Discovering Aspects in Requirements with Repertory Grid

Nan Niu
Department of Computer Science
University of Toronto
Toronto, Ontario, Canada M5S 3G4
nn@cs.toronto.edu

Steve Easterbrook
Department of Computer Science
University of Toronto
Toronto, Ontario, Canada M5S 3G4
sme@cs.toronto.edu

ABSTRACT

The dominant decomposition at the requirements level relies on how requirements are represented and modeled. An aspectual requirement is a broadly scoped concern that cuts across and has impacts on other requirements-level concerns or artifacts. This paper presents a novel use of Repertory Grid Technique with roots in psychology of personal constructs as a systematic and effective way to support analysts for viewing and manipulating requirements models to expose how entities relate to one another, thereby facilitating aspectual requirements identification and conflicts detection. We illustrate the approach with a proof-of-concept example adapted from the literature; in particular, we show how early aspects can be discovered in goal models, and how interference can be detected in viewpoints-based models.

Categories and Subject Descriptors: D.2.1 [Software Engineering]: Requirements — *elicitation methods*

General Terms: Design

Keywords: Early aspects, repertory grid technique

1. INTRODUCTION

Aspect-oriented programming technologies aim to improve system modularity by modularizing crosscutting concerns [11]. Global design and programming issues can lead to aspects, such as error handling, data integrity, efficient use of memory, security, and the like. The underlying premise of aspect-oriented programming is that the more modular the system, the easier it is to produce, maintain, and evolve [12].

As suggested by its name, aspect-oriented programming methodology focuses on the solution domain: developers identify and capture aspects mainly in source code. Recent work on *early aspects* [1] aims to focus on the problem domain, which is inhabited by customers, users, and other constituencies affected by a software-intensive system. Research in this area can help to improve modularity in the requirements and architecture design and to detect conflicting concerns early, when trade-offs can be resolved more economically [2].

[1]http://www.early-aspects.net/ Last accessed on March 13, 2006.

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

The dominant decomposition at the requirements level rests on how requirements are represented and modeled. There are many ways to organize requirements such as (structured) natural languages, use cases, viewpoints, goal models, and software requirements specifications. A crosscutting requirement, then, is a concern that cuts across other requirements-level concerns or artifacts conforming to the organizing structure chosen by the requirements analyst. It is broadly scoped in that it is found in and has an (implicit or explicit) impact on more than one requirements artifact [2].

Aspect-oriented requirements engineering is concerned with identification, modularization, representation, and composition (weaving) of crosscutting requirements [15]. In an existing set of requirements, it is not likely to be clear what aspects are present [2]. Using intuition and domain knowledge is not necessarily sufficient for identifying the potentially broad range of aspects within a reasonable amount of time [1]. Our aim is to develop a systematic and effective method to support analysts for viewing and manipulating requirements models to expose how entities relate to one another, thereby facilitating aspectual requirements identification and conflicts detection.

Our approach is based on Repertory Grid Technique (RGT) [8] with roots in the psychology of personal constructs [10]. RGT aims to build a model of a particular domain of knowledge by enabling people to articulate how they perceive certain factors within the range of convenience. It is used to understand an individual's personal (i.e., idiosyncratic) construction of his or her environment (e.g., artifacts, events). This leads us to consider RGT as a candidate method for capturing how stakeholders construe the problem world and the machine [9], what their desires and needs are, and how these concerns correlate with each other.

We assume there exists a relatively well-organized set of requirements derived from some dominant decomposition criterion. Our task is to gain an early understanding of these requirements and the (crosscutting) concerns they address. This vision is influenced by the work on "weaving together requirements and architectures" [13], which suggests an agenda "from early aspects to late requirements" because identifying aspects too early is counterproductive [14]. In this paper, we present the use of RGT as a means of systematically capturing and analyzing crosscutting entities in requirements models.

Section 2 presents background to RGT. Section 3 illustrates how to leverage RGT to discover early aspects in requirements models through a detailed example adapted from the media shop study [4, 20], where stakeholder goals and intentions are modeled in the i^* framework [19]. In particular, we discuss exposing crosscutting concerns based on two different decomposition criteria: requirements goal models and viewpoints-based models. We review related work in Section 4, and conclude the paper in Section 5.

2. REPERTORY GRID TECHNIQUE

George Kelly's Personal Construct Theory (PCT) [10] assumes that the meaning we attach to events or objects defines our subjective reality, and thereby the way we interact with our environment. Constructs are ways of construing the world, enabling people to respond to what they experience in ways which are "explicitly formulated or implicitly acted out" [10]. For example, the way in which I interact with my desk is determined by the way I construe it — do I polish it carefully because I see it as something to be looked after or do I put my feet up on it because I see it as a convenient resting point? Thus, in Kelly's theory, the idea of the notion of 'objectivity' disappears, and the best we can do along these lines is 'inter-subjectivity', thinking rather of a dimension representing degree of agreement between construers and degree of certainty of judgment [18].

Kelly originally developed PCT in the context of psychotherapy and developed an associated methodology, the Repertory Grid Technique (RGT), so as to explore patients' constructions of their social world. However, RGT has long been recognized as a content-free method for externalizing individuals' personal constructs, and has seen applications in a wide variety of situations, for example, education and market research, which are far removed from clinical psychology.

Underlying RGT is the notion that enables people to verbalize how they construe certain factors within the area of interest. These verbalizations are known as *constructs*, and the factors are called *elements*. A construct is hence a single dimension of meaning for a person allowing two phenomena to be seen as similar and thereby as different from a third [8]. A construct is bipolar in nature, where each pole represents the extreme of a particular view or observation. Kelly suggested RGT as a structured procedure for eliciting a repertoire of these conceptual constructs and for investigating and exploring their structures and interrelations [8].

As an example, the area of interest might be how people feel about certain information sources. In this example, the elements would be various information sources, such as *TV*, *Newspaper*, *Radio*, *NewsGroup*, and so forth. To elicit constructs, the person is initially asked to consider a "triad" of three elements and asked to say how two of the three seem similar and how the third differs. For example, presented with the triad (A) TV, (B) Newspaper, and (C) NewsGroup, the person may say that A and B have many focuses, whereas C is singly focused. The construct ranging from "many focuses" to "single focus" can be considered as a rating scale (typically ranging from 1–5), and each element can now be assigned a rating on that construct. The same triad might elicit more constructs. For instance, the person may say that B and C are text-based, while A delivers multimedia services.

The triads are chosen such that no pair of elements appears in more than one triad, e.g., if "A, B, C" is presented,

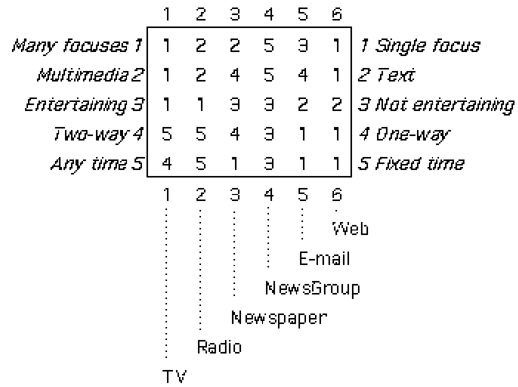

Figure 1: A sample repertory grid

then no other triad will be presented, which includes the pairs "A, B", "A, C", or "B, C". The reason for this is to maximize the participant's opportunity to present different constructs. As more and more constructs are generated using different triads and the elements rated on them, a picture can be built up of an individual's views of the domain.

A sample repertory grid illustrating the "information sources" scenario is shown in Figure 1, in which rows contain constructs and columns represent elements. For a greater degree of differentiation, a five-point scale is used to indicate where an element lay with respect to the poles of each construct: The construct poles to the left of the grid are the "1" end of the scale, and those to the right are the "5" end. The occurrence of the central point "3" in a grid can have two different interpretations: i). Neutral: the element is at neither one pole nor the other of the construct. ii). Not applicable or unknown: the element is outside the range of convenience of the construct. For example, the element 'NewsGroup' is not pertinent to the construct 'Entertaining-Not entertaining'. Therefore, a rating of "3" appears in the third row and the fourth column of Figure 1.

The most interesting feature of RGT is the wide variety of different types of analyses that can be applied to the gathered personal constructs. RGT provides data that lend to the identification of general needs, beliefs and attitudes, and specific inter-artifact differences, as well as inter-person or inter-group differences [8]. In this paper, we mainly focus on cluster analysis to show how early aspects can be identified in requirements models. The interested readers can consult the recent book [8] involving discussion on most commonly used grid analysis methods and tools.

The FOCUS program [18] can be used to perform a two-way hierarchical cluster analysis and reorder the grid so that similarly rated elements are adjacent and similarly used constructs are adjacent. As the program performs the clustering, it builds dendrograms that illustrate the strength of association between elements and between constructs [18]. A dendrogram is strictly defined as a binary tree with a distinguished root, that has all the data items at its leaves. The internal nodes in a dendrogram are used to show the hierarchical clustering results of the given data items.

The reordered "information sources" sample grid is shown in Figure 2, in which the upper and lower dendrograms demonstrate the relationships between constructs and relationships between elements respectively. To highlight the clusters within the grid, ratings of 4 and 5 are given dark shading, ratings of 3 are given light shading, and ratings of 1

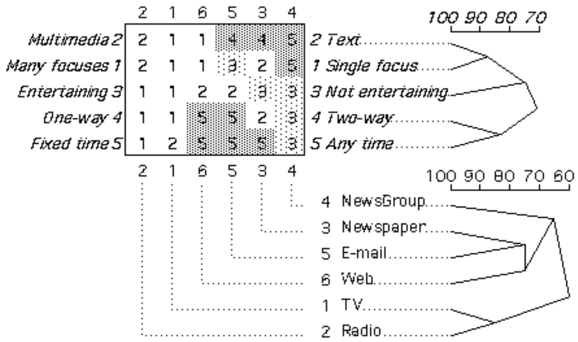

Figure 2: Cluster analysis of the sample grid

and 2 are left unshaded. This enables the easy identification of "blocks" within the grid [18].

Crosscutting concerns in the grid can be spotted in many ways. For example, the elements with ratings 1 or 2 in the fifth row of Figure 2, namely 'Radio' and 'TV', are likely to have the same concern of delivering information during some 'Fixed time', and the ones with ratings 4 or 5 in that row – 'Web', 'E-mail', and 'Newspaper' – share the opposite pole of being able to provide information at 'Any time'. The shading schema and the reordering mechanism provided by the FOCUS program greatly facilitate the ability to visualize concerns that span through some of the entities in the grid. Another approach is to analyze closely clustered elements (constructs) so as to unearth the reason behind this connection. The inferred factors may give rise to sensible concerns that cut across that particular group of entities.

A repertory grid in itself is the outcome of a successful application of the technique. We speculate that RGT combined with diverse analysis methods can have a huge potential to uncover and exhibit crosscutting concerns in a particular context. On one hand, RGT unveils the perceptions and concerns about the person who uses it. On the other hand, it also reveals information about the objects under consideration, i.e., their attributes and properties. From the early aspects standpoint, we are interested in analyses in both prospects.

3. USING RGT TO CAPTURE ASPECTUAL REQUIREMENTS: AN EXAMPLE

3.1 Goal Models

We use the media shop e-business example [4] to illustrate how to leverage RGT to discover early aspects in requirements models. The example adopted the i^* organizational modeling framework [19]. Intentions of stakeholders are modeled as goals which, through some form of goal-oriented analysis, eventually lead to the functional and non-functional requirements (NFR) of the system-to-be [4].

In this study, we started with some requirements goal models that resulted from previous elicitation activities [20]. Figure 3 shows a model slice for the media shop, in which softgoals, tasks, and their relationships are captured using i^* notations. These well-organized (graphical) models are of great value in understanding how stakeholders construe the problem world and the machine, since much effort has been devoted to analyzing raw inputs and conducting the organizational modeling. Instead of following the "triad" method (futilely) trying to elicit constructs, we extracted

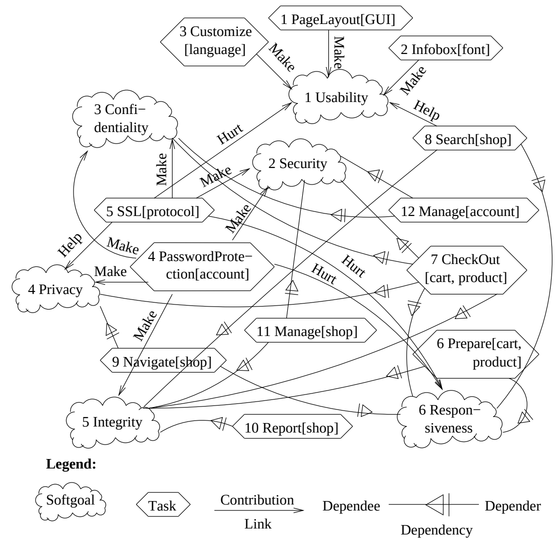

Figure 3: Goal model for the media shop

basic building blocks of repertory grid from existing models such as the one shown in Figure 3. We treated concrete tasks as the common set of elements for grid analysis, and regarded softgoals as constructs to explore candidate aspectual requirements. This is because aspects are usually "units of system decomposition that are not functional" [11] and NFRs are typically expressed as softgoals to suggest that generated software is expected to satisfy within acceptable limits, rather than absolutely, the NFRs [5]. Our purpose of using RGT in this context is not to elicit constructs in the general sense, but to analyze interrelationships among model entities so that early aspects can be identified accurately and their properties can be revealed thoroughly.

From Figure 3, twelve tasks (elements) and six softgoals (constructs) were identified and numbered. Each construct was specified by a pair of polar extremes: 'make the goal' and 'break the goal'. Each element was then rated on each bipolar construct. A five-point scale was defined to make such measures both subtle and specific:

1 – break (strong negative)
2 – hurt (weak negative)
3 – neutral (unknown or don't care)
4 – help (weak positive)
5 – make (strong positive)

Figure 4 shows the cluster analysis result of the repertory grid derived from Figure 3. There were $6 \times 12 = 72$ slots in the resultant grid. The values of the slots correspond to the "contribution links" in Figure 3 could be obtained directly. For example, a rating of "5" appeared in the upper-left corner of Figure 4 because the task 'Customize[language]' makes the softgoal 'Usability' in Figure 3. This principle helped to complete 19.4% (14 out of 72) of the grid data. A domain expert was then asked to complete the rest of the grid based on the five-point rating scale defined above. The participant was told that she was allowed to change any value in the grid as she wanted, but no attempt was made by the domain expert to change the already-existing 14 ratings in the grid.

Candidate aspectual requirements could be graphically identified by visualizing the reordered grid in Figure 4. For instance, the three consecutive ratings of "5" appearing in

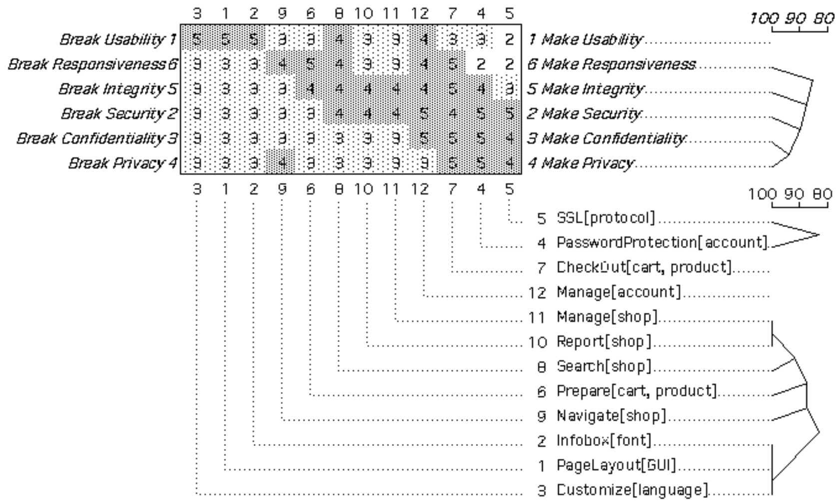

Figure 4: Cluster analysis of the goal model grid

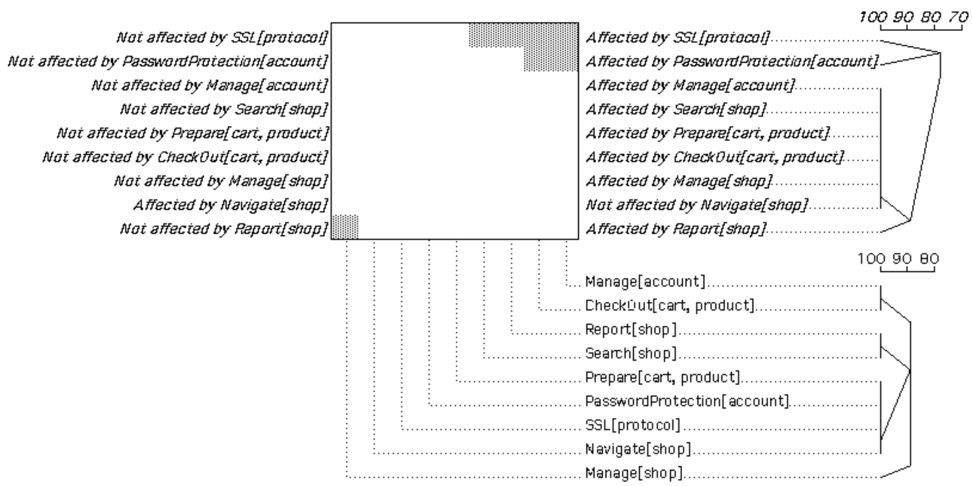

Figure 5: Cluster analysis of dichotomized repertory grid

the top-left corner of the grid isolated tasks 1, 2, and 3 from others. This separation, together with the fact that these requirements joined a cluster by the 100% level, indicated that 'Usability' was the only crosscutting concern among them. Since all these tasks contributed positively to the identified concern, they could be properly considered as advising tasks [20] for the candidate aspect 'Usability'. Therefore, based on this analysis, we obtained early aspect 'Usability' and three advising tasks 'PageLayout[GUI]', 'Infobox[font]', and 'Customize[language]'. Note that no pointcuts or join points relating to this aspectual requirement were identified.

In a similar vein, the unshaded area at the right end of the second row in Figure 4 implied another crosscutting concern – both tasks 4 and 5 negatively affected 'Responsiveness'. But clearly, these two requirements were tangled due to other concerns as well, such as making 'Security' and 'Confidentiality'. To sort out complicated situations like this, a dichotomized repertory grid was constructed, where elements were classified into either pole of the construct [8]. Since aspectual concern has a broadly scoped impact on other requirements, we design constructs of the dichotomized grid to be "whether or not being affected by certain requirement". This treatment also addresses the ongoing RGT debate about whether elements exist independently of constructs, or whether in fact elements are also constructs [8]. From our experience applying RGT to discover early aspects, we feel that it is crucial to explore relationships among elements themselves.

The cluster analysis result of the dichotomized grid is shown in Figure 5, in which shaded blocks represent impact relations. Note that since we have already identified tasks 1, 2, and 3 as advising tasks without having pointcuts and join points, we exclude them from the current analysis. The large blank portion of Figure 5 manifests the characteristics of most requirements under investigation being orthogonal to each other. The indubitably overt sign of the top-right dark region in the grid uncovers another set of aspects and pointcuts in the media shop example [20], and the relevant concepts are highlighted in Figure 6. Note that the bottom-left shadowed slot in Figure 5 can be regarded as a concern that has already been localized: 'Report[shop]' is a sub-task of 'Manage[shop]', therefore the parent task depends on the realization of its child.

Discussion:

According to RGT, four of the six softgoals were identi-

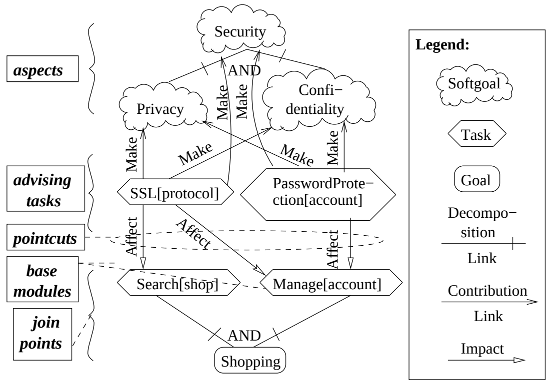

Figure 6: Illustration of early aspects in goal models

fied as candidate early aspects, among which 'Usability' and its three advising tasks were easily captured in Figure 4, and relevant concepts of the remaining three aspects were shown in Figure 6. The reason why softgoals 'Responsiveness' and 'Integrity' were not determined as candidate early aspects by grid analysis was simple: no advising tasks ever existed to contribute to their satisfactions. On one hand, RGT acted as a "filter" on softgoals for early aspects identification. Even though system qualities tended to spread their impacts on several (base) requirements modules (e.g., tasks 6, 7, 8, and 9 all depended on 'Responsiveness' in Figure 3), they should not be viewed as early aspects unless concrete operationalizations [4] were found in goal models. On the other hand, if domain knowledge or other sources strongly suggested that some quality attributes should be considered as candidate early aspects (e.g., 'Integrity') but RGT failed to identify them, more elicitation and analysis were needed to crystallize the outstanding issues. Remember, softgoals are *not* equivalent to early aspects. Otherwise, why bother inventing a new name for an old concept?

Based on RGT, the advising task 'SSL[protocol]' had a broader impact on requirements-level artifacts than 'PasswordProtection[account]' did. This was shown in both Figure 5 and Figure 6, and was reflective of domain expert's perception that tasks 'Report[shop]' and 'Search[shop]' could be done without concerning 'PasswordProtection'. This subtlety could hardly be recognized from Figure 3, but was obvious in the dichotomized grid shown in Figure 5. This distinction between the two closely related advising tasks had effects on weaving aspects and prioritizing requirements, and was one of the key findings in this example. Therefore, our RGT-based approach can discover more interesting properties about the identified early aspects. Otherwise, why bother filling up a 72-slot grid followed by a 9×9 one?

The above discussion points demonstrate the gain by using RGT in goal models: early aspects can be identified accurately (through filtering) and their properties can be revealed thoroughly (by analyzing impact relations).

It is worth pointing out that Figure 6 is already a weaved graph. Early aspects ('Security', 'Privacy', and 'Confidentiality') are weaved into the base modules ('Search[shop]' and 'Manage[account]') through consolidating advising tasks ('SSL[protocol]' and 'PasswordProtection[account]'). The dependency links from base modules to security-related aspects disappear in Figure 6 because tangling and scattering concerns have been localized.

Finally, in order to complete the grid data, human intervention is inevitable. Although qualitative analysis methods, such as label propagation algorithms [5], can offer great benefit for automatically generating (partial) grid contents so that manual workload can be reduced, stakeholder involvement is crucial to the success of our RGT-based early aspects discovery approach, and the participant must be able to change any value in the grid as she wants.

3.2 Viewpoints-Based Models

In viewpoints-based modeling, participants are able to maintain their own (partial) models of the system and its requirements, without being constrained by other participants' models. By keeping the viewpoints of different stakeholders separate, analysts can identify and explore the relationships between them, and participants can understand one another's perspectives better [6].

One major difficulty in identifying early aspects in viewpoints-based requirements models is that stakeholders may have used different vocabularies, or applied a shared vocabulary inconsistently. Table 1 summarizes all four possible relations derived from the overlaps between terminology and concepts. This classification is adopted from Shaw and Gaines' work [17], which compared conceptual structures during the knowledge acquisition process.

If we are able to accurately establish correspondences and effectively detect conflicts between the terms and the underlying concepts used in different viewpoints, then we can compare, contrast, and combine various models to gain an integrated view, on which the aspectual requirements identification can be based. In this subsection, we discuss how to exploit RGT to compare terminological and conceptual structures in different viewpoints. Addressing these issues is a fundamental prerequisite for discovering early aspects in viewpoints-based requirements models.

We continue the discussion with the media shop example. Two stakeholder roles, customer and developer, built their requirements goal models separately. Tasks and softgoals were still treated as elements and constructs in grid analysis. Clearly each person had a different opinion and a different value system. Each of these dimensions was expressed in personally meaningful terms, and was significant to the person who used it. In the same or similar context, each stakeholder's personal construct system is to some degree overlapped with others, and this makes it possible for people to exchange their grids' data to share their individual perceptions of the domain. In our study, we only exchanged concrete entities (tasks) of goal models between stakeholders, and kept the use of abstract constructs (softgoals) inside each person's individual conceptual system.

Table 1: Terminology and concepts

Concepts \ *Terminology*	Same	Different
Same	**Consensus** Stakeholders use terminology and concepts in the same way	**Correspondence** Stakeholders use different terminology for the same concepts
Different	**Conflict** Stakeholders use same terminology for different concepts	**Contrast** Stakeholders differ in terminology and concepts

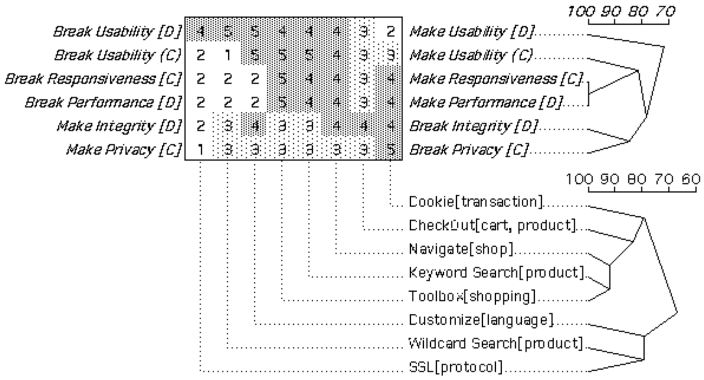

Figure 7: Cluster analysis of viewpoints

In this example, tasks in both customer's and developer's viewpoints were consolidated and shared. This common ground defined the basic prerequisite for any kind of interference analyses [7]. Then, each stakeholder rated all the tasks on his or her own softgoals using the five-point scale defined in the previous subsection. For illustrative purposes, Figure 7 shows the clustering result on 8 tasks and 6 softgoals, where the labeling convention – marking every softgoal's owner after it ("**C**" refers to "customer" and "**D**" represents "developer") – is adopted to distinguish personal constructs from different viewpoints.

It is apparent from Figure 7 that the terms used to express softgoals interfered greatly. For instance, what **C** meant by 'Responsiveness' and the softgoal 'Performance' in **D**'s viewpoint were identical or at least indistinguishable in terms of the tasks shown in the grid. If they were indeed used interchangeably, correspondence would be established. Otherwise, further elicitation and communication should be performed to distinguish these concepts. This analysis is important for deciding whether the same concern, or even the same early aspect, is represented in the same way.

Although both **C** and **D** used the term 'Usability', they probably did not refer to the same concept. These two constructs were associated at the 70% level, one of the second lowest matching scores between any two softgoals shown in Figure 7. From the developer's perspective, the task 'wildcard search' affected 'Usability' positively because people could do a fuzzy search and still retrieve useful information. But the customer thought the task actually broke 'Usability' since using 'wildcard' would involve a steep learning curve for a non-technical user. Exploring such an inconsistency could spark the discussion about what the concern 'Usability' really meant to the media shop organization and whether 'wildcard search' should be implemented.

Establishing correspondences and detecting conflicts at both terminological and conceptual levels are fundamental in requirements engineering when multiple stakeholders and perspectives are involved. Our use of RGT as a means of addressing vocabulary problems is general and appealing, since RGT avoids the problems of imposition of terminology, and the meaning of a term is essentially treated as a relationship between signs and actions. Using RGT to compare and contrast viewpoints-based models not only helps to identify corresponding, conflicting, and crosscutting concerns, but also allows the requirements analyst to generate specific and plausible hypotheses to guide further activities, thereby producing requirements that adequately reflect stakeholders' desires and needs.

Discussion:

Some requirements methods, such as NFR catalogue [5], suggest that requirements analysts equip themselves with a glossary of standard terms to make communication easier and more precise. However, these approaches can cause problems when stakeholders continue to use different interpretations of the terms. They also miss an important opportunity to explore differences in stakeholders' own categories so as to better understand their perspectives. RGT offers a mechanism to compare and contrast objects in the domain of interest, and we applied the technique to analyze early aspects in the problem domain. Since conceptual resolution is common and important in developing and evolving (large-scale) software systems, we believe that RGT can find its proper applications throughout the entire lifecycle.

To summarize, this section presented two examples derived from the media shop study [4] in order to illustrate how RGT can be used to discover early aspects and compare terminological and conceptual structures in requirements models. Based on our experience, we feel that RGT is perhaps best regarded as a particular form of structured interview. Our usual way of eliciting requirements is by conversation. In talking to the stakeholders, we come to understand the way in which they view the problem world and the machine, what goes with what, what implies what, what is important and unimportant, and in what terms they express their interests and concerns. RGT formalizes this process and assigns mathematical values to the relationships between stakeholders' constructs. Collected repertory grids are amenable to many measurements, and the analysis results can not only facilitate aspectual requirements identification and conflicts detection, but also generate specific and plausible hypotheses to guide requirements analyst's work to move "from early aspects to late requirements" [14].

4. RELATED WORK

Baniassad *et al.* [2] presented an integrated approach to manipulating early aspects and exploiting them throughout the software development lifecycle. In addition to intuition and domain knowledge, three general heuristics are provided to facilitate the identification of aspects in requirements: 1). Aspect terms, such as security, reliability, and integrity; 2). Impact requirements, which have influence over other requirements-level concerns or artifacts; and 3). Scattered concerns, which are terms, concepts, or behaviors that appear in multiple (well-organized) requirements. We have taken all these three guidelines into consideration in our RGT-based approach: softgoals are typical aspect terms, and dichotomized grid is used to capture impact requirements and scattered concerns.

Some approaches (e.g. [1], [16]) apply natural language processing techniques to identify early aspects in requirements. Aspect words (e.g. verbs that scattered in multiple requirements) are extracted from (structured) natural language. Our fine-grained method is suitable for handling well-organized requirements models, and the meaning of a term is essentially treated as a relationship between signs and actions in our framework.

Early aspects identification has also been considered in goal-oriented requirements analysis (e.g. [3], [20]). The emerging consensus seems to be that softgoals have huge potential to become early aspects. However, as we pointed out in the example, they are *not* equivalent. Yu *et al.* [20] also used the media shop example in their study, but no attempt was made to show how their process could be extended if multiple viewpoints and conflicting concerns were involved. Our findings are more sensible and enlightening in that subtleties about early aspects can be exposed and plausible hypotheses can be generated. Label propagation algorithms [5] have been used in both [3] and [20] to correlate entities in goal models. We consider that the use of qualitative analysis methods, such as label propagation algorithms, may help our RGT-based method to address some of the anticipated scalability challenges.

5. CONCLUSIONS

Aspects provide the mechanism that enables the source code to be structured to facilitate the representation of multiple perceptions and to alleviate tangling and scattering concerns. Many of these concerns often arise in the problem domain [14], and, therefore, it is important to identify and represent concerns that arise during the early phases of software development, and to determine how these concerns interact. In this paper, we have presented a novel use of Repertory Grid Technique with roots in psychology of personal constructs as a systematic and effective way to discover early aspects in requirements models. We have exploited the approach to identify and analyze early aspects in a proof-of-concept example adapted from the literature; in particular, we showed how aspect-oriented concepts can be uncovered in requirements goal models, and how interference can be detected in viewpoints-based models.

From our initial experience with the proposed approach, we feel that RGT has a rich value in helping analysts to externalize stakeholders' views of the problem world and the machine, explicate interrelationships among the entities appeared in requirements models, detect inconsistencies at both terminological and conceptual levels, and reveal aspects, advices, pointcuts, join points in the early stages of software development life cycle. In-depth empirical studies are needed to lend strength to the preliminary findings reported here. Our future work also includes developing RGT-based conflicts resolution and requirements prioritization methods to explore aspects weaving. Finally, originally developed in the context of psychotherapy, RGT has been successfully applied to analyze how patients' constructions of their social world change over time or from case to case [8]. This makes RGT a very appealing method for coping with requirements and aspects evolution.

All in all, the repertory grid is truly a technique: A grid of itself is nothing more than a matrix of blank cells. It is only limited by the user's imagination. We hope that our work will shed some light on its applications to new situations.

6. ACKNOWLEDGMENTS

We thank Yijun Yu and Jia Wang for help with the media shop study and insightful comments. We also thank the anonymous reviewers for their constructive suggestions. Financial support was provided by NSERC.

7. REFERENCES

[1] E. Baniassad and S. Clarke. Finding aspects in requirements with Theme/Doc. In *Early Aspects Wkshp, Intl. Conf. on AOSD*, 2004.

[2] E. Baniassad, P. C. Clements, J. Araújo, A. Moreira, A. Rashid, and B. Tekinerdoğan. Discovering early aspects. *IEEE Software*, 23(1):61–70, 2006.

[3] I. Brito and A. Moreira. Integrating the NFR framework in a RE model. In *Early Aspects Wkshp, Intl. Conf. on AOSD*, 2004.

[4] J. Castro, M. Kolp, and J. Mylopoulos. Towards requirements-driven information systems engineering: the Tropos project. *Information Systems*, 27(6):365–389, 2002.

[5] L. Chung, B. A. Nixon, E. Yu, and J. Mylopoulos. *Non-Functional Requirements in Software Engineering*. Kluwer Academic Publishers, 2000.

[6] S. Easterbrook. Handling conflicts between domain descriptions with computer-supported negotiation. *Knowledge Acquisition*, 3:255–289, 1991.

[7] A. Finkelstein, G. Spanoudakis, and D. Till. Managing interference. In *Intl. Wkshp on Multiple Perspectives in Software Development*, 1996.

[8] F. Fransella, R. Bell, and D. Bannister. *A Manual for Repertory Grid Technique*. John Wiley & Sons, Ltd., 2nd edition, 2004.

[9] M. Jackson. Problems, subproblems and concerns. In *Early Aspects Wkshp, Intl. Conf. on AOSD*, 2004.

[10] G. A. Kelly. *The Psychology of Personal Constructs*. New York: Norton, 1955.

[11] G. Kiczales, J. Lamping, A. Menhdhekar, C. Maeda, C. Lopes, J.-M. Loingtier, and J. Irwin. Aspect-oriented programming. *LNCS*, 1241:220–242, 1997.

[12] G. Murphy and C. Schwanninger. Aspect-oriented programming. *IEEE Software*, 23(1):20–23, 2006.

[13] B. Nuseibeh. Weaving together requirements and architectures. *IEEE Computer*, 34(3):115–117, 2001.

[14] B. Nuseibeh. Crosscutting requirements. In *Intl. Conf. on AOSD*, pages 3–4, 2004.

[15] A. Rashid, P. Sawyer, A. Moreira, and J. Araújo. Early aspects: A model for aspect-oriented requirements engineering. In *Intl. Conf. on RE*, pages 199–202, 2002.

[16] A. Sampaio, N. Loughran, A. Rashid, and P. Rayson. Mining aspects in requirements. In *Early Aspects Wkshp, Intl. Conf. on AOSD*, 2005.

[17] M. Shaw and B. Gaines. Comparing conceptual structures: Consensus, conflict, correspondence and contrast. *Knowledge Acquisition*, 1(4):341–363, 1989.

[18] M. L. Shaw. *On Becoming A Personal Scientist: Interactive Computer Elicitation of Personal Models of the World*. Academic Press, 1980.

[19] E. Yu. *Modeling Strategic Actor Relationships for Process Reengineering*. PhD Thesis, U. Toronto, 1994.

[20] Y. Yu, J. C. S. do Prado Leite, and J. Mylopoulos. From goals to aspects: Discovering aspects from requirements goal models. In *Intl. RE Conf.*, pages 38–47, 2004.