in our study. We tried to reduce this threat by considering those engineers with high interested in our investigation. Furthermore, the study was conducted in the most appropriate day according to the engineers’ schedule. Therefore, they were completely available to perform the requested task. The last internal validity is regarding the reliability of our oracle. However, the oracle lists were carefully built by considering the consensus of the senior engineers (Section III.B).

External Validity. There are some issues that could go against the generalization of the results. First, we have only considered two asymmetric AORE approaches, even though they are two of the most representative ones. Second, the study relied on the requirements documentation of a single software system. In order to mitigate this threat we have used a real web-application with rich and complex requirements documentation. We are aware that more studies involving other systems should be performed in the future. All our findings should be further tested in replications or more controlled replications of our study.

VI. Related Work

Empirical works in requirements engineering (RE) are broad in nature and cover different aspects of this area of knowledge. In [25], a requirements engineering process maturity model is evaluated and the business improvements obtained by the suggested modifications on the requirements process are analysed. In [3], the authors investigate the correlation of errors in the requirements phase and faults in a software system; they also conduct an empirical study with students to demonstrate that the findings can help quality and productivity.

However, few empirical studies exist that comparatively evaluate AORE approaches [2][4][6][13][17][22][33] regarding quality aspects, such as effectiveness or productivity and investigate the benefits and drawbacks brought to requirements practices. The novelty of those approaches is followed by works and studies demonstrating their use through small and controlled examples. For instance, in viewpoint-based [4][6], goal-based [24], and scenario-based [23] approaches examples are used to illustrate the applications of the approach in practice. However, these existing examples and evaluations do not enable us to understand the impact of AORE approaches in large and realistic requirements descriptions.

The problem of assessing different aspect-oriented approaches is mentioned in [13], although the focus was on code-level aspect-mining approaches. The researchers had to standardize their evaluation processes and result measures to validate the different techniques comparison. Even though this work handled challenges of a different level of abstraction (i.e., code), it was relevant in helping us to address similar problems.

In [22], viewpoint-based [4][6], goal-based [24], MDSoc (Multidimensional Separation of Concerns) [27] and AORA (Aspect-Oriented Requirement Analysis) [1] were compared regarding time-effectiveness and quality of outcome. As in [22], the automation provided by the EA-Miner tool and its reflection on the effort required by the approach stand out in the time-effectiveness measure conducted, and the viewpoint based AORE, although less time-consuming, showed an inferior quality of results. However, while [22] has several similarities with this work, significant distinctions must be observed: (i) the amount of documentation utilized in this work is fifteen times larger, helping the achievement of more realistic results, (ii) this paper also presents a comparative evaluation of EA-Miner with the Theme/Doc approach, thereby adding new results to those obtained in [22], and (iii) the main bottleneck in terms of effort was the identification of requirements model concepts activity, and not the structure requirements specification. The latter can be explained by the sheer size of the requirements document, since in the case of the EA-Miner and the Theme/Doc this activity did not seem to escalate well-enough (Section IV.A).

VII. Conclusions

Some of the goals of AORE approaches are: to automate the specification of the requirements documents and to reduce the time spent by the requirements engineer in understanding the problem domain. In this study we compared EA-Miner and Theme/Doc in two orthogonal dimensions: (i) total elapsed time spent during their various activities, measured by effort data in person-minutes (ii) the quality of their produced outcome, measured by a combination of the precision and recall
data of the concepts present in each requirements specification produced using the selected AORE approaches.

In the first dimension our study show that EA-Miner helps the requirements analyst to find transversal relationships that could be hardly discovered by examining the information obtained during the knowledge's extraction step. Furthermore, our results indicate that the performance of certain AORE approaches is more sensitive to the correctness and the size of the requirements document. For instance, EA-Miner required less time to identify the functional concerns when the requirement documentation presented high detail degree. On the other hand, in the second dimension, Theme/Doc achieved better results, based on the refinement of themes and requirements conducted with the help of the requirements engineer. From this case study one can conclude that EA-Miner is better equipped to assist the requirements engineer during the analysis phase, but Theme/Doc builds a strong foundation for the follow-up of the requirements during the development process.

The main advantage of Theme/Doc is to be part of the Theme approach. It keeps track of requirement changes, and also helps users refine the requirements during the software development process. EA-Miner promotes quick identification of abstract entities, and building the requirements model.

The evaluated approaches, when compared to the manually process, showed good quality. Nonetheless, both approaches require the assistance of the requirements engineer to achieve high-quality results. Thus, the domain knowledge of the problem still plays a decisive role in the success of AORE approaches.

REFERENCES