Using Goal-Models to Analyze Variability

Bruno González-Baixauli
Universidad de Valladolid
bbaixauli@infor.uva.es

Miguel A. Laguna
Universidad de Valladolid
mlaguna@infor.uva.es

Julio Cesar Sampaio do Prado Leite
PUC-RIO
http://www.inf.puc-rio.br/~julio/

Abstract

On our ongoing research on variability analysis, two major drawbacks have been identified: a) the lack of an effective tracing from the rationale to the selected variant, and b) the lack of an effective support to decide what variability should be implemented. In order to perform variability analysis we propose to combine goal-oriented requirements strategies with the concept of early aspects. This paper aims to contribute to the general discussion of variability, in particular variability analysis. We stress some of our initial findings on a meta-model to handle the amalgamation of a goals network with the concept of early aspect. We discuss some of the problems we are facing and the possible ways of addressing them.

1. Introduction

It is our belief that anchoring variability on the realm of goal and softgoal interrelationships is a strong base to tracing. We also believe that organizing the complex interrelationships as aspects contributes to meliorate the central issue of complexity.

The study of systems, taking into account different possible variants or configurations, has been approached from different perspectives. Software Product Lines [3], where variability is embedded in a portfolio of different products sharing a common core, is already been used in industry. Software Customization [13][14] is an approach where all variants are in one single system and the focus is to study what customization is needed by a single user. Exploring alternatives [15] focus on exploring a space of alternatives before selecting the one to be implemented. The work on software configuration management deals with the problem of managing the set of different system compositions and versions for a given system.

Variability can be analyzed from different abstraction levels, but since variability is constrained in each level [19], the sooner the better. Therefore, we propose to start from a requirements perspective.

Deciding which variability must be in the system is a challenging task. Feature oriented domain analysis [11] has aimed to gather abstract concepts of the domain and to organize them as features; yet there is a gap. The early handling of variability is present in different lines of recent research. Halmans and Pohl deal with this using a variation of use cases [9], Schmid [18] uses product portfolios and Lee, Kang and Kim [10] use the idea of product planning. We explore the representation of variability in the context of goal oriented requirements engineering [19].

Features, usually, do not consider, explicitly, the semantics of the interplay among functional and non-functional requirements. As such, those models lack foundations to reason about the relationships among different requirements.

As an addition to being critical for software-intensive systems [2][13][21], Non-Functional Requirements (NFRs) may serve as basis for reasoning about choices. The NFR Framework [2], where AND/OR graphs are used to represent choices of different options of how a given NFR (softgoal, i.e. goals with non clear-cut satisfaction criteria) may be realized, appears to be a reasonable choice to reason about variants.

Using a network of goals and softgoals, different interrelationships may be possible when a goal is satisfied or/and a softgoal is satisficed (i.e. sufficiently satisfied). Choosing to satisfice a softgoal implies in selecting a variant. As such, the softgoal (NFR) is the natural rationale of why a given operationalization (variant) was selected, that is how a given goal will be satisfied given certain restrictions (softgoal).

Goal oriented approaches provides certain "separation of concerns" by modeling different concerns in different AND/OR graphs. But this separation of concerns is partial, since all functionality is finally included in the same graph (both function and non-function goals will need to be operationalized as functions), tangling and scattering concerns. This tangling/scattering is the reason to use aspect orientation, what applied to requirements is named early-aspects [1].

Using the goal softgoal network (AND/OR graphs) annotated with contributions and correlations (interrelationships) brings up the issue of complexity. The better modularization provided by aspect orientation seems to be a reasonable way of dealing with this complexity [17][21].

We are exploring the combination of goal graphs and aspects as to support a better way of producing requirements models with variability, as well as a systematic way of choosing variability (customization).

In the next sections we present our ideas: on the process of dealing with variability (Section 2), on the problems we see as the choices grow (Section 3), on our initial modeling strategy (Section 4), and on dealing with complexity (Section 5). We conclude (Section 6) stressing our findings, problems, and the work we believe lies ahead.

2. Variability Analysis

Our aim is to deal with variability from an intentional and quality driven viewpoint. We see variability analysis as split in three different parts. The SADT model of Figure 1 shows how these parts interact.

EXPLORE - In this part functional and non-functional requirements are modeled to find candidate variability (requirements). Here, concerns are identified (reusing Non Functional Requirements Catalogues [4]), and decomposed successively to define the variability space. Also, crosscutting relationships between concerns' elements are defined. This variability space can be huge, with too many variants to implement; therefore it must be limited to have an affordable system.

PRUNE - Prune space: first analyzing locally solutions in concerns to drop not good enough solution; then, analyzing how these solutions affect other concerns, and deciding what combination must be implemented in the variable system. The result of this analysis is the variability model of the domain (pruned variability model).

REUSE - Previous phases are aimed towards analysis *for* variability (domain engineering), but our goal model is also useful for analysis *with* variability, that is, provide a customized product for a given client. The analysis with variability is a quest for an "optimal" configuration as to fit the product requirements.

Goal-oriented approaches are a good candidate for supporting the three parts. First, in variability exploration, goals allow to start exploration from a higher abstraction level. Abstract functional goals are decomposed in parts (And decomposition) or alternative means to achieve them (Or decomposition). An important advantage is that those alternatives are related with other goals by correlation links. Correlations are defined in the NFR Framework to relate functional elements to, the NFR (softgoals) they affect, producing a rationale for alternatives selection.

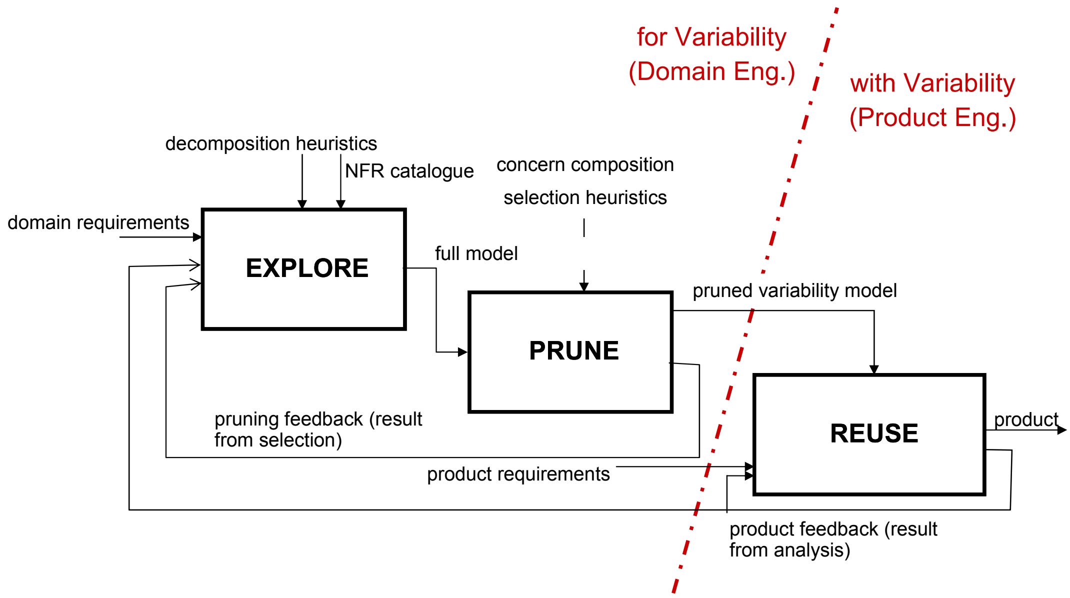

Figure 1: Variability Analysis SADT

Figure 2 shows how the rationale for different *Login* variants can be represented. In this case, three alternatives are given: a) *use password*, which is not so good for authenticate user, and hurts memorability, but it is a widely used mechanism, and consequently more natural; b) *fingerprint*, which is good to assure user authenticity, but hurts commonality, and cost for the user (he needs to buy a fingerprint device); and c) using an *internet certificate*, which also assures authenticity, but hurts naturalness. NFR are not only used to justify alternatives, applying the NFR Framework [2] and the softgoal concept, it is possible to deal with NFR in variability analysis. Softgoals are also decomposed from high level NFR until operationalized NFR (solution), but because of their fuzzy nature, their solutions are in different degrees (similar to correlations).

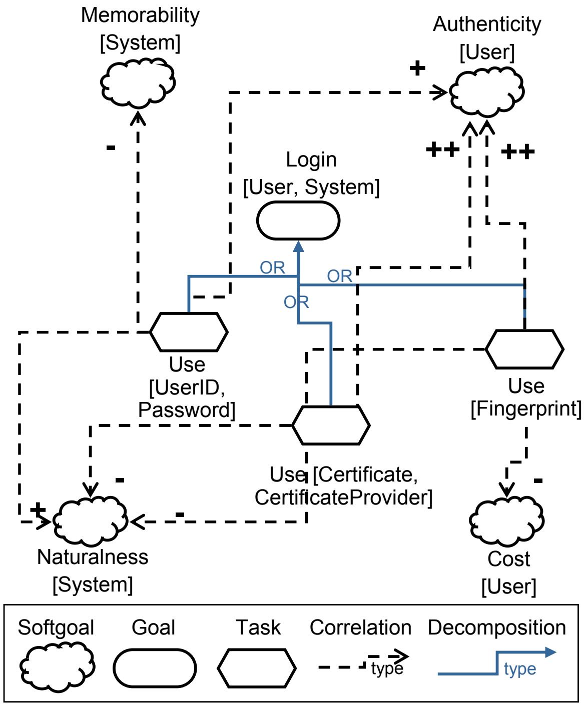

Figure 2: NFR are the rationale for variant decision

On the other hand, early-aspect ideas [1] are a natural way to integrate functional and non-functional models, and to improve modularity in goal oriented approaches. Good modularity is basic in variability analysis since dealing with multiple solutions for a problem increases complexity. Better modularity provides chances of a deeper analysis since the problem is separated in multiple parts, and therefore more variability can be explored. Also, aspect weaving provides a new variability mechanism since an NFR can be composed or not.

Variability concerns (models) must be composed in PRUNE to find the right variability to be implemented. This composition will produce again the problem of complexity, and how to manage it. In our proposal, different non-functional requirements can be combined with functional goals. So, solutions of the non-functional are combined with functional, but also combinations of the non-functional solutions are also possible.

To analyze variability, all possible combinations have to be taken into account, but the number of combinations grows exponentially on the number of non-functional requirements and their solutions. Thus, dealing with this complexity is fundamental. Therefore, we divide PRUNE in three steps: a) concerns are locally analyzed and all goals out of interest or not-enough good solutions[1] are eliminated; b) analyze composition in tasks: for each functional task we look for softgoal solutions that can be composed with it (given by crosscutting relationships), so possible compositions are evaluated; c) with a composed system, it is possible to find better variants giving priorities to different softgoals as done in [7].

Goal oriented variability analysis is useful in selecting the best variant (see REUSE in Figure 1), given product requirements (functional and non-functional). In that context, our previous work [7] uses a functional goal model to calculate the variants, a contribution matrix to relate them to the softgoals (NFR), and a priority scheme that returns a set of charts that compare the variants to choose the optimal one. This work is compatible with our ideas, since the result of composition is a functional model.

3. Goal Composition

Model composition is basic for our proposal, since we need a mechanism to join concerns and get a complete vision of the system. Since our aim is to analyze variability, our focus in composition is to discover what combinations are possible, i.e. we need to know how different goal models can be composed with others. Therefore, we do not need a complex composition mechanism as in early-aspects or AOSD [1][16][17].

Because tasks (Tasks are the functional operationalization of both goals and softgoals) are the specific elements of goal models (versus the generality of goals and softgoals), composition is done at task

[1] For instance, in Figure 2 *Use [Fingerprint]* has the same advantages and disadvantages than *Use [Certificate, CertificateProvider]*, plus another disadvantage.

level, generating new task that will achieve goals and correlate softgoals. Figure 3 shows a simple composition where a Softgoal (SG2) with two possible solutions (with contribution value c1 and c2) is related to a Goal (G1) decomposed in two tasks (decomposition can be And or Or).

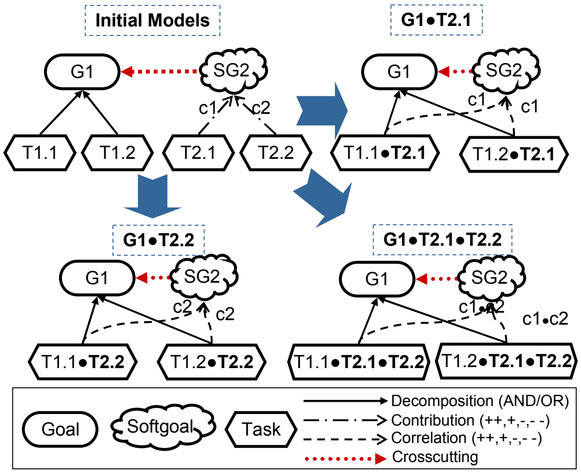

Figure 3: Possible compositions for a goal (G1) with two sub-task and a softgoal (SG2) with two solutions

This simple composition produces three new variants: one where Goal subtasks are combined with T2.1 solution of SG2, another where they are combined with T2.2, and one more with the combination of both solutions if they are compatible. Actually, compatibility is a decision of the domain engineer. Note that contributions (decompositions in softgoal graph) are transformed to correlations to represent that composed tasks are solutions of functional goal, instead of non-functional softgoals. Considering that G1 is decomposed by And, we will have four variants for it, but if it is an Or, variants will be multiplied by the number of subtasks, that is the number of variants for this goal. Therefore, we need a mechanism to deal with the number of possible compositions, and consequently variants.

Note that all these variants should be represented in the same model as shown in Figure 4, since our main interest is to show and analyze variability. Figure 4 represent different means to get the password in a Login system. It can be understood in two ways: as the way that the goals is represented in a typical goal model; or as the result of composing the goal Get [Pass] with two solutions (Ask or Use previous password) and Privacy of password, with two operationalizations (Show password with asterisks instead of letters and Hiding the input). This graph shows the resulting composition, therefore the initial elements are represented as references (dotted elements), but also new solutions taking into account the softgoal and new correlations links are shown. In this case, softgoal solutions are not compatible, so there is only 2 new goals (has no sense show asterisks and hide the input at the same time). Note that other contributions are not shown (e.g. Use [Previous] hurts Privacy [System] or Hide [Input] hurts Usability, since number of letters is not shown).

In general, for a goal G1 decomposed in N tasks: G1{T1.1, T1.2, …, T1.N}; and a softgoal SG2 solved by M tasks: SG2 {T2.1, T2.2, …, T2.M}, we will have 2^M variants (combinations of 0 elements for the initial solution, of 1 element for composition of one solution, …, of M elements for all solutions composed) if G1 decomposition is an And or $Nx2^M$ if it is an Or. Nor M or N will usually be too big, but several Softgoals can be composed at the same time, in this case we will have $Nx2^L$, where L is the sum of all solutions of the

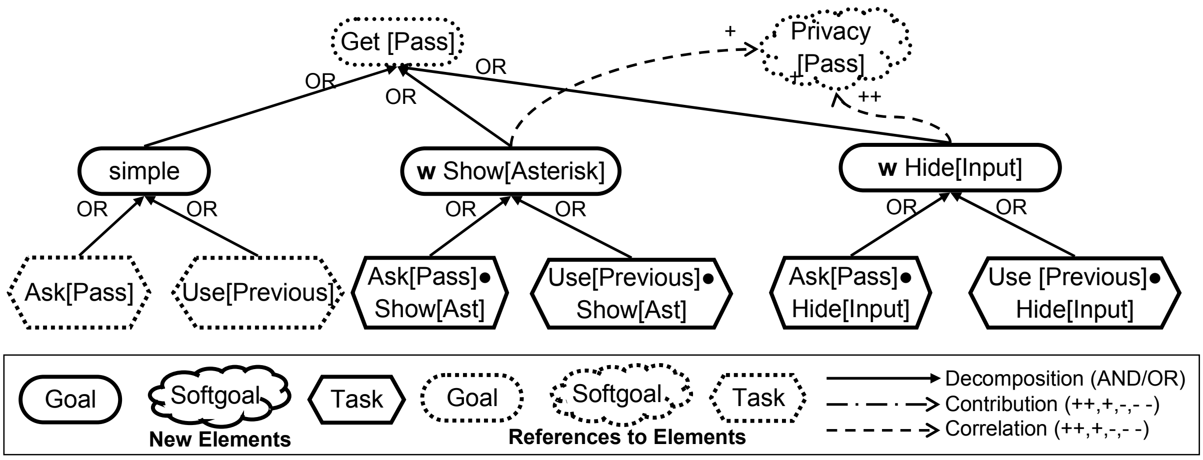

Figure 4: Goal model showing different means for get password in a login

softgoals that affect the goal. Considering that softgoals may interact at different levels of a goal graph, it is easy to imagine that we are facing a combinatorial explosion.

4. Variability Model

In this section, we present an initial idea of the variability model needed to support the intentional and quality based variability analysis presented. In our proposal, variability is represented by goal oriented relationships (Or-decomposition and contributions), and by aspect oriented since concerns can be composed or not.

This model has to deal with the characteristics of our proposal that are: 1) to support goal oriented elements as goals, softgoal and task, and their decomposition; 2) to allow goal analysis, by defining correlation and contribution relationships; 3) to deal with concerns, separating them in different models; 4) to define where tasks must be adapted to achieve softgoal satisficing; 5) to support composition, including new models to contain the resulting composition.

We take as basis of our model the goal oriented models of [21] and [12], but adding the idea of a meta-model as to organize relationships among different sets (named **Concern**).

Meta modelling is also applied to the model **View,** created by composition of other models. Existing approaches that integrate goal with aspects usually create completely new models when they make the composition, but we support the idea of references to avoid new elements, and the possible inconsistencies.

Figure 5 shows a representation of the modelling elements, and their relationships: a Project is composed by Concerns and Views. Concern is a model with goals and tasks (usually functional) or with softgoal and tasks (usually non-functional). All elements can be decomposed (And or Or) by the same kind of element, but tasks can also decompose goals and tasks. Goals, function tasks and softgoals can correlate, indicating how they affect (make, help, hurt and break), softgoals. Softgoal tasks and Softgoals can contribute, solving at some degree (make, help, hurt and break), a softgoal. Correlations and contributions are different since contributions have a decomposition meaning, task that correlates a softgoal solves that softgoal in some degree. Meanwhile, correlations represent how decomposition of other elements affects the softgoal.

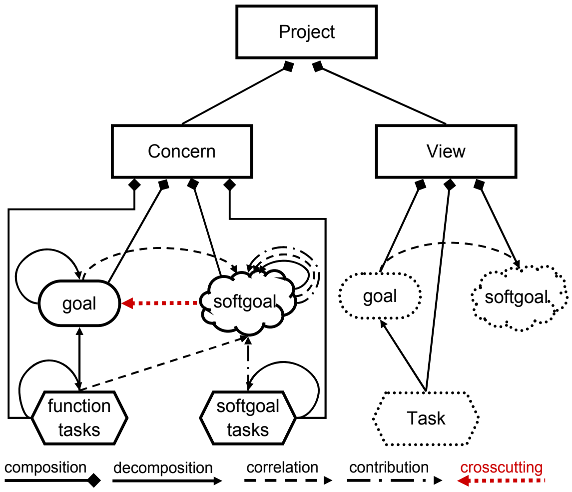

Figure 5: Modeling elements and relationships. Dotted elements can be references or elements

Views are composition of other models (Concerns or Views), and they have references to Concern's elements (goal, softgoals or tasks), but also can have new elements as a result of the composition (see Figure 4).

Note that in composition, new goals gain a correlation of the same value of the contribution from softgoals task composed with it to the corresponding softgoal.

We defined a first version of the modelling elements and relationships in [8] using a meta-model. Also, we implemented a prototype modelling tool using a meta-modelling application: Generic Modeling Environment (GME 5) [5].

5. Dealing with Complexity

Complexity in this environment has two causes: a) goal models usually integrate all functionality in one model: b) variability space size. Our first solution to complexity is to divide the system in different models, modularizing concerns as much as possible. To support this, we applied an aspect oriented approach. In addition, better model modularization implies more independence in concerns, what permits to perform local analyses, as explained before (Section 3). So, a first step of variability decision is done locally, analyzing solutions in concerns, and taking off what are not a "good enough" solution.

But this is not enough, it is necessary to analyze how those solutions affect each system part, which in goal-oriented approaches is represented by tasks. Therefore, we need to know what softgoal solutions affect to each task. However, representing the graph with all possible combination of tasks with softgoal solutions that affect them will create a too huge model (see Figure 4 for two simple graphs). This task analysis could be done by following crosscutting relationships in an indirect way, however since crosscutting relationships can affect different goals in the decomposition hierarchy; it is difficult to know what softgoals affect a task. To solve this problem, we propose to use a labelling mechanism to attach what softgoals (and solutions) can affect a sub-graph (tree).

Figure 6 shows an example of labels for two simple goal tree (goal G1.2 decomposition is not shown). The labelling algorithm will go through the softgoal graph since they are the source of crosscutting relationships, and will create the label structure. Then, it will go through the goal graph from the target of crosscutting and will add the structure to the tasks. Note that lower level softgoals can be more specific in their crosscutting relationships as happen with SG2.1.

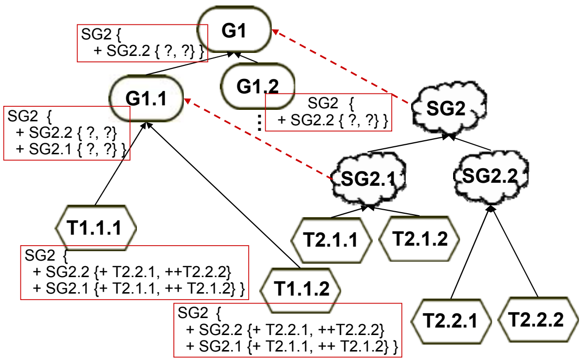

Figure 6: Composition labels example. SG2 affects all sub-goals, but SG2.1 only G1.1

Not only tasks have labels, goals too. This is a structure to analyze if a softgoal's solution is going to be used or not. So, task compatibility analysis will result in what task compositions are compatible and can be on the system, but this local analysis can produce partial compositions, not enough to contribute to the softgoal. Therefore, selections in tasks must be propagated to upper goals as in the Label Propagation Algorithm in [2][6]. Here, since the model is functional, we have only the decomposition relationship (without correlations as in softgoal graphs): if we have an And, all tasks should implement the composition to propagate. In the other hand, if it is an Or, only one is needed. Finally, labels of the root goal (considering the crosscutting relationship sub-tree) will provide us what softgoal's solutions are going to be implemented. Figure 7 shows an example of propagation: G1.1 is And-decomposed in two tasks, but in analysis of the composition of T1.1.1 with T3.1 it was concluded that it will not be in the system. So, even if the composition of the other task (T1.1.2) could be done, the propagation shows it is not enough, and deletes softgoal's solution T.3.1.

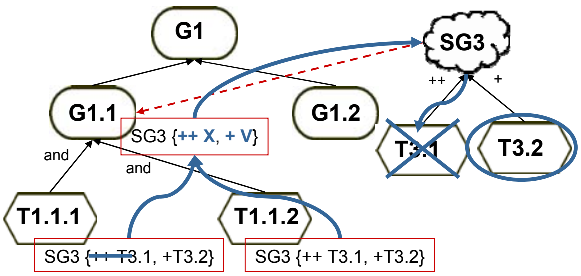

Figure 7: Label Propagation scheme

Obviously, it has no sense to have T1.1.2 with T3.1, consequently the analysis should be redone to allow the combination with the first task or discard the one with the second task.

Label propagation, as shown, is not enough to analyze the combination of several softgoals, e.g. compose T1.1.1 with T2.2.1 *and* T2.1.1 at the same time. Labels should be enriched with all possible combinations.

Note that this labelling is done from the concern definition, similar to weaving in AOSD approaches. Therefore, since concerns are defined independently, separation of concerns is not affected.

We are looking for yet another mechanism to help shrinking the variable space. We believe that annotations with hints to restrict the possible combinations are a good idea. These restrictions would be based on softgoal interference, if more than one softgoal was been considered. As such, some possibilities could be eliminated due to pre-defined unwanted interferences.

We have implemented a composition mechanism, without new task creation, using XSLT [8]. First results are promising and we are planning to continue as we consider our annotation model.

6. Conclusion

In this paper we have explored how goal models and early aspect concepts can be applied to deal with variability, and to decide what should be in the variable system-to-be. Goal model brings to light the rationale behind variability by linking variants with softgoals. This is achieved by means of correlations as in the NFR framework. Combining goal modeling with aspects improves modularity and helps dealing with complexity.

We have shown how complex the models turn out to be as one builds with variability. Our approach is to break the variability space, pruning the model as much as possible as to reduce its complexity and the complexity of its analysis. We have shown a preliminary strategy based on annotations that seem to be worth pursuing.

Several problems lie ahead. How to define the composition operator, and its properties? How to allow crosscutting relationships between functional goals (not only from softgoals to goals)? How to refine the composition mechanism, allowing some types of task to be affected (constraints similar to when in [12])? How to deal with alternative crosscutting relationships, that is, crosscutting relationship from one softgoal to several goals, with one being enough to achieve the goal? How to deal with more operationalizable softgoal solutions that could be added as goals instead tasks (as in *Login [System]*, or *Log [Transactions]*, or *Encryption [Data]*)? How to analyze the reusability of concerns (here [12] can help)? How to handle anti-goals (obstacles)?

Finally we need tool support. We have developed a simple modeling environment [8] using a meta-modeling tool (GME 5 [5]), but more work on implementing composition and labels is needed.

7. References

[1] R. Chitchyan, A. Rashid, P. Sawyer, A. Garcia, M. Pinto Alarcon, J. Bakker, B. Tekinerdogan, S. Clarke, and A. Jackson, "Survey of Aspect Oriented Analysis and Design Approaches", *AOSD-Europe Network of Excellence*, Avail. at http://www.early-aspects.net/ (Last visited on Oct 06)

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

[3] Clements, P.C. and Northrop, L. *"Software Product Lines: Practices and Patterns"*. SEI Series in Software Engineering, Addison-Wesley. 2001.

[4] L.M. Cysneiros, E. Yu, and J.C.S.P. Leite "Cataloguing Non-Functional Requirements as Softgoal Networks" In: Requirements Engineering for Adaptable Architectures @ RE'03. pp.13-20

[5] Generic Modeling Environment (GME 5). Vanderbilt University (Nashville). Available at (Last visited on Oct 06) http://www.isis.vanderbilt.edu/projects/gme/

[6] P. Giorgini, J. Mylopoulos, E. Nicchiarelli, and Sebastian, R., "Reasoning with goal models", P*roc. of ER'02*, Tampere, Finland, October 2002.

[7] B. González-Baixauli, J.C.S.P. Leite, and J. Mylopoulos, "Visual Variability Analysis with Goal Models", *Proc. of the RE'2004*, IEEE Computer Society, Sept. 2004, Kyoto (Japan), pp. 198-207.

[8] B. González-Baixauli, M. Laguna, and J.C.S.P. Leite, "A Meta-model to Support Visual Variability Analysis" (submitted to conference).

[9] G. Halmans, K. Pohl, "Communicating the Variability of a Software-Product Family to Customers", *Journal of Software and Systems Modeling 2(1),* 2003, pp. 15-36.

[10] J. Lee, K.C. Kang, and S. Kim. "A Feature-Based Approach to Product Line Production Planning", 2004.

[11] K.C. Kang, S. Cohen, J. Hess, W. Nowak, and S. Peterson, "Feature-Oriented Domain Analysis (FODA) Feasibility Study", *Technical Report, CMU/SEI-90-TR-21*, SEI (Carnegie Mellon), Pittsburgh, 1990

[12] J.C.S.P. Leite, Y. Yu, L. Liu, E.S.K. Yu, and J. Mylopoulos "Quality-Based Software Reuse". *Proc of CAiSE 2005*, Porto (Portugal), pp. 535-550

[13] S. Liaskos, A. Lapouchnian, Y. Wang, Y. Yu, and S. Easterbrook, "Configuring Common Personal Software: a Requirements-Driven Approach", *Proc of RE'05*, Paris, (France), 2005, pp. 9-18.

[14] S. Liaskos, A. Lapouchnian, Y. Yu, E. Yu, and J. Mylopoulos, "On Goal-based Variability Acquisition and Analysis," *Proc of RE'06*, Minneapolis/St. Paul (Minnesota), sept 2006, pp. 76-85,

[15] J. Mylopoulos, L. Chung, S. Liao, H. Wang, and E.S.K. Yu, "Exploring Alternatives during Requirements Analysis". *IEEE Software 18(1)*, jan 2001, pp.92-96

[16] A. Rashid, A. Moreira, and J. Araujo, "Modularisation and Composition of Aspectual Requirements", *Proc. of 2nd AOSD*, 2003, ACM, pp. 11-20.

[17] L.F. Silva, "An Aspect-Oriented Requirements Modeling Strategy", *Tese de Doutorado. Departamento de Informática, PUC-Rio*, Rio de Janeiro, (Brazil), 2006 (in Portuguese)

[18] Schmid, K. "Strategically Defining and Exploiting Product Portfolios with a Product Line Approach", in Perspectives on Software Requirements, Kluwer, 2004

[19] J. van Gurp, J. Bosch, and M. Svahnberg, "On the Notion of Variability in Software Product Lines", *Proc. of WICSA 2001*, Amsterdam (The Netherlands), aug. 2001

[20] A. van Lamsweerde, "Goal Oriented Requirements Engineering", *Proc. of 5th IEEE Intl. Symposium in Requirements Engineering*, Toronto (Canada), aug. 2001, pp.249-263

[21] Y. Yu, J.C.S.P. Leite, and J. Mylopoulos, "From goals to aspects: discovering aspects from requirements goal models", *Proc. of RE'04*. Kyoto (Japan), sep.2004, pp.38-47.