Analysis of Early Aspects in Requirements Goal Models: A Concept-Driven Approach

Nan Niu and Steve Easterbrook

Department of Computer Science, University of Toronto
Toronto, Ontario, Canada M5S 3G4
{nn,sme}@cs.toronto.edu

Abstract. *Early aspects* are stakeholder concerns that crosscut the problem domain, with the potential for a broad impact on questions of scoping, prioritization, and architectural design. Analyzing early aspects improves early stage decision-making, and helps trace stakeholder interests throughout the software development life cycle. However, analysis of early aspects is hard because stakeholders are often vague about the concepts involved, and may use different vocabularies to express their concerns. In this paper, we present a rigorous approach to conceptual analysis of stakeholder concerns. We make use of the repertory grid technique to identify terminological interference between stakeholders' descriptions of their goals, and formal concept analysis to uncover conflicts and trade-offs between these goals. We demonstrate how this approach can be applied to the goal models commonly used in requirements analysis, resulting in the clarification and elaboration of early aspects. Preliminary qualitative evaluation indicates that the approach can be readily adopted in existing requirements analysis processes, and can yield significant insights into crosscutting concerns in the problem domain.

Keywords: early aspects, goal-oriented requirements analysis, repertory grid technique, formal concept analysis.

1 Introduction

It has long been recognized that modular systems are easier to produce, maintain, and evolve [23, 32]. However, complex problems are hard to decompose cleanly, and any choice of decomposition will inevitably give rise to concerns that crosscut the resulting structure. Aspect-oriented software development (AOSD) provides explicit means to model concerns that crosscut multiple system components. Initially, much of AOSD research was focused on the solution domain: developers identify and capture aspects mainly in the source code. Recently, a considerable amount of work on *early aspects* [8] has been carried out to identify and model crosscutting properties in the early phases of software development, including the requirements engineering (RE), domain analysis, and architecture design activities.

A. Rashid and M. Aksit (Eds.): Transactions on AOSD III, LNCS 4620, pp. 40–72, 2007.

Early aspects focus on the problem domain, representing the goals and constraints of users, customers, and other constituencies affected by a software-intensive system. Current requirements techniques offer a variety of structures for organizing the requirements, such as (structured) natural languages, use cases, viewpoints, goal models, features, etc. [24, 29]. No matter how the requirements are structured, an early aspect crosscuts the dominant decomposition, and has an (implicit or explicit) impact on more than one requirements artifact. Research on early aspects 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 [1]. Analyzing early aspects also enables stakeholder interests to be traced throughout the software development life cycle.

We assume there exists a relatively well-organized set of requirements derived from some dominant decomposition criteria. 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" [30], which suggests an agenda "from early aspects to late requirements" because identifying aspects too early is counterproductive [31]. This paper presents a rigorous approach to systematically capturing and analyzing crosscutting entities in requirements goal models.

1.1 Aspects and Goal-Oriented Requirements Analysis

Goal modeling has become a central activity in RE. It shifts the emphasis in requirements analysis to the actors within an organization, their goals, and the interdependencies between those goals, rather than focusing on processes and objects. This helps us understand *why* a new system is needed, and allows us to effectively link software solutions to business needs.

Requirements goal models use *goal* decomposition to support the description and analysis of stakeholder intentions that underlie the required software system. Some goal-oriented frameworks, such as i^* [53], also explicitly model the agents who hold these intentions. Goal modeling frameworks also distinguish between regular goals and *softgoals*. A regular goal has a clear criterion that determines whether it is satisfied. In contrast, softgoals are those goals for which there is no clear sense in which they can be fully satisfied. Softgoals therefore often capture non-functional requirements (NFRs) such as usability, reliability, maintainability [5]. In the design process, each softgoal can be used to induce a preference ordering over different design solutions.

The process of goal decomposition produces a goal hierarchy, which expresses how low-level, concrete goals and tasks contribute to (or detract from) the higher level goals. The most abstract goals, especially softgoals, tend to crosscut the goal model, connected by contribution links to many other lower level goals [1]. We treat such goals as *candidate early aspects*, and analyze the extent to which they capture key problem domain concepts that crosscut the structure of requirements and architectural designs. Candidate aspects are suited to be implemented as code aspects, but developers may choose other means to address these crosscutting concerns. Even in the latter case, it is desirable to keep candidate early aspects modularized so that one does not have to recover them from the code at a later date.

Our analysis focuses on clarifying the problem domain concepts that underlie the candidate early aspects. By investigating the meanings of the terms that stakeholders use to describe these high-level goals, we can determine whether they do indeed represent concerns that crosscut requirements and design artifacts. Not all high-level goals are crosscutting. For example, a softgoal might be relevant just to a single design decision, or it might impact a large number of design decisions. If the early aspect requires specific implementation steps to be taken, these may be contained in a single part of the program code, or may be code aspects that crosscut the program structure.

A key problem with goal models is the high level goals, which are candidate aspects, are often described by stakeholders using vague terms such as "reliable" and "user friendly". Hence, our approach includes a rigorous analysis of the stakeholder *concepts* relevant to their stated softgoals. Concepts are fundamental building blocks of human learning [28]. We use formal concept analysis (FCA), a mathematical technique that treats concepts as binary relations between objects and their properties [12]. We apply FCA to the contribution links between system tasks and softgoals in a goal model. The resulting concept lattice offers non-trivial insights into relationships between candidate aspects and various concerns appearing in the problem domain.

The development of any sizeable software system invariably involves many stakeholders, whose concerns may overlap, complement, or contradict each other. Models with built-in notions of agents or perspectives add another challenge to requirements' crosscutting structure: stakeholders may express their concerns using overlapping, yet incoherent, vocabularies. For example, what one calls "responsiveness" may correspond to "performance" in another person's description, thus the same concept is *scattered* over multiple terms. As another example, "usability" for software running in a cell phone may be interpreted as "easy to learn" by one stakeholder, and as "mobility" by another stakeholder, causing different concepts *tangled* in one expression. The challenge is to align concepts with respect to stakeholder vocabularies. We argue that an early aspects framework must provide mechanisms to avoid tangling of distinct concepts expressed using the same term, and to prevent scattering of one concept over dissimilar lexicons.

Analysis of conceptual and terminological interference is only possible if we are able to discover relationships between different stakeholders' mental models and the terms they use to describe them. Kelly's personal construct theory (PCT) [18] addresses this issue. It explains how an individual constructs a personal (i.e., idiosyncratic) view of his or her environment (e.g., artifacts, events). The theory has been used to develop techniques for exploring personal constructs, most notably the repertory grid technique (RGT) [10]. We present a novel use of RGT as a means of exploring how stakeholders interpret the labels attached to softgoals, thereby helping to build a vocabulary map to tackle early aspects' intertwinement in different viewpoints.

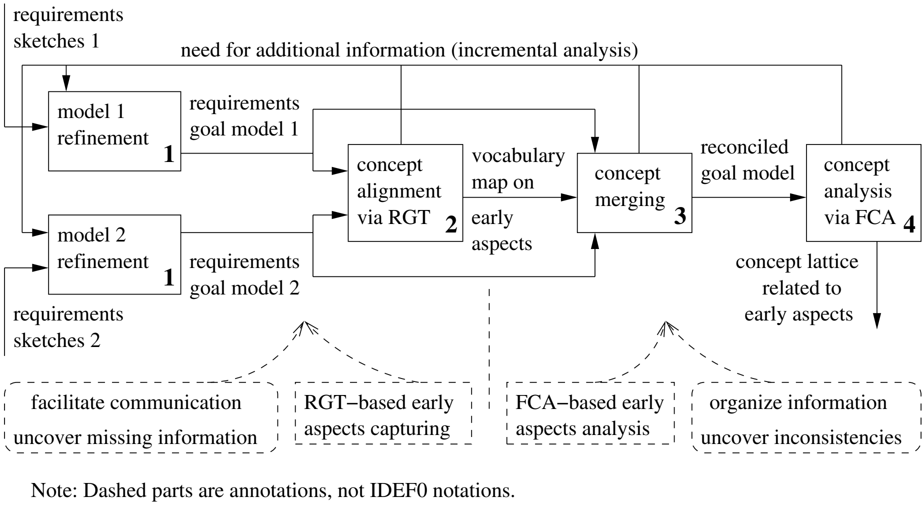

Fig. 1. Process overview of the concept-driven framework in IDEF0 notations

A detailed analysis of candidate early aspects in requirements goal models offers a number of benefits:

– Explicit reasoning about interdependencies between stakeholder goals;
– Improving the modularity of requirements and architectural models;
– Identification of the impact of early aspects on design decisions can improve the quality of the overall design and implementation;
– Test cases can be derived from early aspects to enhance stakeholder satisfaction; and
– Tracing stakeholder interests becomes easier since crosscutting concerns are captured early on.

From the standpoint of aspect-oriented program analysis and evolution, early aspects provide a baseline to justify and validate code aspects against their purpose of existence: are they required by specific stakeholders or refactored based on particular implementations?

1.2 Approach Overview

Our objective is to leverage available and appropriate techniques to form a coherent early aspects analysis framework. Although they originated from separate disciplines, RGT and FCA share a common underlying structure: a cross-reference table. This allows these two techniques to be incorporated seamlessly.

This paper presents a concept-driven framework for capturing and analyzing early aspects in goal models based on RGT and FCA. The process overview of our framework is depicted in Fig. 1 using the integration definition for function modeling (IDEF0) notations [14]. The indexed boxes in Fig. 1 represent the sequence of main actions. Incoming and outgoing arrows model inputs and outputs of each function, respectively.

The motivation to employ a composite (RGT + FCA) approach is to utilize each technique under proper conditions and for proper tasks. Having roots in the psychology of personal constructs makes RGT suitable for aligning and merging stakeholder softgoals (step 2 in Fig. 1). Being a mathematically rigorous technique, FCA provides a basis for conflict detection and trade-off analysis among various concerns addressed by goal models (step 4 in Fig. 1). Jointly, these two techniques offer a coherent early aspects framework, while facilitating communication, organizing information, and uncovering missing information and inconsistencies (dashed boxes in Fig. 1) [13].

Preliminary work on our use of repertory grid was published in [25, 26, 27]. The emphasis of [25] was to present RGT as a general method to support aspectual requirements identification, while the applicability of leveraging RGT to reconcile softgoals with respect to stakeholder vocabularies was further investigated in [26, 27]. This paper integrates the new FCA-based early aspects analysis to provide a more complete treatment of our framework. Throughout the paper, we demonstrate our approach using a media shop e-business example adapted from the literature [4, 54], where stakeholder goals and intentions are modeled using i^* notations [53].

This paper has mainly focused on analyzing stakeholder softgoals in the problem domain. However, we believe that our proposed concept-driven approach could also facilitate the analysis of aspects in the solution domain or pertinent to functional requirements, such as buffering and caching. Section 2 describes goal modeling of the media shop problem domain. A concept-driven approach is then introduced: Sect. 3 focuses on the RGT-based concept alignment method and Sect. 4 discusses the FCA-based early aspects analysis method. In Sect. 5, we report on the results of an initial qualitative study of the utility of our approach. Section 6 reviews related work. Section 7 draws some concluding remarks and outlines future work.

2 Goal Models for Media Shop

Media shop is a store selling different kinds of media items such as books, newspapers, magazines, audio CDs, and videotapes [4]. We assume that a goal-oriented analysis has been performed to investigate requirements for new on-line services, using i^* as the requirements modeling framework [54]. Early requirements analysis focuses on the intentions of stakeholders. In i^*, stakeholders are represented as (social) actors who depend on each other for goals to be achieved, tasks to be performed, and resources to be furnished [53].

The i^* framework emphasizes the analysis of strategic relationships among organizational actors. A strategic rationale (SR) model exposes the reasoning within each actor by identifying goals, tasks, softgoals, and their relationships. The SR models for two media shop stakeholder roles, customer and developer, are

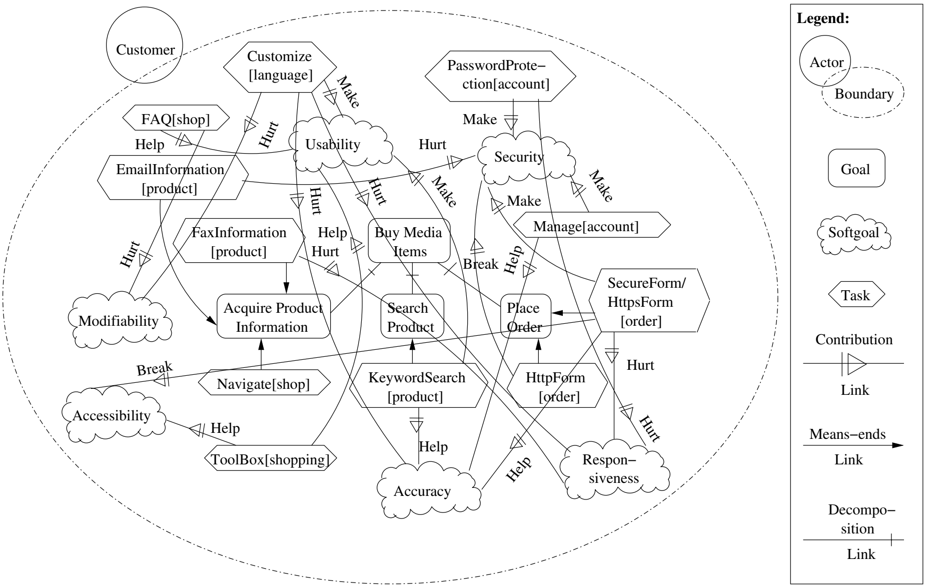

Fig. 2. Strategic rationale (SR) model for media shop customer

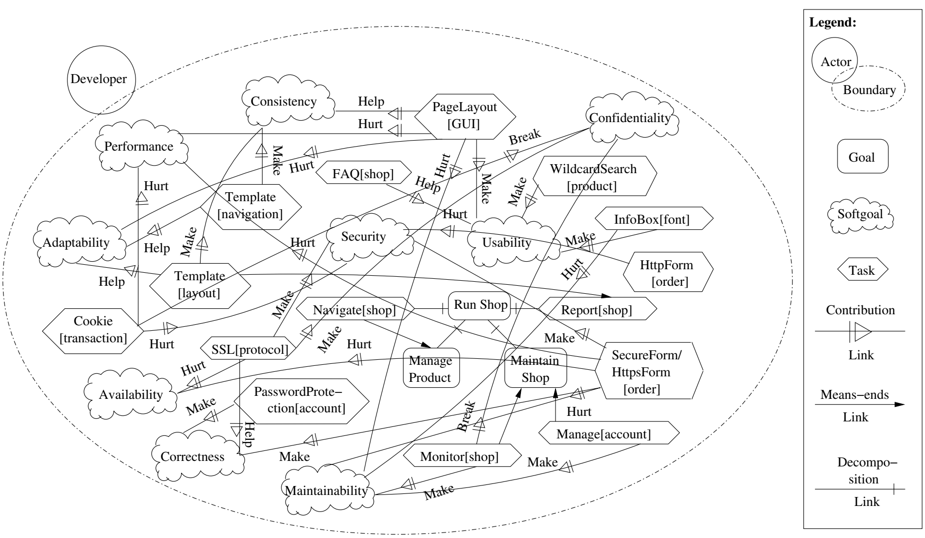

Fig. 3. Strategic rationale (SR) model for media shop developer

shown in Figs. 2 and 3, respectively. Goals are states of desires, such as customer wants to “buy media items” (Fig. 2). Goals can be hierarchically decomposed to subgoals, and can be achieved by various means-ends relations linking to tasks.

Tasks are actions with goals and commitments, such as customer receives "emails about product information" (Fig. 2). NFRs are expressed as softgoals, which can only be *satisficed* within acceptable limits, rather than satisfied absolutely [5]. The satisficing degree is labeled with "make", "help", "hurt", and "break" along the contribution links pointing to specific softgoals in Figs. 2 and 3.

Softgoals are candidate aspects because aspects are usually "units of system decomposition that are not functional" [19]. However, softgoals are often viewed as abstract concepts because they are difficult to express in a measurable way. Zave and Jackson [55] argued that goals by themselves do not make a good starting point for RE, as almost every goal is a subgoal with some higher purpose. Concrete operationalizations must be sought to ensure the satisfaction of goals.

Each goal model defines a particular context for requirements comprehension. Within this context, we identify concepts pertinent to softgoals. A *concept* [1] stands for the knowledge about objects having certain attributes or properties. In a given context, a concept has an extent and an intent. The *extent* is the set of objects in the context that are constituents of the concept, and the *intent* is the set of attributes that capture the essence of the concept. For example, the concept "country" has the extent of Canada and Portugal, but does not subsume Toronto or Europe. The intent may include "a territorial division", "a federation of communities", and the like.

In our approach, we treat each softgoal as a concept. The intent is the concern that a particular softgoal addresses, and the crosscutting nature or broad impact of that concern in the goal model. The extent is the set of model entities that are affected by the candidate aspect. We use tasks that contribute to the softgoal as a baseline to crystallize the concept. The reason is twofold. First, tasks are more concrete than softgoals. While people may have different criteria for determining whether the softgoal, for example, "usability" of the media shop is fulfilled, they generally tend to have a common interpretation of the task "keyword search of product" (Fig. 2). Second, to become true early aspects, not just candidate ones, softgoals need to have specific operationalizations, or advising tasks [54], to contribute to their satisfactions [25]. Concrete entities like tasks are what make softgoals understandable, operationalizable, localizable, and measurable.

In the following two sections, we introduce the concept-driven approach by discussing how contribution links between tasks and softgoals in a goal model can be used to align stakeholder softgoals and analyze candidate aspects' crosscutting properties. We use the media shop example to illustrate the proposed approach.

3 Early Aspects Alignment

When people observe a complex problem domain, their observations are inevitably incomplete. Personal values and experiences act as a filter, leading people to focus on aspects that are particularly salient to them personally. This gives rise to many partial conceptual structures. When asked to articulate these, people choose terms that are meaningful to them. Often, they find it necessary

[1] Formal definition of concept is given in Sect. 4.

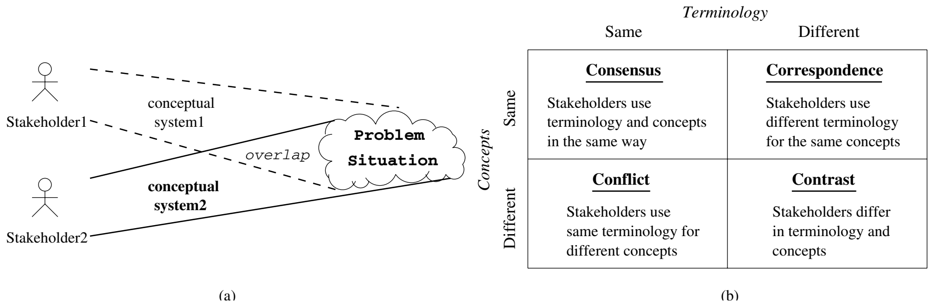

Fig. 4. Conceptual and terminological interference. **a** Overlap in two conceptual systems. **b** Relations between terminology and concepts.

to adapt or invent new terms to describe situations that they have not previously needed to articulate.

In a well-established scientific domain, experts develop a consensus over relevant distinctions and terms. Over time, they identify and test *objective knowledge* independent of individuals [34]. However, such objective knowledge is not yet available for most RE problem domains. If there is no pre-existing consensus over terminology, it is important to be able to compare the conceptual structures among multiple experts [11].

Figure 4a illustrates the situation. Two stakeholders, or more accurately, people playing particular stakeholder *roles*, have developed overlapping conceptual systems that they use to make sense of a problem situation. When these two stakeholders attach terms to their concepts, there are four possible conditions for the relationship between their concepts and terms, as summarized in Fig. 4b [43].

The challenge in knowledge elicitation is to discover which of the situations in Fig. 4b apply for a given set of stakeholder terms:

– *Consensus* is a desirable situation, since stakeholders then have a basis for communication using shared concepts and terminologies.
– *Conflict* (also known as *terminological inconsistency* [16]) can cause significant communication problems throughout the requirements process.
– Discovering *correspondence* is important because it lays the grounds for mutual understanding of differing terms through the availability of common concepts.
– Strictly speaking, *contrast* does not involve terminological interference, but the lack of shared concepts could make communication and understanding among stakeholders very difficult.

We interpret both conflict and correspondence as instances of *terminological interference*. Both have the potential to cause communication problems, if they are not identified and managed. We believe that terminological interference is both inevitable and useful in RE. It is inevitable because stakeholders have complementary perspectives, and are unlikely to have evolved a well-defined terminology for describing the problem situation. It is useful because it provides an opportunity to probe differences in the stakeholders' conceptual systems, to challenge ill-defined terms, and to identify new and productive distinctions for important concepts in the problem domain. However, this can only be achieved if terminological interference is identified and managed. Explicit consideration of terminological interference also helps to keep stakeholders from reaching a false, and often too early, consensus [43].

3.1 Personal Construct Theory and Repertory Grid Technique

George Kelly's PCT [18] presented a view of "man-the-scientist". From within this view, people were assumed to construct a model of the world (much as a scientist constructs a theory), act on the basis of their model (as the scientist creates an experiment to test the theory), and then alter their model in the light of feedback from the results of their actions (as the scientist uses data from the experiment to modify the theory). It is interesting to note that this view shares much of the spirit of the Inquiry Cycle presented in [35], in which requirements models are theories about the world, and designs are tests of those theories.

From the PCT perspective, 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" [18]. 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 [42].

Kelly originally developed PCT in the context of psychotherapy and developed an associated methodology, the 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 [10]. 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 [10].

Figure 5a shows a sample repertory grid for the media shop, in which rows represent constructs and columns represent elements. For a greater degree of differentiation, a five-point scale is used to indicate where an element lies with

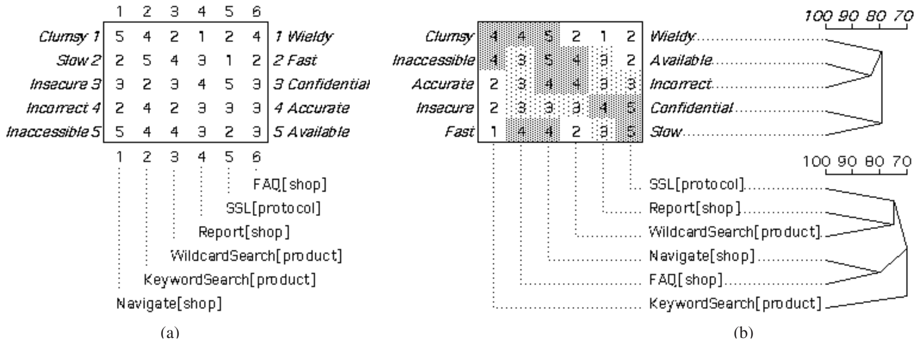

Fig. 5. A sample repertory grid for the media shop. **a** Display of repertory grid. **b** Cluster analysis of repertory grid.

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: 1. **Neutral:** the element is at neither one pole nor the other of the construct. 2. **Not applicable or unknown:** the element is outside the range of convenience of the construct. For example, the element "Report[shop]" is not pertinent to the construct "Inaccessible-Available". Therefore, a rating of "3" appears in the fifth row and the fourth column of Fig. 5a.

The most interesting feature of RGT is the wide variety of different types of analyses that can be applied to the gathered personal constructs. In our approach, the FOCUS program [42] is used to perform a two-way hierarchical cluster analysis, and to reorder the grid so that similarly rated elements are adjacent and similarly used constructs are adjacent.

Figure 5b shows the reordered media shop sample grid. The upper and lower dendrograms illustrate the strength of association between constructs and between elements respectively. To highlight the clusters within the grid, ratings of four and five are given dark shading, ratings of three are given light shading, and ratings of one and two are left unshaded. This enables the easy identification of "blocks" within the grid [42]. For example, the top left cluster in Fig. 5b shows that keyword search, FAQ, and navigation are similar with respect to the "Clumsy–Wieldy" criterion in this particular context. Next, we show how to leverage RGT to align candidate aspects regarding to different stakeholder vocabularies.

3.2 Early Aspects Alignment Via Repertory Grid Technique

A competent early aspects framework shall provide mechanisms to avoid tangling of distinct concepts expressed in the same term, and to prevent scattering of one concept over dissimilar lexicons. We present a novel use of RGT with roots in PCT as a means of addressing terminological problems in different viewpoints

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

Fig. 6. Qualitative scale for measuring softgoal contributions

("correspondence" and "conflict" areas shown in Fig. 4b). We associate softgoals with stakeholders' personal constructs, and use the tasks that contribute to these goals as elements that stakeholders can rate using their constructs. The key idea is to compare the stakeholders' constructs by how they relate to a shared set of concrete entities, rather than by any terms the stakeholders use to describe them. In this way, we avoid making any assumptions about the meanings of individuals' constructs. Four highly iterative and interactive activities are involved in our framework: extraction, exchange, comparison, and assessment. We now describe each step in more detail using the media shop example described in Sect. 2.

***Extraction*:** Given a set of goal models, we need to extract relevant information within some context to identify constructs and elements for grid analysis. A key underlying assumption of PCT and RGT is that elements define the context. Elements need to be carefully chosen to be within the range of convenience of the constructs we wish to study [10]. For instance, it bends our minds to consider "antique" or "modern" numbers and "prime" or "non-prime" furniture.

When analyzing goal models, we begin with some core agent or key activities in the system, and this generally provides a well-scoped area of interest. We carefully record the context of each grid so that sensible and relevant exchange and comparison can be performed.

Softgoals are candidate aspects and are often difficult to express in a measurable way, so it is hard to ensure that different stakeholders understand them in the same way. Softgoals within the context are selected as personal constructs, and each construct is identified as a pair of polar extremes corresponding to "make the goal" and "break the goal". Concrete entities of the same type (e.g., tasks), which are related to the chosen constructs, are selected as elements. The reason is twofold. First, empirical evidence suggests that people are better at comprehending and making analogies between concrete concepts rather than abstractions in RE [33]. Second, heterogeneous elements are likely to result in range of convenience problems as well as decreasing the validity of the grid [10].

Each element is then rated on each bipolar construct. For each grid, some ratings can be obtained from the goal models directly, some can be derived through label propagation algorithms [5], and the remainder need to be completed by the stakeholder. A five-point scale is defined in Fig. 6 to make such measures both subtle and specific. This multi-rating scale captures softgoals' satisficeability: softgoal fulfillment is relative and "good enough" rather than absolute [5].

The goal models shown in Figs. 2 and 3 share the media shop context, and the extracted tasks and softgoals from these strategic rationale models are listed

Tasks / Elements		Softgoals / Constructs	
Navigate[shop]	Customize[language]		
KeywordSearch[product]	WildcardSearch[product]		Consistency [D]
ToolBox[shopping]	Report[shop]	Usability [C,D]	Availability [D]
FAQ[shop]	PageLayout[GUI]	Responsiveness [C]	Maintainability [D]
PasswordProtection[account]	InfoBox[font]	Security [C,D]	Adaptability [D]
Manage[account]	SSL[protocol]	Modifiability [C]	Correctness [D]
EmailInformation[product]	Cookie[transaction]	Accessibility [C]	Performance [D]
FaxInformation[product]	Template[layout]	Accuracy [C]	Confidentiality [D]
HttpForm[order]	Template[navigation]		
SecureForm/HttpsForm[oder]	Monitor[shop]		

Fig. 7. List of elements and constructs for media shop

in Fig. 7. In our approach, the set of tasks determines the common ground and is shared among all analysts. In addition to its name, each task also contains the subject matter [55] shown in squared parentheses. Softgoals are treated as personal constructs, so we specify the owner(s) after each softgoal in Fig. 7. Where softgoals have the same name but different owners, we treat them as distinct constructs. The labeling convention — explicitly marking every softgoal's owner after it ("C" refers to "customer" and "D" represents "developer") — is adopted in this paper. This also addresses the traceability concern from the model management perspective [40].

The construction of a repertory grid is perhaps best regarded as a particular form of structured interview with each stakeholder. The answers to such questions as, "how does this task affect the system's maintainability?", may give us an understanding of the interweaving of the stakeholder's terminology and provide us with an understanding of her outlook that no dictionary could offer.

***Exchange*:** Each grid expresses the way in which a particular stakeholder views the domain and in what terms she seeks to make sense of the underlying elements. Each of these dimensions is expressed in personally meaningful terms, and is significant to the person who used it.

In a shared context, each stakeholder's personal construct system overlaps to some degree with others, and this makes it possible for people to exchange their grids data to share their individual perceptions of the domain. Such exchange needs to be administered in a structured manner in order to reduce stakeholders' cognitive burdens, and at the same time, to achieve sensible results that are amenable to interference analysis.

We only exchange the extracted common set of tasks between stakeholders, and keep the use of softgoals inside each person's individual conceptual system. Structural exchange allows the tasks in the goal model derived from one stakeholder to be assessed by another in order to determine whether the two stakeholders have consensus or conflict in their use of terminology and concepts.

In our approach, only concrete entities, i.e., tasks, are exchanged, because at this stage, the abstract constructs only have meaning within each person's individual conceptual system. A construct is a discriminator, not a verbal label [10].

	Navigate[shop]	KeywordSearch[product]	ToolBox[shopping]	Customize[language]	FAQ[shop]	PasswordProtection[account]	Manage[account]	EmailInformation[product]	FaxInformation[product]	HttpForm[order]	SecureForm/HttpsForm[order]	WildcardSearch[product]	Report[shop]	PageLayout[GUI]	InfoBox[font]	SSL[protocol]	Cookie[transaction]	Template[layout]	Template[navigation]	Monitor[shop]	
Break Usability {C}	3	5	5	5	5	3	3	2	4	3	3	1	3	4	4	2	3	3	3	3	*Make Usability {C}*
Break Responsiveness {C}	3	4	4	2	3	2	3	3	2	4	2	2	3	2	3	2	3	2	4	3	*Make Responsiveness {C}*
Break Security {C}	3	3	3	3	3	5	5	2	3	2	4	3	2	3	3	4	2	3	3	2	*Make Security {C}*
Break Accuracy {C}	3	4	3	2	3	4	5	3	3	2	4	2	3	3	3	4	3	3	3	3	*Make Accuracy {C}*
Break Modifiability {C}	3	4	3	2	2	2	4	3	3	3	2	2	3	4	2	2	3	4	4	4	*Make Modifiability {C}*
Break Accessibility {C}	3	3	4	2	3	3	2	3	3	2	1	3	3	2	2	2	2	3	3	3	*Make Accessibility {C}*
Break Usability {D}	3	4	5	2	4	3	3	4	1	3	3	5	3	5	5	2	3	3	3	3	*Make Usability {D}*
Break Security {D}	3	3	3	2	3	4	2	3	4	2	5	3	3	3	3	5	2	3	3	4	*Make Security {D}*
Break Consistency {D}	4	3	3	2	3	3	3	2	4	3	3	2	3	3	2	3	3	5	5	4	*Make Consistency {D}*
Break Availability {D}	3	3	4	2	3	3	3	3	3	3	2	3	2	2	2	2	2	3	3	2	*Make Availability {D}*
Break Maintainability {D}	3	3	2	1	2	3	5	3	3	3	2	3	3	4	2	2	2	4	4	5	*Make Maintainability {D}*
Break Adaptability {D}	4	3	2	4	3	2	3	4	4	4	3	4	3	2	3	2	2	4	4	3	*Make Adaptability {D}*
Break Correctness {D}	3	4	3	3	3	5	4	3	3	2	5	3	3	3	3	4	2	3	3	3	*Make Correctness {D}*
Break Performance {D}	3	4	4	2	4	2	3	3	3	3	2	2	3	2	2	2	2	3	4	3	*Make Performance {D}*
Break Confidentiality {D}	3	3	3	3	3	5	5	2	3	2	4	3	2	3	3	5	1	3	3	1	*Make Confidentiality {D}*

Fig. 8. Repertory grid for media shop

Construct names are merely convenient labels to remind the person of the thoughts and feelings that the construct provoked, and hence are not transferable to another person without discussion and negotiation [42].

On the other hand, the concrete entities *are* exchanged, because to make comparisons across individuals and investigate construct similarity requires that they each construe the same set of elements [22]. We assume that people focusing on similar topics would agree on the definition of a common set of concrete tasks within the area of interest, i.e., when presented with specific and relevant tasks that are devised by others, people are likely to grasp the essential meaning behind the notions.

In the media shop example, the tasks from different viewpoints are consolidated and shared. Then, each stakeholder rates all twenty tasks (elements) on his or her own softgoals (constructs) using the 5-point scale defined in Fig. 6. Figure 8 presents an integrated view of individual repertory grid filled up by the customer and the developer.

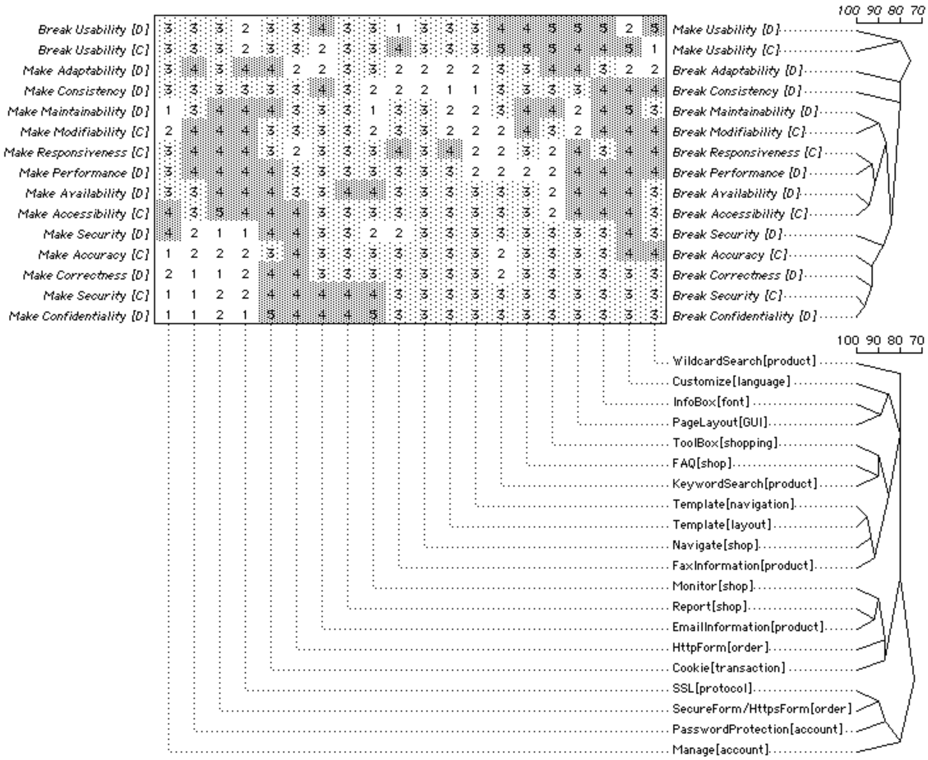

Fig. 9. Cluster analysis for the repertory grid in Fig. 8

***Comparison*:** After focused extraction of repertory grids from goal models and structured exchange of tasks between stakeholders, stakeholders' softgoals can now be compared. The set of all tasks offers a common ground to evaluate softgoals in a particular context. We compare the softgoals according to the extent to which they array the set of tasks.

The relationship between any two constructs can be examined by seeing to what extent the ratings of all the elements on one of the constructs tend to match, or differ from, the ratings on the other construct [39]. If two softgoals orchestrate the tasks in the same or very similar way, "correspondence" (Fig. 4b) is established between these constructs even though they may be labeled differently. If two softgoals that have been labeled using the same terms align the tasks in a markedly dissimilar fashion, then "conflict" (Fig. 4b) is detected.

Figure 9 presents the cluster analysis result for the repertory grid shown in Fig. 8. It is apparent from Fig. 9 that the terms used to express softgoals interfered greatly. For example, what the customer meant by "Security" and the concern "Confidentiality" in the developer's viewpoint were associated at the 96.2% level. In terms of the tasks shown in the grid, these two softgoals differ only in the degrees, not in the poles' extremity, rated by three elements: one rated two on “Security [C]” and one on “Confidentiality [D]”, and the other two rated four on “Security [C]” and five on “Confidentiality [D]”. Thus, these two crosscutting concerns are considered to be very similar based on grid analysis. If they are indeed used interchangeably, correspondence would be established. Otherwise, further elicitation should be performed to distinguish these constructs.

Although the customer and the developer both used the term “Usability”, they probably did not refer to the same concept. These two constructs were associated at the 80% level, one of the lowest matching scores between softgoals appeared in Fig. 9. 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.

The grid analysis results, such as the correspondence and conflict relationships identified above, not only enable us to gain insights into stakeholders’ use of terminologies and concepts, but also allow us to generate specific and plausible hypotheses to be tested with subsequent efforts in eliciting and communicating requirements. This indeed characterizes part of incremental analysis depicted in Fig. 1, and leads us to assessing the RGT-based early aspects alignment method.

***Assessment*:** Analogies can be drawn between RGT and structured interview. Stakeholders undergo the interview by completing extracted and exchanged grids. Analysis of resultant grids raises a plethora of new questions for further exploration. Essentially, a repertory grid is the start of a dialogue between analyst and stakeholders. For example, if we have identified a potential correspondence between two softgoals, we might ask the stakeholders to suggest further examples of concrete instances, to see if they confirm or refute the correspondence. If stakeholders suspect their softgoals do not correspond, they may be inspired to find tasks that disambiguate them. If they suspect two conflicting constructs really do mean the same thing, they might discuss the apparent discrepancy in their ratings of the concrete elements against this construct.

Assessment can thus be considered as a follow-up interview to address the newly generated questions, and for stakeholders to provide evaluation and feedback about the quality and usefulness of the obtained data. In our framework, there is no independent measurement since the collected data is context-laden and is open to interpretation in a number of ways. Our exploratory RGT-based approach is of practical value if our findings can provoke fruitful discussions and guide further RE activities to precisely comprehend stakeholders’ terminologies and conceptual systems, thereby producing requirements that adequately reflect their desires and needs.

Our concept alignment approach is appealing, since PCT and RGT avoids the problems of imposition of terminology, and the meaning of a term is essentially treated as a relationship between signs and actions. One desired outcome of aligning concepts could be a vocabulary map between different viewpoints.

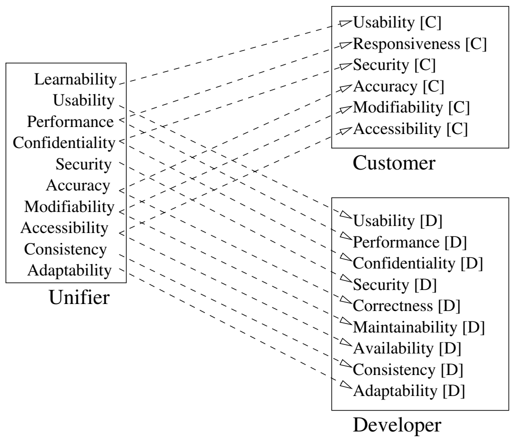

Fig. 10. An attempt to establish a unifier and vocabulary map for media shop

Figure 10 shows our initial attempt to build such a vocabulary map for the media shop example. This mapping is based on grid analysis of Fig. 9, and each attempt to establish a unifier and vocabulary connector can be seen as a hypothesis about how to put different viewpoints together, in which choices have to be made about which concepts overlap, and how the terms used by various stakeholders are related. If a particular set of choices yields an unacceptable result, it is better to perform the concept merge, investigate the resulting inconsistencies, and carry out the incremental analysis (Fig. 1), rather than restrict the available mapping choices. Detailed processes of exploratory vocabulary reconciliation are beyond the scope of this paper, and are considered as future work. We use Fig. 10 as the baseline mapping to facilitate discussion for the rest of the paper.

3.3 Discussion

We have presented a PCT- and RGT-based approach to systematic and effective concept alignment, and have illustrated the method by identifying synonyms and homonyms of candidate aspects in goal models. Our approach is lightweight in that both elements and constructs are extracted from well-organized models from goal-oriented RE [29], rather than being elicited from scratch. Currently, focused grid extraction based on key activities is used as a heuristic to reduce irrelevance between different views. Future work is planned for ensuring the range of convenience of repertory grids offered by different stakeholders.

Softgoals in goal models are a good starting point for analyzing early aspects, and candidate aspects can be seen as operationalizations of softgoals and the relations to functional goals [54]. Our RGT-based concept alignment approach is also applicable to the analysis of crosscutting functional requirements. In [25], dichotomized repertory grids are constructed as an impact analysis tool to reveal crosscutting properties of goal model entities. It is thus advantageous to align candidate aspects in goal models by using the tasks they crosscut, and RGT could provide support for doing so.

To investigate construct similarities and differences across individuals using RGT, we assume that concrete entities, such as tasks, in the given context can be mutually understood among stakeholders. It might not be the case that people can precisely and accurately interpret other people's elements, because the actual phrasing and labeling of elements will have crucial impact on our proposed method. Our overall perception is that people make good approximations when trying to understand concrete and pertinent concepts, so our assumption is "good enough" for applying RGT to deal with early aspects alignment in goal models. Besides, any comparison of conceptual systems necessarily involves approximation since a complete conceptual system may involve indefinitely complex relations and different concepts will never be identical in all respects [43].

Finally, like any process, the quality of the output of our RGT-based method is only as good as the quality of the inputs and the controls over the process. Our inputs rely on various stakeholders in the problem domain, on which early aspects research is based. And we believe that our approach not only has strong yet flexible controls over the extraction, exchange, comparison, and assessment process, but also has a profound and solid foundation, namely the PCT, of aligning concepts in different viewpoints. Our output, such as the clustering result shown in Fig. 9 and the vocabulary map in Fig. 10, can enable the requirements analyst to both gain insights into stakeholders' use of domain concepts and to generate plausible hypotheses to guide further early aspects analysis activities.

4 Early Aspects Analysis

This section describes the application of FCA to supporting trade-off analysis and conflict detection of early aspects in requirements goal models. We also discuss the seamless integration of FCA with RGT so as to form a coherent concept-driven framework. Before diving into the details of FCA, we examine the connection that "concept merging" (step 3 in Fig. 1) makes between candidate aspects alignment and analysis.

The early aspects alignment method introduced in Sect. 3 can assist in highlighting discrepancies over the terminology being used and the concepts being modeled. Concept merging in goal models can thus be considered as grid merging of consolidated tasks and reconciled softgoals. We assume that in the adapted merged view, there may only be one contribution relationship between a task and a softgoal: either being negative or being positive, but cannot be both, [2] i.e., the merged model exhibits an internal consistency with respect to softgoal contributions. For example, although both the customer and the developer expressed "Usability" in their original views (Figs. 2 and 3), a terminological

[2] We simplify the scale for measuring softgoal contributions to be "positive" and "negative". In Fig. 6, "break" and "hurt" are associated with "negative", "help" and "make" correspond to "positive", and "neutral" means there is no contribution relationship between the task and the softgoal.

conflict was detected between these two concepts (Fig. 9) and the customer's softgoal was aligned to the "Learnability" concern by grid analysis (Fig. 10). Thus, in the reconciled view, occurrences of terminological interference are identified and managed, and distinct early aspects are represented using distinct lexicons.

4.1 Formal Concept Analysis

Formal concept analysis (FCA) is a mathematical technique for analyzing binary relations. The mathematical foundation of concept analysis was laid by Birkhoff [3] in 1940. For more detailed information on FCA, we refer to [12], where the mathematical foundation is explored.

FCA deals with a relation $\mathcal{I} \subseteq \mathcal{O} \times \mathcal{A}$ between a set of objects $\mathcal{O}$ and a set of attributes $\mathcal{A}$. The tuple $C = (\mathcal{O}, \mathcal{A}, \mathcal{I})$ is called a *formal context*. For a set of objects $O \subseteq \mathcal{O}$, the set of common attributes $\sigma(O)$ is defined as:

$$\sigma(O) = \{a \in A \mid (o, a) \in \mathcal{I} \text{ for all } o \in O\}. \quad (1)$$

Analogously, the set of common objects $\tau(A)$ for a set of attributes $A \subseteq \mathcal{A}$ is defined as:

$$\tau(A) = \{o \in O \mid (o, a) \in \mathcal{I} \text{ for all } a \in A\}. \quad (2)$$

A formal context can be represented by a relation table, where columns hold the objects and the rows hold the attributes. An object o_i and attribute a_j are in the relation $\mathcal{I}$ if and only if the cell at column i and row j is marked by "×". As an example related to the media shop, a binary relation between a set of objects {CD, MAGAZINE, NEWSPAPER, VIDEOTAPE, BOOK} and a set of attributes {free-distribution, timely, paper, sound} is shown in Fig. 11a. For that formal context, we have:

$$\sigma(\{\text{CD}\}) = \{\text{free-distribution, sound}\},$$
$$\tau(\{\text{timely, paper}\}) = \{\text{MAGAZINE, NEWSPAPER}\}.$$

A tuple $c = (O, A)$ is called a *concept* if and only if $A = \sigma(O)$ and $O = \tau(A)$, i.e., all objects in c share all attributes in c. For a concept $c = (O, A)$, O is called the *extent* of c, denoted by $extent(c)$, and A is called the *intent* of c, denoted by $intent(c)$. Informally speaking, a concept corresponds to a maximal rectangle of filled table cells modulo row and column permutations. In Fig. 11b, all concepts for the relation in Fig. 11a are listed.

The set of all concepts of a given formal context forms a partial order via the superconcept-subconcept ordering $\leq$:

$$(O_1, A_1) \leq (O_2, A_2) \iff O_1 \subseteq O_2, \quad (3)$$

or, dually, with

$$(O_1, A_1) \leq (O_2, A_2) \iff A_1 \supseteq A_2. \quad (4)$$

MEDIA SHOP	free-distribution	timely	paper	sound
CD	X			X
MAGAZINE		X	X	
NEWSPAPER	X	X	X	
VIDEOTAPE				X
BOOK			X	

(a)

$\top$	({CD, MAGAZINE, NEWSPAPER, VIDEOTAPE, BOOK}, Φ)
c_1	({CD, VIDEOTAPE}, {sound})
c_2	({CD, NEWSPAPER}, {free-distribution})
c_3	({MAGAZINE, NEWSPAPER, BOOK}, {paper})
c_4	({MAGAZINE, NEWSPAPER}, {timely, paper})
c_5	({NEWSPAPER}, {free-distribution, timely, paper})
c_6	({CD}, {free-distribution, sound})
$\perp$	(Φ, {free-distribution, timely, paper, sound})

(b)

Fig. 11. An example relation between objects and attributes. **a** Formal context. **b** Concepts for the formal context.

Note that (3) and (4) imply each other by definition. If we have $c_1 \leq c_2$, then c_1 is called a *subconcept* of c_2 and c_2 is a *superconcept* of c_1. For instance, in Fig. 11b, we have $c_5 \leq c_3$.

The set $\mathcal{L}$ of all concepts of a given formal context and the partial order $\leq$ form a complete lattice, called *concept lattice*:

$$\mathcal{L}(C) = \{(O, A) \in 2^{\mathcal{O}} \times 2^{\mathcal{A}} \mid A = \sigma(O) \text{ and } O = \tau(A)\}. \tag{5}$$

The *infimum* ($\sqcap$) of two concepts in this lattice is computed by intersecting their extents as follows:

$$(O_1, A_1) \sqcap (O_2, A_2) = (O_1 \cap O_2, \sigma(O_1 \cap O_2)). \tag{6}$$

The infimum describes a set of common attributes of two sets of objects. Similarly, the *supremum* ($\sqcup$) is determined by intersecting the intents:

$$(O_1, A_1) \sqcup (O_2, A_2) = (\tau(A_1 \cup A_2), A_1 \cup A_2). \tag{7}$$

The supremum yields the set of common objects, which share all attributes in the intersection of two sets of attributes.

The concept lattice for the formal context in Fig. 11a can be depicted as a directed acyclic graph whose nodes represent the concepts and whose edges denote the superconcept-subconcept relation $\leq$ as shown in Fig. 12. Figure 12 is also called a *line diagram*, which consists the names of all objects and all attributes of the given context. The nodes represent the concepts and the information of the context can be read from the diagram by the following simple reading rule:

An object o has an attribute a if and only if there is an upwards leading path from the node named by o to the node named by a,

or, dually, with

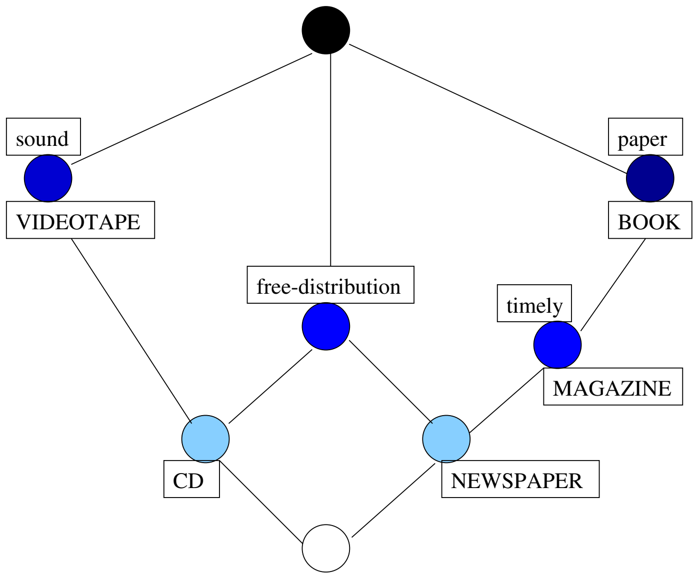

Fig. 12. Concept lattice for the example context in Fig. 11

> An attribute a has an object o if and only if there is an downwards leading path from the node named by a to the node named by o.

Hence, we recognize from the line diagram in Fig. 12 that the node "MAGAZINE" has exactly the attributes "timely" and "paper", and the node "timely" has exactly the objects "MAGAZINE" and "NEWSPAPER". As a consequence of the reading rule, we can easily read from the line diagram the extent and the intent of each concept by collecting all objects below respectively all attributes above the node of the given concept. Hence, the object concept "MAGAZINE" has the extent "MAGAZINE" and "NEWSPAPER" and the intent "timely" and "paper". The extent of the top concept ($\top$) is always the set of all objects, and the intent of the bottom concept ($\bot$) is always the set of all attributes. While in the context of Fig. 11a, the intent of the most general concept ($\top$) does not contain any attribute. In other contexts, it may occur that the intent of $\top$ is not empty. For example, if we add to the given context the attribute "media" with crosses in each row in Fig. 11a, then the top concept would be the attribute concept of "media" and the intent of $\top$ would contain just the attribute "media".

4.2 Early Aspects Analysis Via Formal Concept Analysis

Although they originated from separate disciplines, RGT and FCA share a common underlying structure: a cross-reference table. This allows these two techniques to be incorporated seamlessly. In our approach, the consolidated and reconciled repertory grid provides the cross-reference table from which one can

MEDIA SHOP	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Navigate[shop]																		X		X
KeywordSearch[product]		X		X		X						X								
ToolBox[shopping]		X		X		X										X			X	
Customize[language]		X	X		X				X		X		X		X		X			X
FAQ[shop]		X		X									X							
PasswordProtection[account]					X			X		X		X	X						X	
Manage[account]								X	X			X		X						
EmailInformation[product]	X			X			X			X							X			X
FaxInformation[product]		X	X		X													X		X
HttpForm[order]							X		X		X									X
SecureForm/HttpsForm[order]					X			X		X		X	X		X					
WildcardSearch[product]	X			X	X						X						X			
Report[shop]							X								X					
PageLayout[GUI]		X		X	X									X	X					X
InfoBox[font]		X		X									X		X					
SSL[protocol]	X		X		X			X		X		X	X		X				X	
Cookie[transaction]					X		X		X						X				X	
Template[layout]														X				X		X
Template[navigation]					X								X					X		X
Monitor[shop]							X			X				X				X		

Attributes:
1 −Learnability 2 +Learnability 3 −Usability 4 +Usability 5 −Performance 6 +Performance
7 −Confidentiality 8 +Confidentiality 9 −Security 10 +Security 11 −Accuracy 12 +Accuracy
13 −Modifiability 14 +Modifiability 15 −Accessibility 16 +Accessibility 17 −Consistency
18 +Consistency 19 −Adaptability 20 +Adaptability

Fig. 13. Formal context for media shop derived from the repertory grid in Fig. 8 and the vocabulary map in Fig. 10. Elements in the repertory grid are transformed to objects for the context, and constructs in the grid, which are mapped to negatively (−) and positively (+) contributed softgoals, form attributes for the formal context.

derive the formal context and then identify sub-super relationships among the concepts in the line diagram to facilitate trade-off analysis on early aspects.

To transform a repertory grid to a formal context, we map elements in RGT (the tasks in goal models) to the set of objects $\mathcal{O}$ for the formal context. This can be seen as a direct translation since in our approach, the set of tasks offers a common ground to evaluate early aspects in a particular context and is shared among all stakeholders.

Constructs in RGT are associated with attributes in the formal context. We simplify the finer-grained scale for measuring softgoal contributions to be positive or negative, and assume that if there is a contribution relationship between a task and a softgoal, there can only be one: either negative or positive, but cannot be both. So, for each softgoal a, we have the labels "$-a$" (being negatively affected) and "$+a$" (being positively contributed) appear in the set of attributes $\mathcal{A}$ for the formal context. Note that FCA does have mechanisms for handling multi-valued formal context [12], but the discussion for that support is beyond the scope of this paper.

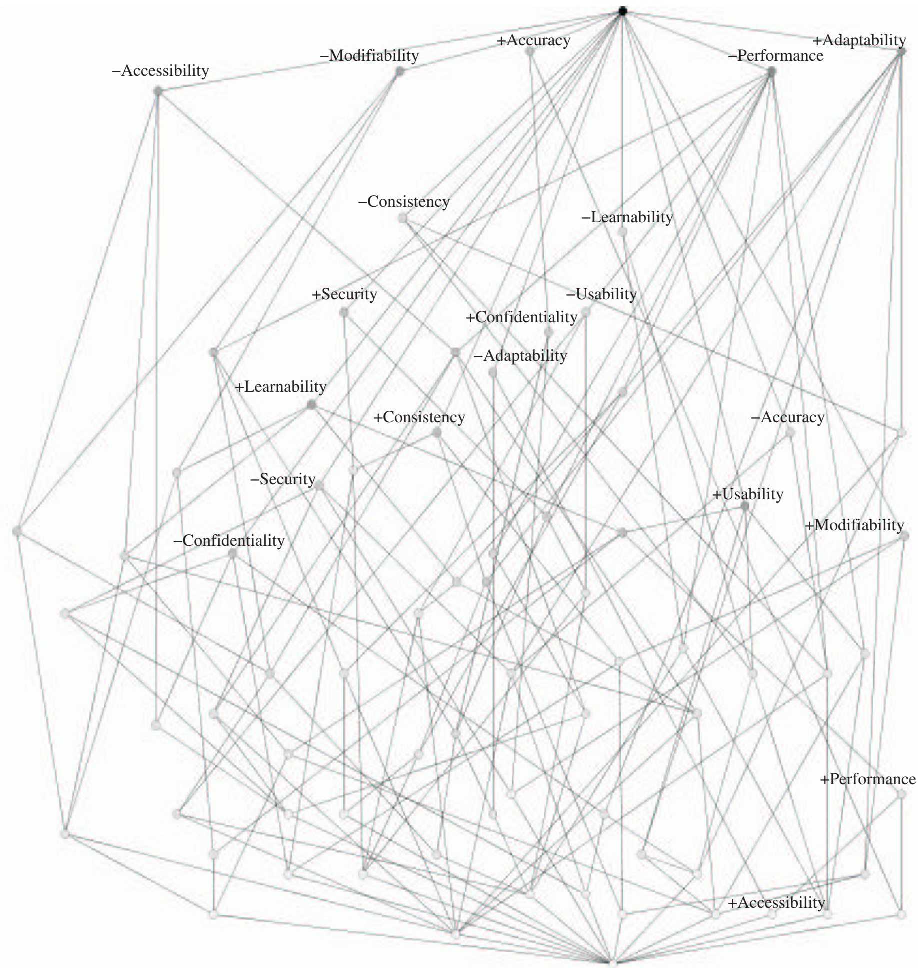

Fig. 14. Concept lattice for the media shop context in Fig. 13. To reduce the clutter of the figure, only the attribute concepts are labeled.

Figure 13 shows the formal context for media shop derived from the repertory grid in Fig. 8 and the vocabulary map in Fig. 10. The concept lattice of this context is shown in Fig. 14, in which only attribute concepts are labeled to reduce the clutter of the line diagram. For each concept c in the media shop example, $extent(c)$ is the set of tasks that have same contribution links to a set of softgoals, and $intent(c)$ is the set of softgoals that crosscut those tasks.

A line diagram provides a rich graphical representation of the information within a particular context, and is amenable to various analyses at the concept level. For example, between the concepts of any context, there is a natural hierarchical order — the subconcept–superconcept relation. This relation can be easily identified in the line diagram by path traversing [12], and can play an important

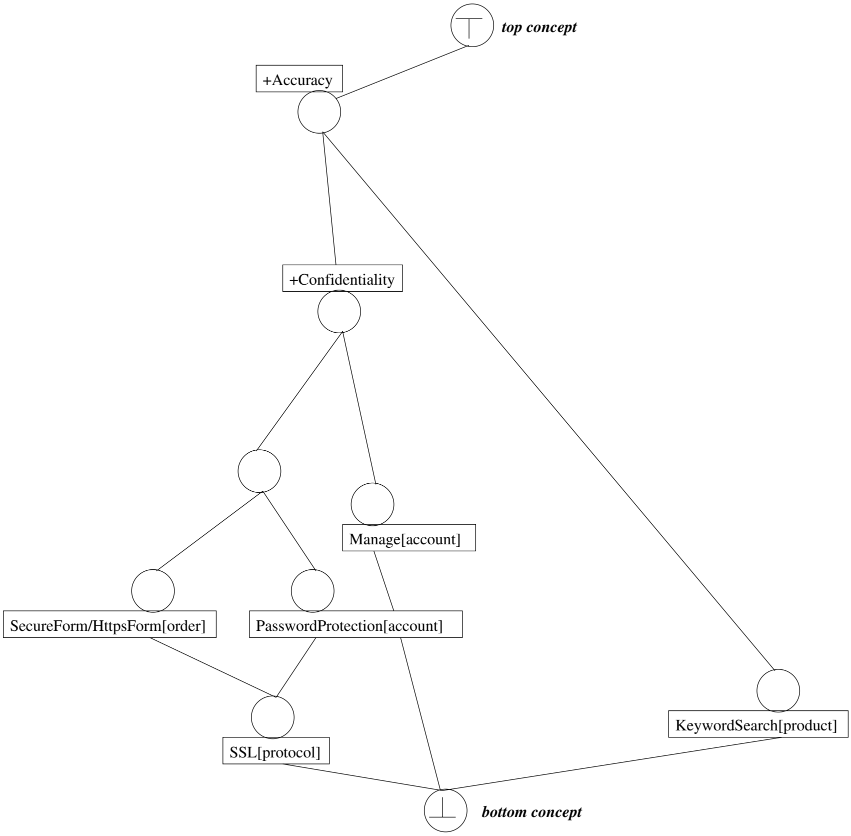

Fig. 15. Sliced concept lattice showing subconcept–superconcept relations

role in analyzing trade-offs, preferences, and priorities on various concepts presented in goal models. A useful technique is to have a slice of interested concepts, i.e., stakeholder concerns, projected from the whole context. Our heuristic is to analyze softgoal fulfillment in a bottom–up way (subconcept to superconcept), and to select concrete operationalizations top–down. This is because fulfillment of subconcept infers fulfillment of superconcept, and selection of superconcept infers selection of one or more subconcepts.

To illustrate subconcept-superconcept relations, a sliced concept lattice for media shop is shown in Fig. 15. By definition [cf. (3) and (4) in Sect. 4.1], the concept labeled with "+Confidentiality" is a subconcept of the one named "+Accuracy". This indicates that in the media shop context, all the tasks that positively contribute to the "Confidentiality" aspect will have positive contributions to "Accuracy". In another word, if the softgoal "Confidentiality" is achieved by the intended software, so is "Accuracy". However, if for some reason, e.g., due to conflicting requirements, "Confidentiality" cannot be satisficed, we still have a choice of implementing the task "KeywordSearch[product]" to address the "Accuracy" concern.

In a similar vein, identifying subconcept–superconcept relations in the concept lattice can help to perform trade-off analysis on concrete tasks. In the media shop case, for instance, the "SSL[protocol]"-labeled concept is a subconcept of the "PasswordProtection[account]"-labeled one, which implies that in addition to all the concerns that "PasswordProtection[account]" addresses, "SSL[protocol]" has (negative or positive) contributions to other softgoals. If these two are competitive tasks for the software system to implement, we may start analyzing the concerns that are uniquely linked to "SSL[protocol]", and neglecting the commonly shared, identically contributed softgoals by both tasks. If our goal is to satisfice as many softgoals as possible, [3] based on the context of Fig. 13, we may choose to include "PasswordProtection[account]" in our architectural design, because "SSL[protocol]" contributes "Learnability" and "Usability" negatively, and these two tasks have exactly the same contribution relationships to the rest of the softgoals in the context. Such analysis helps the analyst to specify preferences and priorities on (aspectual) requirements.

Conflicting candidate aspects in the context can be detected by using the concept lattice. In our setting, one way is to start with softgoals of stakeholder interests that form a subconcept-superconcept relationship. If they have different signs, i.e., one labeled with "−" and the other labeled with "+", then we note them as potential conflict. For example, in Fig. 14, "−Performance" is a superconcept of "+Confidentiality". We mark them as potential conflict since all the tasks that help to fulfill "Confidentiality" will negatively affect media shop's "Performance".

Another way of detecting conflicting concepts is to treat the formal context, such as the one in Fig. 13, as a distance matrix. If the distance between two softgoals is small with respect to some threshold, and they have different signs ("+" versus "−"), then there is a potential conflict between them. This conflict detection method has a similar flavor to the RGT-based hierarchical cluster analysis described in Sect. 3.2, in that both methods examine how different softgoals align a common set of tasks.

In order to perform the trade-off and conflict analyses mentioned above, a query language or some line diagram slicing algorithms shall be provided to facilitate the analyst to select and project interested (clustered) concepts in the concept lattice. However, these features are not yet available for most FCA-based analysis tools [2, 48]. Our early aspects analysis suggests the need for improvement in some areas of FCA tool support.

4.3 Discussion

We have applied FCA for conflict detection and trade-off analysis on early aspects in requirements goal models, and have seamlessly incorporated this FCA-based method with the PCT- and RGT-based early aspects alignment method

[3] This criterion is intuitive, since softgoals may contradict each other. We will show the FCA-based conflict detection method later in this section.

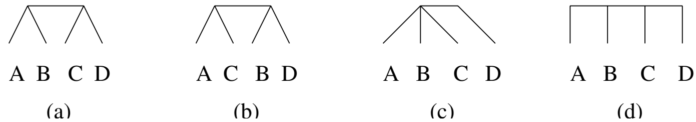

Fig. 16. Illustration of limitation of hierarchical clustering analysis. Even though the same information is presented, cognitions may vary among **a**–**d** due to different representational strategies: **a** cluster (AB) joins cluster (CD) **b** cluster (AC) joins cluster (BD) **c** cluster (ABC) joins D **d** single cluster (ABCD).

described in Sect. 3. A table of crosses represents a very simple and frequently used data type. We have showed how the reconciled and consolidated repertory grid can provide the cross-reference table from which the formal context of problem domain concepts can be derived: The set of elements in RGT are treated as objects in FCA, and personal constructs define the attributes in the formal context.

Although the translation from RGT to FCA is fairly straightforward and it seems that our composite approach replicates the same information in different formats: grid and table, we cannot replace one technique with the other. Firstly, each technique has an underlying theory and can only be applied under proper conditions and for proper tasks. Having roots in the psychology of personal constructs makes RGT suitable for concepts alignment among multiple stakeholder views; and with its origins in lattice theory, FCA offers mathematically rigorous analyses for a particular context.

Furthermore, these techniques are complementary to each other, and one of our purposes of employing a composite approach is to leverage the strengths of different approaches in overcoming each other's weaknesses. For example, the hierarchical clustering analysis of RGT is generally criticized because it can yield different inputs for the same data as in some instances several clusters are equidistant, leading to an arbitrary choice of the next clustering step [49]. This also increases analysts' cognitive overhead when they are in the process of identifying clustering potentials based on visual inspection, as illustrated in Fig. 16. FCA overcomes the problem since equidistant clusters are always grouped within one concept. Conversely, RGT helps to solve certain drawback of FCA. For instance, the constructs' reordering feature of FOCUS [42] addresses FCA's tool support limitation on analyzing trade-offs and detecting conflict, as pointed out by the media shop study in Sect. 4.2. The much needed slicing and projection features for FCA tools can be achieved by focusing on specific sub-clusters resulted from the FOCUS program. As a result, these techniques complement each other, and play appropriate roles in our coherent concept-driven framework.

Early aspects research in RE relies on stakeholders' inputs, and aims at presenting insightful analysis results to stakeholders for eliciting further information and feedback to better understand the problem world and the machine [15]. Thus, it is crucial for an early aspects framework to generate sensible analyses and plausible hypotheses to advance the RE process, and to guide system design and implementation. In that regard, our proposed approach is able to offer insights into early aspects' trade-offs and architectural solution rationale. For instance, in media shop, we analyzed the trade-offs between the aspects of "Accuracy" and "Confidentiality", and discussed the rationale for choosing particular implementations such as "SSL" and "PasswordProtection". Conflict detected in our approach can be used to drive negotiation strategies among stakeholders. The initial investigations into designing such a concept-driven framework also sharpen our understanding of the role modeling plays in RE: modeling is a way of telling people ideas, and like any telling, it clarifies; it also allows stakeholders to display and reflect on different ways of seeing the data.

5 Evaluation

An initial evaluation method for assessing the effectiveness of the concept-driven approach is defined on the basis of diffusion theory [38], which examines the rate and the motivations of adoption of a technological innovation by a group of potential users. Such an approach may also be fruitful for the evaluation of a novel conceptual framework (such as a design or requirements method), by assessing whether it is appreciated by a community of stakeholders [17].

The diffusion theory defines five perceived quality attributes of an innovative product. *Triability* is the degree by which the product can be tried on a limited basis before adoption. *Observability* refers to the observable results deriving from the use of the new product. *Relative advantage* is the perception of how much better the innovation is than the competing solutions currently adopted. *Complexity* refers to the fact that the innovative product should not be overly complex to understand and to use. *Compatibility* measures how the innovation is perceived as compatible and consistent with existing practices shared among the community of users [38].

On the basis of these attributes, a qualitative evaluation about the proposed approach was conducted, since we were still in the process of theory discovery and theory exploration, trying to reflectively learn something during the evaluation exercise rather than definitely test already known hypotheses. Qualitative data are records of observation or interaction that are complex and contexted, and they are not easily reduced immediately, or sometimes ever, to numbers. Qualitative research seeks to make sense of the way themes and meanings emerged and patterned in the data records built up from observations, interviews, document analysis, surveys and questionnaires, literature reviews, and other research media [37].

We used the media shop study described in this paper as an illustrative example to solicit feedback from seven requirements analysts, among whom five were computer science graduate students with some industrial RE experiences, and the other two were industry requirements engineers in small- and medium-sized companies.

Obviously, the number of sample users is not representative of the community of (goal-oriented) requirements modelers and analysts, and any quantitative data analysis will lack statistical significance and credibility. However, qualitative analysis methods, such as categorizing and coding used in our evaluation, can give an initial reaction to how such a systematic approach to early aspects is considered by requirements analysts.

Based on the quality criteria provided by diffusion theory, five free-form questions were designed. We collected data through interviews and questionnaires [21], and used coding (relating answer sections to proper quality attributes under testing) and categorizing (classifying answers to be positive or negative) [37] to perform qualitative data analysis. Requirements analysts' quotes are represented *italic* and cited in double quotation marks (" ") when analyzing the following five questions.

Question 1: According to your experience, do you think that this approach provides sufficient constructs and guidelines to be tested on a limited basis before adoption?

Quality Attribute: Triability.

The answer to this question was quite positive. Most requirements analysts were familiar with the notions of softgoals and tasks, and supported the idea of using contribution links to uncover and analyze goal models' crosscutting properties. To quote from feedback provided by an analysts: "... *(for one of our projects), we're currently conducting interviews (with multiple stakeholder roles), and would like to try out the (proposed) idea for separating concerns ...*"

Question 2: Do you see preliminary observable results from the application of the proposed approach to the analysis of crosscutting concerns in goal-oriented requirements engineering?

Quality Attribute: Observability.

The response to this question was very positive. Analysts found the results in the given media shop example "*non-trivial*", "*insightful*", and "*useful*".

Question 3: Compared to relevant techniques you are aware of, do you think that the adoption of the proposed approach can better help you improve the quality of the goal-oriented requirements analysis?

Quality Attribute: Relative advantage.

Most analysts did not provide answers to this question. Instead, some mentioned that "*there's yet requirements method (that I'm aware of) that focuses on crosscutting concerns (explicitly).*"

Question 4: Do you think that the proposed approach is overly complex to be understood and used?

Quality Attribute: Complexity.

The answers to this question diverged. Some analysts regarded the approach proposed was easy to apply since no new modeling notations were introduced. Others admitted that even though the concept lattice offered interesting analyses, the line diagram produced for any sizeable model "*was somewhat too cluttered and* *complicated to understand.*" Tools that supported focused representations would be desirable.

Question 5: Do you perceive the proposed approach to be compatible and consistent with the existing practices, values, standards, and technologies shared in your organization or institution?

Quality Attribute: Compatibility.

"*Too early to tell*" was a representative response to this question. Some analysts expressed concerns to the proposed approach, such as scalability, tool support, and so on.

Although this initial evaluation was preliminary and much improvement was needed for the proposed approach, analysts felt that having a conceptually precise treatment for early aspects was necessary and important, and RGT and FCA were to be readily used in RE to tackle interesting problems.

6 Related Work

Baniassad et al. [1] presented an integrated approach to manipulating early aspects and exploiting them throughout the software development life cycle. The approach emphasizes the use of intuition, domain knowledge, and heuristics to facilitate the identification of aspects in requirements. In goal-based modeling, softgoals represent tangled and scattered concerns, and have been treated as candidate early aspects recurrently in the literature (e.g. [25, 54]). Niu and Easterbrook [25] argued that to become true early aspects, not just candidate ones, softgoals need to have specific operationalizations, or advising tasks [54], to contribute to their satisfactions. Yu et al. [54] 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 approach extends previous work and existing knowledge, thoroughly analyzes the contribution links between tasks and softgoals in requirements goal models, and provides a conceptually rigorous framework for handling various concerns addressed in those models.

NFR catalogue [5] attempts to collate, from a wide range of sources, verifiable information on non-functional requirements in specific domains. One of the main motivations of catalogue or ontology building is the possibility of knowledge sharing and reuse: as soon as a particular domain (such as banking or meeting scheduling) is fixed, it seems reasonable to expect a large part of domain knowledge to be the same for a variety of applications, so that the high costs of knowledge acquisition can be better justified. Catalogue-based methods suggest that requirements analysts equip themselves with a glossary of standard terms to make communication easier and more precise. However, these methods can cause problems when stakeholders continue to use different interpretations of the terms codified in the catalogue. They also miss an important opportunity to explore differences in stakeholders' own categories so as to better understand their perspectives. Our approach takes the view based on personal constructs theory, which offers a mechanism to compare and contrast objects in the domain of interest. Our PCT- and RGT-based concepts alignment method avoids the problems of imposition of terminology when individuals construe and describe concepts in the problem domain, and the meaning of a term is essentially treated as a relationship between signs and actions in our framework. NFR or early aspects catalogue building could benefit from our approach since a well-categorized taxonomy is viewed as a catalogue of how constructs represented by linguistic symbols relate formally in a particular context.

Natural language processing (NLP) plays an important role in identifying aspects in requirements. Many early aspects frameworks (e.g. [6, 41, 46]) adapt certain steps in linguistic engineering (flatten, tokenize, stem, and remove stop words) so that aspect words (e.g. verbs that scattered in multiple requirements) can be extracted from requirements statements. Although NLP-based techniques could reach high recall and precision under certain circumstances, taking for granted that natural language-based requirements statements are an unproblematic starting point is a historical accident rather than a position grounded in the realities for RE [47]. We assume that there exists a relatively well-organized set of requirements models, and present a concept-driven framework that takes advantage of these models. Our approach complements existing NLP-based early aspects methods by providing mechanisms to capture and analyze overlapping, corresponding, conflicting, and crosscutting concerns addressed in fine-grained requirements models.

FCA has typically been applied in the field of software engineering to support software maintenance activities [49, 51], such as program understanding [52], object-oriented class identification [50], reengineering of class hierarchies [44], dynamic analysis [9], and software configurations [20]. In the analysis of software systems, especially source code exposing certain structural and behavioral properties, several relationships among the composing entities emerge. For this reason, FCA has found a very productive application area associated with static and dynamic analyses. Recent work, such as [36], has also reported the application of FCA in RE activities. The proposed approach exploits structural properties in requirements goal models, in particular, the binary contribution relations between tasks and softgoals in those models. Our novel application enhances the overall competence of FCA in exploring crosscutting properties in the early phases of software development.

The idea of translating RGT into FCA due to their common cross-reference data structure is not new. Much work has been carried out in the knowledge engineering field. For example, Delugach and Lampkin [7] presented a hybrid approach containing RGT and FCA to facilitate the knowledge acquisition process of transferring and transforming information from the domain experts to the expert system. In their approach, the traditional "triad" method [10] associated with RGT was applied to elicit knowledge from domain experts, and the translation from RGT to FCA was performed via an intermediate knowledge representation schema — conceptual graphs [45]. In contrast, our approach outlines a direct translation mechanism from RGT to FCA, thus seamlessly glues these two techniques together so that they complement each other's effectiveness. The coherent framework proposed in this paper is, to the best of our knowledge, the first attempt to employ a composite concept-driven approach to systematically characterize crosscutting concerns during RE activities.

7 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 [31], 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 our initial investigations into designing a composite concept-driven framework for capturing and analyzing early aspects in requirements goal models based on RGT and FCA. We have illustrated the proposed approach on a simple, but not simplistic, example. The results obtained from a preliminary qualitative evaluation of the approach suggest that RGT and FCA can be readily used during RE activities. We have mainly focused on analyzing stakeholder softgoals in the problem domain in this paper. In the future, we plan to extend the proposed concept-driven approach to facilitate the analysis of aspects in the solution domain or pertinent to functional requirements, such as buffering and caching.

Our approach is considered as an integral part of goal-oriented RE, with an emphasis on in-depth analysis of interdependencies between stakeholder goals. Crosscutting concerns are clarified and elaborated early to guide architectural design and implementation of the system, and to trace stakeholder interests onward. The resulting goal structure is modularly improved, so that code aspects can have a baseline to justify and validate their existence: are they images of stakeholder needs or reflections of implementation decisions?

From our initial experience with the proposed approach, we feel that the combination of RGT and FCA 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, and uncover early aspects' conflicts and trade-offs. In-depth empirical studies are needed to lend strength to the preliminary findings reported here. A combination of qualitative and quantitative analyses is needed to examine more quality attributes of the concept-driven framework, such as scalability, scope of applicability, relevance to functional requirements, and capability to deal with complex specifications. It is of critical importance to justify the level of effort involved in applying our approach, and to examine whether the effort expended (at such an early stage) will really affect the bottom line in the long run. We look forward to collaborating with researchers and practitioners in designing experiments and case studies to investigate these issues.

Our future work also includes developing efficient methods for producing a core set of common elements that a group of participants can all meaningfully construe. This is critical to all RGT-based approaches, and can lead discussion to exploring the ongoing debate about "whether elements exist independently of constructs, or whether in fact elements are also constructs". Also of interest would be addressing the need for improvement in some areas of FCA tool support, such as focused projections of concept lattice, query languages, line diagram slicing algorithms, etc. Finally, we plan to develop concept-driven conflicts resolution and requirements prioritization methods to explore aspects weaving.

All in all, the repertory grid and the concept analysis are truly techniques: a grid or a formal context of itself is nothing more than a matrix of blank cells. They are only limited by the user's imagination. We hope that our work has shed some light on their applications to new situations, especially, to aspect-oriented software development.

Acknowledgement. We would like to thank Paul Clements, Awais Rashid, João Araújo, Ana Moreira, Yijun Yu, Ruzanna Chitchyan, Haifeng Liu, Harvey Siy, and Erik Simmons for helpful discussions, and for insightful comments on the media shop example. We also thank the anonymous reviewers for their constructive suggestions. Financial support was provided by NSERC and BUL.

References

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

[2] Becker, P., Correia, J.H.: The TosCanaJ suite for implementing conceptual information systems. In: Ganter, B., Stumme, G., Wille, R. (eds.) Formal Concept Analysis. LNCS (LNAI), vol. 3626, pp. 324–348. Springer, Heidelberg (2005)

[3] Birkhoff, G.: Lattice Theory. Providence, RI.: Am. Math. Soc., 25 (1940)

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

[5] Chung, L., Nixon, B.A., Yu, E., Mylopoulos, J.: Non-Functional Requirements in Software Engineering. Kluwer Academic, Dordrecht (2000)

[6] Clarke, S., Baniassad, E.: Aspect-Oriented Analysis and Design: The Theme Approach. Addison-Wesley, Reading (2005)

[7] Delugach, H.S., Lampkin, B.E.: Troika: using grids, lattices and graphs in knowledge acquisition. In: Intl. Conf. on Conceptual Structures, pp. 201–214 (2000)

[8] Early aspects portal: (Cited on March 7, 2007), http://www.early-aspects.net/

[9] Eisenbarth, T., Koschke, R., Simon, D.: Locating features in source code. IEEE Transactions Software Engineering 29(3), 195–209 (2003)

[10] Fransella, F., Bell, R., Bannister, D.: A Manual for Repertory Grid Technique, 2nd edn. Wiley, NewYork (2004)

[11] Gaines, B.: An overview of knowledge acquisition and transfer. International Journal of Human-Computer Interaction 12(3-4), 441–459 (2000)

[12] Ganter, B., Wille, R.: Formal Concept Analysis. Springer, Heidelberg (1996)

[13] Hickey, A.M., Davis, A.M.: Elicitation technique selection: how do experts do it? In: International RE Conference, pp. 169–178 (2003)

[14] IDEF0 portal: http://www.idef.com/idef0.html (Cited on March 7, 2007)

[15] Jackson, M.: Problems, subproblems and concerns. In: Early Aspects Workshop, International Conference on AOSD (2004)

[16] Jutras, J.: An automatic reviser: the transcheck system. In: Christodoulakis, D.N. (ed.) NLP 2000. LNCS (LNAI), vol. 1835, pp. 127–134. Springer, Heidelberg (2000)

[17] Kaindl, H., Brinkkemper, S., Bubenko Jr., J.A., Farbey, B., Greenspan, S.J., Heitmeyer, C.L., do Prado Leite, J.C.S., Mead, N.R., Mylopoulos, J., Siddiqi, J.I.A.: Requirements engineering and technology transfer: obstacles, incentives and improvement agenda. RE Journal 7(3), 113–123 (2002)

[18] Kelly, G.: The Psychology of Personal Constructs, Norton, New York (1955)

[19] Kiczales, G., Lamping, J., Menhdhekar, A., Maeda, C., Lopes, C., Loingtier, J.-M., Irwin, J.: Aspect-oriented programming. In: Aksit, M., Matsuoka, S. (eds.) ECOOP 1997. LNCS, vol. 1241, pp. 220–242. Springer, Heidelberg (1997)

[20] Krone, M., Snelting, G.: On the interference of configuration structures from source code. In: ICSE, pp. 49–57 (1994)

[21] Lethbridge, T.C., Sim, S.E., Singer, J.: Studying software engineers: data collection techniques for software field studies. Empirical SE 10(3), 311–341 (2005)

[22] McKnight, C.: The personal construction of information space. Journal of the American Society for Information Science 51(8), 730–733 (2000)

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

[24] Niu, N., Easterbrook, S., Yu, Y.: A taxonomy of asymmetric requirements aspects. In: Early Aspects Workshop at AOSD (2007) (in press)

[25] Niu, N., Easterbrook, S.: Discovering aspects in requirements with repertory grid. In: Early Aspects Workshop at ICSE, pp. 35–41 (2006)

[26] Niu, N., Easterbrook, S.: Managing terminological interference in goal models with repertory grid. In: International RE Conference, pp. 303–306 (2006)

[27] Niu, N., Easterbrook, S.: So, you think you know others' goals? A repertory grid study. IEEE Software 24(2), 53–61 (2007)

[28] Novak, J.D.: Learning, Creating, and Using Knowledge. Lawrence Erlbaum, Hillsdale (1998)

[29] Nuseibeh, B., Easterbrook, S.M.: Requirements Engineering: A Roadmap. In: The Future of Software Engineering (2000)

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

[31] Nuseibeh, B.: Crosscutting requirements. In: International Conference on AOSD, pp. 3–4 (2004)

[32] Parnas, D.: On the criteria to be used in decomposing systems into modules. Communications of the ACM 15(12), 1053–1058 (1972)

[33] Pisan, Y.: Extending requirement specifications using analogy. In: ICSE, pp. 70–76 (2000)

[34] Popper, K.: Epistemology without a knowing subject. In: International Congress for Logic, Methodology and Philosophy of Science, pp. 333–372 (1968)

[35] Potts, C., Takahashi, K., Anton, A.: Inquiry-based requirements analysis. IEEE Software 11(2), 21–32 (1993)

[36] Richards, D.: Merging individual conceptual models of requirements. RE Journal 8(4), 195–205 (2003)

[37] Richards, L.: Handling Qualitative Data: A Practical Guide. Sage, Beverely Hills (2005)

[38] Rogers, E.M.: Diffusion of Innovations, 4th edn. Free Press, NewYork (1995)

[39] Ryle, A.: Frames and Cages: The Repertory Grid Approach to Human Understanding. Sussex University Press, Sussex (1975)

[40] Sabetzadeh, M., Easterbrook, S.: Traceability in viewpoint merging: a model management perspective. In: International Workshop on Traceability in SE (2005)

[41] Sampaio, A., Chitchyan, R., Rashid, A., Rayson, P.: EA-Miner: a tool for automating aspect-oriented requirements identification. In: International Conference on ASE, pp. 352–355 (2005)

[42] Shaw, M.: On Becoming A Personal Scientist: Interactive Computer Elicitation of Personal Models of the World. Academic Press, Dublin (1980)

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

[44] Snelting, G., Tip, F.: Reengineering class hierarchies using concept analysis. ACM Transactions on Programming Languages System 22(3), 540–582 (2000)

[45] Sowa, J.F.: Conceptual Structures: Information Processing in Mind and Machine. Addison-Wesley, Reading (1984)

[46] Suen, R., Baniassad, E.: Isolating concerns in requirements using latent semantic analysis. In: Early Aspects Workshop, OOPSLA (2005)

[47] Sutton, D.C.: Linguistic problems with requirements and knowledge elicitation. RE Journal 5(2), 114–124 (2000)

[48] Tilley, T.: Tool support for FCA. In: Eklund, P.W. (ed.) ICFCA 2004. LNCS (LNAI), vol. 2961, pp. 104–111. Springer, Heidelberg (2004)

[49] Tilley, T., Cole, R., Becker, P., Eklund, P.: A survey of formal concept analysis support for software engineering activities. In: Ganter, B., Stumme, G., Wille, R. (eds.) Formal Concept Analysis. LNCS (LNAI), vol. 3626, pp. 250–271. Springer, Heidelberg (2005)

[50] Tonella, P.: Concept analysis for module restructuring. IEEE Transactions Software Engineering 27(4), 351–363 (2004)

[51] Tonella, P.: Formal concept analysis in software engineering. In: ICSE, pp. 743–744 (2004)

[52] Tonella, P.: Using a concept lattice of decomposition slices for program understanding and impact analysis. IEEE Transactions Software Engineering 29(6), 495–509 (2003)

[53] Yu, E.: Modeling strategic actor relationships for process reengineering. PhD Thesis, University of Toronto (1994)

[54] Yu, Y., do Prado, J.C.S., Mylopoulos, J.: From goals to aspects: discovering aspects from requirements goal models. In: International RE Conference, pp. 38–47 (2004)

[55] Zave, P., Jackson, M.: Four dark corners of requirements engineering. ACM TOSEM 6(1), 1–30 (1997)