GT4CCI: An Approach Based on Grounded Theory for Crosscutting Concerns Identification in Requirements Documents

Larissa de Alencar Sobral, Lyrene Fernandes da Silva

Departamento de Informática e Matemática Aplicada (DIMAp)
Universidade Federal do Rio Grande do Norte
lariasobral@gmail.com, lyrene@dimap.ufrn.br

Abstract. When crosscutting concerns identification is performed on the activities involved in requirements engineering there are many gains in terms of quality, cost and efficiency throughout the lifecycle of software development. However, despite these gains, this identification faces several difficulties such as the lack of systematization and tools that support it and the difficult to justify why some concerns are identified as crosscutting or not, since this identification is often made without any methodology that systematizes and bases it. In this context, this paper proposes and evaluates an approach based on Grounded Theory, called GT4CCI, for systematizing the process of identifying crosscutting concerns in requirements document. Through the use of GT4CCI it is possible to better modularize the requirements document, make it more consistent, detect possible failures and improve traceability among requirements, adding significant gains in terms of quality and reliability to crosscutting concerns identification and to requirements engineering.

Keywords. Grounded Theory, Crosscutting Concerns Identification, Software Modularity, GT4CCI, Crosscutting Concerns Approach.

1 Introduction

According to [1], identifying and documenting crosscutting concerns in the beginning of the software lifecycle, in phases involved in requirements engineering, is essential. This action provides significant improvement in requirements traceability, facilitates the evaluation of the impact of changes in the system, facilitates requirements evolution and improves the modularization of the system, among other advantages. According to [2], despite all these favorable points, the crosscutting concerns identification in the initial stages of the software development process has been neglected in most software projects. This neglecting is mainly caused by the absence of habit of applying this kind of identification and by the lack of methodologies that basis this identification.

Considering the difficulties in identifying crosscutting concerns during the requirements engineering and, more importantly, the need to justify why some concerns are considered crosscutting, we present a new approach, called GT4CCI, which organizes and supports this process, making it more grounded. GT4CCI is based on the analysis process proposed by Grounded Theory [9], a renowned and well known methodology originate from the social sciences that enable qualitative analysis of data by codifying these data. The use of Grounded Theory (GT) adds the following significant gains in the process of crosscutting concerns identification:

- GT is based on contextual analysis of data, making analysis more complete and consistent, since it takes into account the context in which the concern is embedded;
- GT bases its results on the data existing on the document analyzed, consequently the results obtained through its use can be easily traced in this document;
- GT does not limit its use to documents that are previously structured in accordance with established standards. This means that GT allows the analysis of any requirements document already developed;
- GT does not restrict the data to be analyzed. This means that analysis is not limited to one type of concern, for example non-functional requirements or use cases, ignoring the other concerns present in the document. The use of GT allows the analysis of any relevant item, or part of the requirements document;
- There are tools that support to the process proposed by GT and that automate some of its activities. In this work, we have used Atlas.ti Tool [10].

GT4CCI approach uses the requirements document as input for the analysis and identification of crosscutting concerns. Thus, the main goal of GT4CCI is to systematize and improve the process of identifying these concerns. By the use of this approach is believed that there are gains in highlighting the intrinsic complexity of some concerns, which are scattered and tangled on requirements document, and warning the analysts that it is necessary to analyze these concerns more carefully.

Some approaches, such as Theme/Doc [3], DISCERN [2], Early-AIM [7] and CCCINPL [8], have been developed for the purpose of systematizing the identification of crosscutting concerns on the more initial steps of the software development process. However, these approaches have some limitations, best exposed in section 5 of this work. Consequently, GT4CCI is an alternative to these approaches, since the gain brought by the use of Grounded Theory decreases some of these limitations.

In order to present the GT4CCI approach, this paper is organized as follows. Section 2 presents a Toy Example, which serves as a basis example for demonstrating the use of GT4CCI. Section 3 presents the GT4CCI approach, detailing each of its steps and presenting the results generated from the use of the approach in the Toy Example. Section 4 presents an experimental study applied to evaluate the approach GT4CCI. Section 5 presents and compares some related work and GT4CCI. Finally, Section 6 contains final remarks and future work.

2 A Toy Example: Crisis Management Systems

In order to facilitate understanding and to demonstrate the process defined by the GT4CCI approach, the requirements document of Crisis Management Systems [11] is used as a Toy Example in the section below.

Crisis Management Systems (CMS) is a requirements document defined as a standard case study used by researchers on aspect- oriented modeling. This document was firstly defined and used in the Transactions on Aspect-Oriented Software Development VII (TAOSD), published in 2010 [11]. CMS describes the requirements and defines a system that helps identify, assess and deal with crisis situations, allowing communication between all parties involved in managing the crisis. This is done through the allocation and management of resources and also through access to information concerning the crisis, made by allowed users.

This requirements document is composed of eight sections. The first section presents an overview of the system. The second section describes all requirements for this system. Section 3 presents the feature model. In Section 4 are presented all use cases involved in the system. The following sections of this document present the Domain Model, the Activity Diagrams and the Informal Physical Architecture Description. Finally, in the last section, the Selected Design Models are presented.

The data considered most important in the document CMS were analyzed and coded by the GT4CCI approach in order to make the identification of crosscutting concerns. Thus, special attention was given to the sections relating to the description of use cases and detailed description of system requirements, since they are more geared to the objective of this work, containing the information about the requirements of this system. A part of the results from the application of GT4CCI in the Toy Example are presented and discussed in the following sections.

3 GT4CCI Approach

GT4CCI – acronym to Grounded Theory for Crosscutting Concerns Identification - is an approach based on the process of collection, analysis and data coding proposed by Grounded Theory (GT). GT is a methodology, arising from the social sciences, which is based on qualitative analysis of data and data coding in order to determine the relationships among them. Following and adapting the process proposed by this methodology, it is possible to extract information that facilitates the establishment of relations between relevant points of a requirements document. As a result, it is possible to identify which of these points are scattered and tangled, determining whether they may be said crosscutting concerns.

The process proposed by GT4CCI consists of five steps: Open Coding, Axial Coding, Selective Coding, Graph Analysis 4CCI and Results Table Creation. Figure 1 shows the flowchart of the process proposed by GT4CCI. It is important to note that the first three steps are original of the GT, while the last two steps are defined by the approach GT4CCI. These two steps are used to accommodate the identification and documentation of crosscutting concerns. Each of these steps are detailed and illustrated in the following subsections.

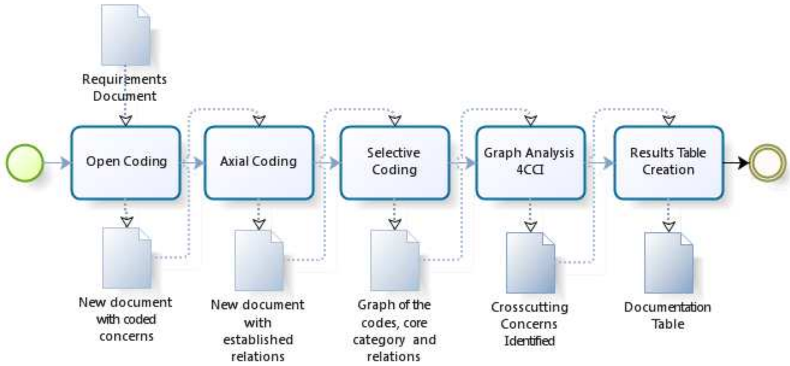

Fig. 1. Flowchart of the GT4CCI Process

3.1 Open Coding

GT4CCI approach is initiated by applying the Open Coding, that has the requirements document as input. In this step, all the relevant data in this document are analyzed, compared and coded. Consequently, all the requirements and other relevant information outlined in this document are analyzed and codes are created for each of these. These codes are created in order to identify and register the data considered relevant in the document analyzed and they are to be treated in the next step.

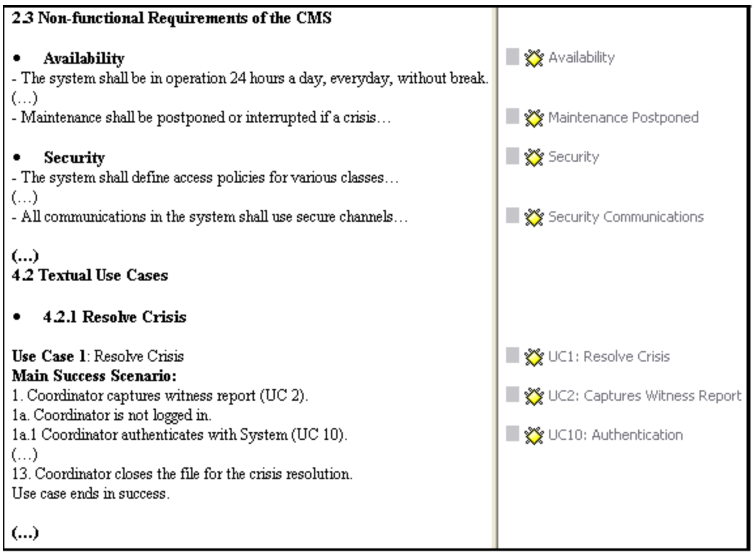

Fig. 2. Open Coding in CMS

In our Toy Example, codes were created to identify each concern specified by CMS document. Figure 2 presents the results generated by open coding in only a small part of this document. This part is related to two non-functional requirements (Availability and Security) and a one textual use case (Resolve Crisis). In the right side of Figure 2 are the codes created for each concern identified in this section. It is possible to see in this figure, for example, the code ‘Availability’, created for the concern Availability and the code ‘UC1: Resolve Crisis’, generated for the Use Case 1. Is worth mentioning that all concerns identified in the CMS document were also coded, although not shown in Figure 2.

3.2 Axial Coding

After establishing codes during the Open Coding, presented in the previous subsection, the coded requirements document is submitted to the Axial Coding. In this step are establishes the relations between the codes previously created.

These relations are established through connectors. Each connector identifies the type of relationship between two codes. In GT4CCI two special connectors are used: the connector ‘is part of’, that indicates that a code is tangled within another and the connector ‘is in’, that indicates that a code is scattered. The establishment of these relations in the document, however, is done by the user with the support of Atlas.ti tool, once it provides a Codes Manager, which supports the establishment of relationships between these codes. For instance, considering the CMS, the relation between the codes 'UC1: Resolve Crisis' and 'Security' is established. In accordance to information extracted from the requirements document, 'UC1: Resolve Crisis' is related to 'Security' through connector "is part of" which means that UC1 is tangled within non-functional requirement Security.

Is worth noting that, in addition to the relationship between these two codes, relations with other concerns of CMS were also established. These relationships are explained by the graph, generated for each of these codes, in Selective Coding.

3.3 Selective Coding

After Axial Coding, the Selective Coding is applied to the requirements document, which were defined the relationships among codes. At this stage of the process (established by Grounded Theory), the entire coding process by which the document in question has already been submitted is refined. This refinement consists in analyzing the whole document and codes defined and, thereafter, set the core category. The core category is the most relevant code of analysis, from which a graph is generated, showing all relations between this and other codes, established in the preceding steps.

It is also important to note that GT4CCI treats each code individually, in order to facilitate the understanding and visualization of relationships established with them. This means that each code is subjected to Selective Coding in an individual way, so as to be regarded as core category of analysis. Consequently, a graph presenting the results of coding process is generated for each category. Figure 3 illustrate the graphs generated for two categories that we defined for CMS, ‘Security’ and ‘UC1:Resolve

Crisis’, respectively. These graphs were automatically generated for Atlas.ti tool, and explained in next section.

The graphs generated in this step are analyzed in detail in the next step of the GT4CCI: Graph Analysis 4CCI.

3.4 Graph Analysis 4CCI

In this step, entirely conceived by GT4CCI approach, the graphs generated to the core categories are carefully analyzed in order to identify crosscutting concerns. This analysis is based on the relationships between the codes presented in graph in order to identify and determine whether the core category may or may not be said a crosscutting concern. For the correct identification, GT4CCI sets some basic guidelines:

- The identification of crosscutting concerns is made by verifying scattering and tangling of the codes defined for a requirements document;
- A concern is considered scattered when its specification is necessarily scattered between many others concerns (whether requirements, use cases, functionalities, etc.) of the same document. This scattering is represented by at least two relations ‘is in’ between the core category and other codes;
- A concern is considered tangled when its specifications is interleaved with the specifications of others concerns (whether requirements, use cases, functionalities, etc.) in some parts of the same document. This tangling is represented by at least two relations ‘is part of” between the core category and other codes;
- Thus, GT4CCI approach considers crosscutting concern, that concern is the origin point of at least two relations ‘is in’ and the target of at least two relations ’is part of’.

It is essential to highlight that the data analyzed are not restricted, for example, to the description of the requirements and use cases, as in many other approaches. In the case of GT4CCI approach, any data considered relevant within the document can and should be analyzed. The more points in the document are analyzed, the higher the quality and reliability of the conclusions reached at the end of the analysis process.

The left top of Figure 3 presents the graph generated from the step of Selective Coding to the code 'Security'. From the analysis of this graph, it is possible to see that the core category 'Security' is scattered as it is associated with many other concerns of this system through 'is in' relations. Thus, this category can be said tangled, since it is target of many ‘is part of’ relations. This way, according to the concepts adopted by this approach and considering only a part of CMS requirements document, the non-functional requirement Security is a crosscutting concern, since it can be considered both tangled and scattered.

Continuing with the CMS Toy Example, the right top of Figure 3 shows the graph generated for 'UC1: Resolve Crisis'. In this case, can be seen that 'UC1: Resolve Crisis' is considered tangled, since it is associated with many others concerns of this system through connector 'is part of'. This core category is also considered to be scattered, since it is the origin of many ‘is in’ relations. Thus, it is possible to affirm that the Use Case 1, present in the CMS, is a crosscutting concern since it has both the scattering and tangling characteristics.

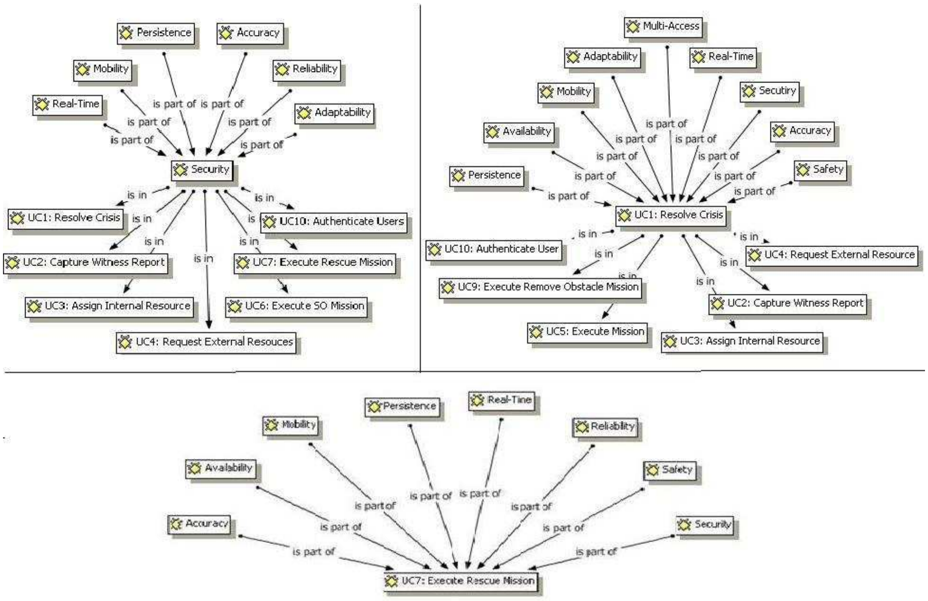

Fig. 3. Graphs of core categories ‘Security’, ‘UC1: Resolve Crisis’ and ‘UC7: Execute Rescue Mission’

Finally, the bottom of Figure 3 shows the graph generated for 'UC7: Execute Rescue Mission’, from the CMS documents. From analysis of this graph, it is possible to see that the core category 'UC7: Execute Rescue Mission' is tangled as it is target of many 'is part of' relations. However, this category cannot be said scattered, since it is not associated with any other concern of this system by any ‘is in’ relation.

This way, according to the concepts adopted by this paper, it is possible to affirm that ‘Use Case 7: Execute Rescue Mission’ is not considered a crosscutting concern, since it cannot be considered scattered.

3.5 Results Table Creation

After identifying, in the previous steps, the final step proposed by GT4CCI is the Creation of a Results Table. The purpose of this Results Table is to document, objectively, all data resulting from application of the approach in a requirements document, enabling a simpler and agile query of these data when necessary in requirements validation and verification or in subsequent stages of software development.

This table contains four columns: 'Concern', Scattered', 'Tangled’ and 'Crosscutting Concern'. Table 1 illustrates a small example of the results table created to CMS. The column 'Concern' contains the core category analyzed, to which the other fields will be directly related. The column ‘Scattered' lists in which other concerns of the system that core category is scattered. The field ‘Tangled’ lists all the other concerns that the core category is tangled within. Finally, the Column 'Crosscutting Concerns' indicates, in an objective way, if the core category in question may or may not be considered crosscutting concern. It is worth noting that the names of the columns, consisting of very simple terms, are done so in order to facilitate the identification and understanding during any future queries made to this documentation.

Table 1. Part of the Results Table of Toy Example

Concern	Scattered	Tangled	Crosscutting Concern
Security	UC1: Resolve Crisis UC2: Capture Witness Report UC3: Assign Internal Resource UC4: Request External Resource UC 6: Execute SuperObserver Mission UC7: Execute Rescue Mission UC10: Authenticate Users	Adaptability Accuracy Mobility Persistence Real-Time Reliability	**Yes**
UC1: Resolve Crisis	UC2: Capture Witness Report UC3: Assign Internal Resource UC4: Request External Resource UC5: Execute Mission UC 9: Execute Remove UC10: Authenticate Users	Accuracy Adaptability Availability Mobility Multi-Access Persistence Real-Time Safety Security	**Yes**
UC7: Execute Rescue Mission	-	Accuracy Availability- Mobility Persistence Reliability Safety Security	**No**

Table 1 shows only a part of the Results Table generated for CMS requirements document, illustrating part of the final results, since the CMS document is very complete and extensive. In this table, it is possible to see three concerns. One of these concerns is a non-functional requirement and two functional requirements. The first, Security, is considered scattered and tangled, and for this reason it is considered a crosscutting concern. The second and third concerns are use cases, representing functional requirements, called Resolve Crisis and Execute Rescue Mission, respectively. UC1: Resolve Crisis is tangled within and scattered among the system, and for this reason this concern is considered a crosscutting concern, while UC7: Execute Rescue Mission is considered tangled, but not scattered, and therefore is not considered a crosscutting concern. Once again it is emphasized that this is just an excerpt of Results Table generated for the CMS. With the use of the entire table is possible to make a quick identification of all the concerns of this system, as well as analyzing in which part of the requirements document these can be said scattered and tangled, and especially check whether a concern can or cannot be considered crosscutting.

During the application of the GT4CCI approach in the requirements document CMS, some points might be highlighted. 21 concerns were analyzed in the CMS document, thus defined as core categories of analysis. 11 of these concerns are related to non-functional requirements, while 10 are related to use cases of the system. During the analysis of the CMS document were defined more than 150 codes and over 200 relationships were established.

With this, it was possible to analyze in detail each concern identified in CMS. By applying this approach in the document CMS was possible to identify the scattering and tangling of these concerns in the system and determine whether it may or may not be considered crosscutting concerns. With these results, it is possible to understand which concerns of this system need to be better modularized in order to add quality to it. Furthermore, by the results obtained through the use of GT4CCI is possible to call attention to possible errors in the development of the requirements document analyzed, such as poor specification of a requirement, for example.

Thus, beyond simply identifying if a concern can or cannot be said or crosscutting, GT4CCI approach also aims to better view the requirements document, highlighting potential problems and issues that should be better understood and analyzed.

4 Experimental Study

In order to evaluate GT4CCI, the approach proposed by this paper, an experimental study was designed and implemented. The main objective of this study is to evaluate the correctness of the results presented from the use of the approach GT4CCI. Details of the context and execution of this study and the analysis of their results are presented below.

4.1 Context and Implementation of the Experimental Study

Nine postgraduate students in Systems and Computing at the Federal University of Rio Grande do Norte voluntarily participated in this experimental study. These participants were divided into two groups in order to evaluate and use the approach in two distinct scenarios, since each of the groups performed this experimental study using different requirement documents. Group I performed this experimental study using the Methodology Explorer [15] system requirements document as a basis for the analysis, while Group II used the Meeting Scheduler [16] system requirements document.

The sectioning of participants in two groups was made according to their previously declared level of knowledge and experience in Requirements Engineering and in Identification of Crosscutting Concerns, before the execution of the experimental study, by each one of the participants. The distribution of these participants into the groups was made attempting to establish equivalent groups in terms of the previously declared knowledge held by them.

Before the experimental study's implementation, a training with all the participants was conducted, aiming at exposing some of the basic knowledge needed to this study, such as the definition of crosscutting concerns, the explanation of the process proposed by Grounded Theory methodology and the exhibition of examples, both theoretical and practicals, of the identification of these in a requirements document. This training strived to reduce the difference in the level of knowledge presented by the participants. Also in this training, the GT4CCI approach and the Atlas.ti were introduced. Each step of this approach was minutely explained and demonstrated, through a simple example, using the Atlas.ti tool. During and immediately after this training, the participants had the opportunity to clarify their doubts regarding the use of both the approach and the tool.

In order to perform this experimental study, a sequence of activities was performed by the participants in a full and sequential fashion, thus ensuring more safety to the resulting data. Firstly, all participants were required to access the Atlas.ti tool and the requirements document to be analyzed. Then each one of the participants individually made a preliminary reading of the requirements document. Shortly thereafter, the participants, together, applied the Open Coding phase in the requirements document they were analyzing. The application of this group phase is made with the objective of assuring that, in future phases, the participants will work with the same concerns defined in this first stage. The individual application of the Open Coding could lead to distinct definition and identification of interests by each participant, which would produce a less coherent result and a less accurate analysis. Finally, the concerns identified in the previous phase were equally divided among the participants and each one of them applied, individually, the next phases defined by the GT4CCI approach, aiming to ascertain whether each of the identified interest could be defined as crosscutting.

In the end of the experimental study implementation, all the resulting data of the GT4CCI approach application were collected to be analyzed. The analysis of these data will be presented in the following section.

4.2 Analysis of Results

Two metrics, recall and precision, were utilized in order to evaluate the correctness of the obtained results through the use of GT4CCI approach. Both of these metrics are hereby utilized for displaying efficiency in relation to the correctness of data, as well as for being extensively employed in other experimental works of similar nature.

According to [17], Recall can be defined as a metric utilized to evaluate the plenitude of the obtained data. Applying this metric into the context of the study, we can state that Recall evaluates the rates of plenitude presented by the identification of concerns done through the use of GT4CCI approach. In other words, the amount of actual concerns existent within a certain requirements document that this approach was able to identify. The metric of Recall is calculated through the following formula:

$$Recall = \frac{Elements\ Correctly\ Identified}{Correct\ Elements}$$

Still in accordance with [17], Precision can be defined as a metric utilized to evaluate precision, or fidelity of obtained data. Therefore, applying this metric into the context of this study, we can state that the Precision metric evaluates the rates of fidelity presented by the identification of concerns made through the use of GT4CCI approach, that is, how many concerns identified by the approach corresponds exactly those concerns actually in the document. The Precision Metric is calculated through the following formula:

$$Precision = \frac{Elements\ Correctly\ Identified}{Identified\ Elements}$$

It is worth noting that the definition of the correct elements used as the basis of this experimental study were defined by the author of this work in conjunction with an expert in crosscutting concerns identification. The analysis of the collected data through the application of the experimental study enabled us to grasp some relevant information. First, it is possible to observe that Group I identified 18 concerns in the requirements document Methodology Explorer. 13 of these were functional concerns, while the other 5 were non-functional concerns. The identification of functional concerns through the GT4CCI approach in the Methodology Explorer requirements document, utilized by Group I, resulted in a rate of Precision and Recall of 100%. This demonstrates that all the existing functional concerns within this document were correctly identified by GT4CCI approach, without neglecting any one of them. The same occurred with the identification of both non-functional concerns and non-functional crosscutting concerns. On the other hand, it is possible to find a Recall rate of 100% amongst the identification of functional crosscutting concerns, although the encountered Precision rate is 75%. This also occurred with the case of identification of crosscutting concerns (both functional and non-functional). This identification obtained a Recall rate of 100% as well. Albeit the Precision rate encountered was 89%.

Employing the same analysis regarding the application of GT4CCI approach in the Meeting Scheduler requirements document, utilized as an analytical artifact by Group II, it is possible to highlight certain points. Group II identified 21 concerns in the requirements document Meeting Scheduler. 13 of these were functional concerns, while the other 8 were non-functional concerns. One may observe, then, that the identification of functional concerns, the identification of non-functional concerns and the identification of non-functional crosscutting concerns present a Precision rate of 100% along with a Recall rate of 87,5%. Thus implying that all of the existing concerns in these types of documents were correctly identified through the use GT4CCI approach. Having said that, one may conclude through the Recall rate of 87,5%, that the GT4CCI approach neglected a few interests of such types existent in the requirements document. Furthermore, the case of identification of functional crosscutting concerns made by Group II in this experimental study, permits one to observe that the Precision and Recall rates were 78%. In the case of crosscutting concerns (both functional and non-functional) we encountered a Precision rate of 87,5% and a Recall rate of 82%.

In face of these results, it is possible to conclude that the GT4CCI approach provides results with good correctness, considering that in both analyzed scenarios the rates of Recall and Precision never appeared inferior to 75% in any one of the identification of interests. These numbers reflect every type studied, having presented in several situations Recall and Precision rates of 100%, demonstrating identifications without error or negligence. Further information on this experimental study can be seen in [18].

5 RELATED WORKS

With the perception of the benefits that come from identification and documentation of crosscutting concerns in the earliest phases of software lifecycle, some approaches that systematize this identification were developed. Some of these are applied to a specific type of requirements model, such as i* [13], AOV-graph [12], BPMN [14] and Use Cases [4, 5, 6], while other analyze textual requirements documents. Among the principal of these approaches, four are highlighted by this paper: Theme/Doc [3], DISCERN [2], Early-AIM [7] and CCCINPL [8].

Each of these approaches has characteristics that are individual to them. These characteristics may sometimes represent limitations or gains of an approach in relation to another. Table 2 is a comparative table that allows easy viewing the main differences and similarities presented by each of these approaches and GT4CCI.

Table 2. Approaches Comparison Table

Approach	Identification Technique	Artifacts Analyzed	Requirements Analyzed	Automation and Tools
GT4CCI	Contextual Analysis	Any Req. Document	Analysis of All Relevant Data	Semi-automatic. Atlas.ti Tool
Theme/Doc	Contextual Analysis	Structured Req. Document	Any Textual Req.	Semi-automatic. Theme/Doc Tool
DISCERN	Contextual Analysis	Any Req. Document	Non-functional Textual Req.	Without automation. Does not cite any tool.
Early-AIM	Natural Language Processing	Any Req. Document	Any Textual Req.	Semi-automatic. EA-MINER Tool
CCCINPL	Natural Language Processing and Contextual Analysis	Any Req. Document	Any Textual Req.	Automatic. C3I Tool

In Table 2, four comparison criteria were established. These criteria are: technique used to identify crosscutting concerns, artifacts analyzed to proceed the identification, kind of requirements analyzed by the approach and the level of automation of the approach and what tools it uses to support its process. From this table, we can observe some interesting points. The first of these points, referring to the identification technique, shows that most approaches perform the contextual analysis of the requirements document. However, Early-AIM approach does not make this kind of analysis, disregarding the context in which the concern analyzed is embedded.

Another point worth mentioning is the artifacts analyzed. Theme/Doc approach, for example, only makes the analysis of requirements documents previously structured by a predefined pattern by this approach. This way, makes its use impractical in documents developed without this pre-defined structure.

The requirements analyzed by the approach are also a relevant point of this comparison. It can be noticed that one of the approaches analyze just one kind of requirement, the non-functional requirements, limiting the analysis. Moreover, a fact that deserves mention is the possibility of analysis of any relevant data, provided by GT4CCI. This means that this approach not only analyzes textual requirements, but also enables the analysis of the use-case diagram, for example.

Finally, another point worth mentioning is the level of automation of these approaches. DISCERN approach have no automation and do not have tool that supports the any activity of its process. This makes the application of the approach generally slower and less reliable. In contrast there is also a fully automatic approach, the CCCINPL approach. The total automation provides certainly gains in celerity. However, the analyst is unable to follow the process of identifying crosscutting concerns and is unable to see possible flaws in the requirements document and in the analysis, since the approach returns only the final result of analysis.

6 CONCLUSIONS

This paper briefly introduced GT4CCI approach. This approach aims to organize and assist the identification of crosscutting concerns in the initial stages of the software development process using the requirements document as artifact for analysis.

Using the approach GT4CCI is possible to identify the crosscutting concerns of a system through a qualitative analysis of data, based on the context in which the concerns are embedded into the requirements document. Thus, it is possible to identify crosscutting concerns in any requirements document by analyzing all data deemed relevant in this document.

Moreover, the results obtained by applying GT4CCI are strongly based on data present in the document analyzed, and thus can be traced and more easily justified. Therefore, through the benefits arising from the use of GT4CCI approach, it is expected that it provides a better view of the requirements document, highlighting potential problems and issues that should be better understood and analyzed. Moreover, from the results found from the execution of the experimental study reported in this work, we can conclude that GT4CCI approach presents a good correctness of the results, once showed high values of recall and precision metrics in both scenarios evaluated. This ensures the quality and reliability found from the use of GT4CCI approach in identifying crosscutting concerns in requirements documents.

As future work, we suggest performing experimental studies to measure GT4CCI performance compared to others existing approaches. From the results of these experiments, it is expected to make possible improvements in this approach and expand the use of GT4CCI for other kinds of artifacts developed during the software lifecycle. With this, it will be possible to track and analyze whether these crosscutting concerns are propagated in several activities of software development or to analyze the nature of crosscutting concerns in each of these activities.

References

1. Ali, B.S., Kasirun, Z.M. A Review on Approaches for Identifying Cross-cutting Concerns. In Advanced Computer Theory and Engineering, ICACTE '08. 2008.
2. Rosenhainer, L. The DISCERN Method: Dealing Separately with Crosscutting Concerns. In Proceedings of OOPSLA Early Aspects 2005. San Diego, USA, 2005.
3. Baniassad, E., Clarke, S. Finding Aspects in Requirements with Theme/Doc. In Proceedings of Early Aspects 2004 (AOSD 2004). Lancaster, United Kingdom, 2004.
4. Araújo, J., Moreira, A., Brito, I., Rashid, A. Aspect-Oriented Requirements with UML. In Workshop on Aspect-Oriented Modelling with UML (UML 2002). Dresden, Germany, 2002.
5. Jacobson, I., Ng, P. Aspect-Oriented Software Development with Use Cases. Addison-Wesley. 2005.
6. Brito, I. Aspects Oriented Requirements Engineering. In 7th International Conference on Unified Modelling Language (UML 2004). Lisbon, Portugal, 2004.
7. Sampaio, A., Rashid, A., Rayson, P. Early-AIM: An Approach for Identifying Aspects in Requirements. Poster Paper. In Requirements Engineering Conference. Paris, France., 2005.
8. Ali, B. S., Kasirum, Z. M. Crosscutting concern identification at requirement level. In Malaysian Journal of Computer Science, vol. 21(2), pp.78-87, 2008.
9. Glaser, B.; Strauss, A. The discovery of Grounded Theory: Strategies for Qualitative Research. New York: Aldine Transaction, 1967.
10. Atlas.ti. Version 6.2. Available: <http://www.atlasti.com>. Accessed in: 28 oct. 2011.
11. Kienzle, J., Guelfi, N., Mustafiz, S. Crisis Management Systems: A case study for aspect-oriented modeling - Technical Report. McGill Univ. 2009.
12. Yu, Y., d. P. Leite, J.C.S., Mylopoulos, J. From goals to aspects: Discovering aspects from requirements goal models. In *Proceedings of the 12th IEEE International Requirements Engineering Conference*. 2004.
13. Alencar, F., Castro, J., Moreira, A., Araujo, J., Silva, C., Ramos, R., Mylopoulos, J. Intgration of Aspects with i* Models. In *Agent-Oriented Information Systems IV*. 2008.
14. Cappelli, C., Leite, J.C.S.P., Batista, T., Silva, L. An aspect oriented approach to business process modeling. In *Proceedings of the 15th workshop on Early aspects, EA '09*. ACM. New York, NY, USA, 2009.
15. Silva Júnior, C.R., Perrelli, H., Mesquita, S. Documento de Requisitos do Sistema Methodology Explorer. Available: http://www.cin.ufpe.br/~mexplorer, 2001.
16. van Lamsweerde, K., Darimont, R., Massonet, P. The Meeting Scheduler System – Preliminary Definition. Internal Report. Université Catholique de Louvain, 1993.
17. Anacleto, A.C.S. Aplicação de Técnicas de Data Mining em Extracção de Elementos de Documentos Comerciais. Tese de Mestrado. Universidade do Porto. Porto, Portugal. 2009.
18. Sobral, L.A. GT4CCI: Uma abordagem para identificação de Interesses Transversais em Documentos de Requisitos. Dissertação de Mestrado. UFRN. Natal, Brasil. 2013.