Identifying Candidate Aspects with I-star Approach

Fernanda Alencar[1*], Carla Silva[2], Ana Moreira[3], João Araújo[3], Jaelson Castro[2†]

[1]Dept. Eletrônica e Sistemas - Universidade Federal de Pernambuco
50740-530, Recife – PE – Brazil; +55 81 2126-8995

fmra@ufpe.br

[2]Centro de Informática, Universidade Federal de Pernambuco
50732-970, Recife – PE – Brazil; +55 81 2126 8430

{ctlls, jbc}@cin.ufpe.br

[3]Dept. Informática, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa
2829-516 Caparica, Portugal; +351 21 294 85 36

{amm,ja}@di.fct.unl.pt

ABSTRACT

When developing systems, we should have a broad understanding of the organizational environment and the goals that govern it. The i* framework is a well-equipped technique for capturing this information since it can represent how the intended system meets organizational goals, why the system is needed, what alternatives were considered, what the implication of the alternatives are for the various stakeholders, and how the stakeholders' interests and concerns might be addressed. The aim of this paper is to use the i* (eye-star) models to identify crosscutting concerns early in the software development. To accomplish this we start by analyzing the strategic dependency and strategic rational models to identify those model elements that might cut across several other elements. The result is then validated by applying a set of heuristics to derive a use case model from the i* models and then identify potential crosscutting concerns in the resulting UML model.

Keywords

Requirements engineering, organizational modeling, early aspects, UML

1. INTRODUCTION

Requirements engineering has been considered a key activity in any software engineering process. It is well known that a requirements specification should include not only software specifications but also business models and other kinds of information describing the context in which the intended system will function [20].

Current goal-oriented approaches have demonstrated that the use of goal analysis offers benefits and advantages at almost every stage of the requirements engineering process. In goal elaboration, the modeling of goals has been proposed during requirements elicitation in order to describe organisational behaviour. The i* (eye-star) approach [20, 21] provides a description of work organisation in terms of dependency relationships among actors.

Goal-oriented approaches can help identifying and separating concerns in requirements engineering [7, 9, 10, 20, 21]. However, the crosscutting nature of some of those concerns is not handled explicitly. This paper reports on an investigation of how the i* models can be used to identify crosscutting concerns [4] during early requirements modeling. The results are then validated by first deriving a use case model from the i* models using [18, 19] and then using [1, 2, 3, 5] to identify crosscutting concerns in a use case model.

This paper is organised as follows. Section 2 presents some background that has been used as a starting point for our own. Section 3 illustrates i* by means of a case study (this case study is then used to illustrate our approach). Section 4 discusses how i* models can be used to identify crosscutting concerns. Section 5 validates our proposal by using other techniques to achieve equivalent results. Section 6 discusses some related work. Finally, Section 7 summarizes our work and points out open issues.

2. BACKGROUND

This work is built on three other approaches: the i* framework, the derivation of use cases from the i* [20] and the aspect-oriented requirements with use cases and sequence diagrams in UML [1, 3, 5, 12].

2.1 The i* Framework

When developing systems, we usually need to have a broad understanding of the organizational environment and goals. To model and understand issues of the application and business domain a developer can use the *i** framework which allows a better description of the organizational relationships among the various agents of a system as well as an understanding of the rationale of the decisions taken. The i* models provide a framework for understanding the organisational environment and goals.

---

[*] Currently on leave of absence at Universidade Nova de Lisboa, Portugal
[†] Currently on leave of absence at IRST, Trento, Italy

The participants of the organisational setting are actors with intentional properties such as goals, beliefs, abilities and commitments. The term actor was used to refer generically to any unit to which intentional dependencies could be ascribed. An intentional actor does not simply carry out activities and produce entities, but has motivations, intents, and rationales behind its actions. Actors are strategic in the sense that they are not merely focused on meeting their immediate goals, but are also concerned with opportunities and vulnerabilities, and seek rearrangements of their environments that would better serve their interests [21].

The i* technique consists of two models: The Strategic Dependency Model (SD) and Strategic Rationale Model (SR). SD includes a set of nodes and links connecting them, where nodes represent actors and each link indicates a dependency between two actors (see example in Fig. 1). The links can be goal, task, resource and soft-goal dependencies.

The SR model is used to: (i) describe the interests, concerns and motivations of participants' process; (ii) enable the assessment of the possible alternatives in the definition of the process; and (iii) research in more detail the existing reasons behind the dependencies between the various actors. Nodes and links also compose this model. It includes the previous four types of links (present in the SD model) and also two new types of relationships: (i) means-end that suggests that one model element can be offered as a means to achieve another model element; (ii) task-decomposition that describes what should be done in order to perform a certain task. Fig. 3 shows an example. Finally, i* supports the analysis of opportunities and vulnerabilities for different actors.

To clarify all the concepts involved in the i* (eye-star) approach, Section 3 presents a case study modeled using this framework.

2.2 Integrating I* with UML

Castro et al. [6, 8] claim that UML alone is not adequate to deal with all different types of analysis and reasoning that are required during the requirements capture phases. They advocate the use of two complementary modeling techniques, *i** and *UML*. The organizational requirements, in this context modeled with i*, must be related to functional requirements represented with techniques such as use cases. Santander and Castro assure that the heuristics presented in the literature to develop use cases are not sufficient to allow a systematic development [19]. Indeed, they do not consider relevant organizational concerns such as goals and softgoals. Based on this claim, they propose a set of guidelines to support the integration of i* and use case modeling that allow requirement engineers to derive use cases (and associated scenarios) from i* organizational models (see Fig. 1). These guidelines will be used in Section 5 as a stepping stone to validate our proposal.

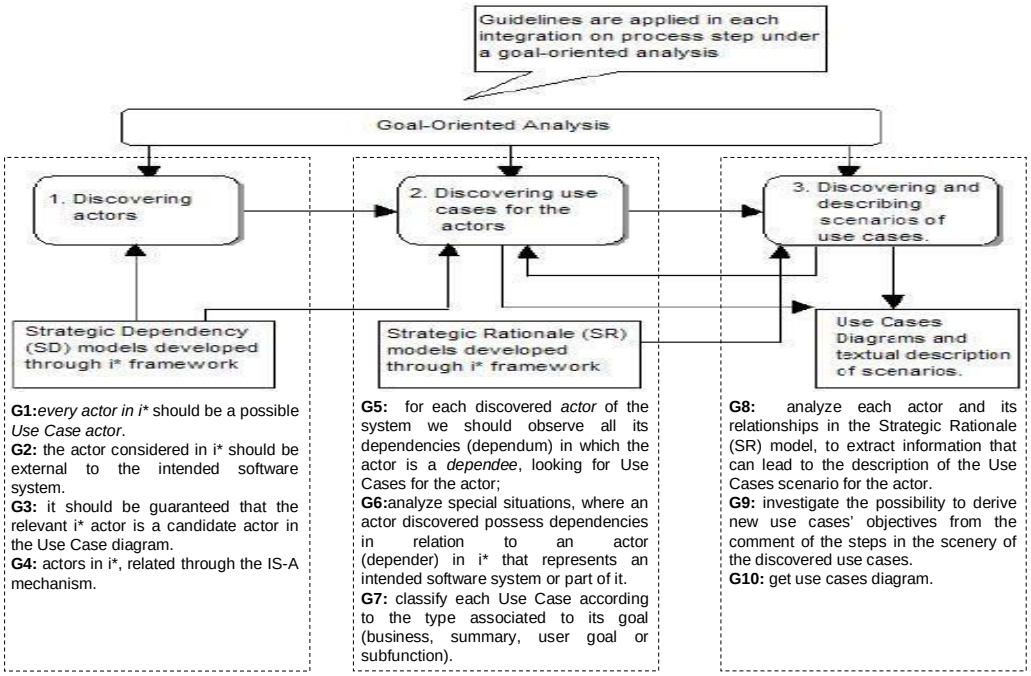

Figure 1. Steps of the Integration Process bet between i* and Use Cases in UML

In this figure, steps 1, 2 and 3 represent the discovery of system actors and its associated use cases and scenarios. The input for the integration process are the SD and SR models developed through i* framework. In steps 1 and 2, the input is the SD model. The description of scenarios for use cases (step 3) is derived from elements represented in the SR model. The result is a use case model and scenarios, described textually, for each use case.

Step 1, *Discovering Actors*, proposes four guidelines (1-4). This helps identifying actors and also relationships between them. Step 2, *Discovering Use Cases for the Actors*, proposes three guidelines (5-7). This helps identifying use cases. Goal dependencies will be mapped to use cases; task dependencies will be mapped to use cases if the corresponding tasks are decomposed into other sub-tasks; resource dependencies are mapped to use cases if there is a well justified reason. Softgoal dependencies are

not mapped into use cases, but instead left as internal constraints to several use cases. Finally, step 3, *Discovering and Describing Scenarios of Use Cases*, proposes three guidelines (8-10). This is accomplished by analyzing each i* actor and its relationships to extract information that can lead to the description of the use case scenarios. Therefore, internal elements which are used by the actor to achieve goals and softgoals, to perform tasks or obtain resources must be considered. Typically, the dependencies associated with the actor are satisfied internally through means-ends and task-decomposition relationships. These relationships must be analyzed to derive scenario steps for the use cases.

2.3 Identifying crosscutting use cases

Having identified and specified use cases, we are able to check whether there is a use case included by more than one use case, or if there is a use case that extends more than one use case. Both situations characterize a crosscutting (or aspectual) use case, because it crosscuts several use cases. This use case may be designed and implemented as an aspect (using AspectJ, for example). Aspectual use cases have been described in [1, 2, 3, 11, 12].

3. The Case Study

The case study we have chosen to illustrate our ideas is based on the Meeting Schedule system, where a new goal to announce the meeting date and location, and a restriction to impose confidentiality on the personal participant and meeting details has been added. The aim of the Meeting Schedule system is to support the organization of meetings. For each meeting request, the meeting scheduler should try to determine and broadcast a meeting date and location so that most of the intended participants will participate effectively. The system would find dates and locations that are as convenient as possible. The meeting initiator would ask potential participants for information about their availability to meet during a date range, based on their personal agendas. This includes an exclusion set of dates preferred by the participant for the meeting. The meeting scheduler comes up with a proposed date. The date to be chosen must be an available date, and should ideally belong to as many preference sets as possible. Participants would agree to a meeting date once an acceptable date has been found. Confidentiality for dates and location of the meeting, as well as the participants' data is required. When the meeting date is agreed, it should be announced to all interested parties. Fig. 2 illustrates the i* Strategic Dependency Model for this example.

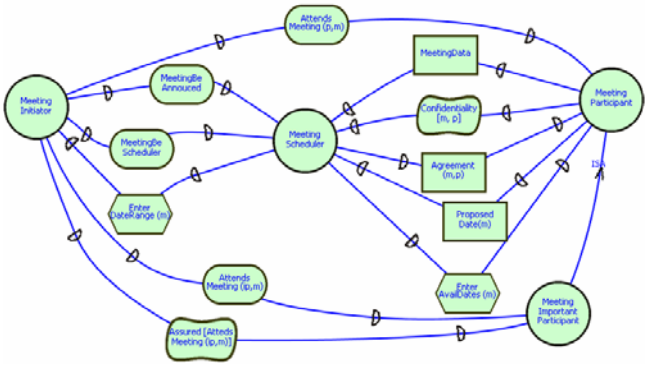

Figure 2. Meeting Schedule Strategic Dependency Model

The meeting initiator depends on participants to attend the meeting. S/he delegates much of the work of the meeting scheduling to the meeting scheduler. This determines what are the acceptable dates, given the availability information (*task dependency EnterAvailDates(m*)). The meeting initiator does not care how the scheduler does this, as longer as the acceptable dates are found. This is reflected in the goal dependency *MeetingBeScheduled* from the initiator to the scheduler. Also, it does not care how the scheduler does the meeting announcements. After finding a meeting date, it will broadcast the meeting date to all participants. The participant requires confidentiality of the meeting and personal information (softgoal dependency "*Confidentiality [m, p]*). On the other hand, to arrive at an agreeable date, participants depend on the meeting scheduler for date proposals (*resource dependency ProposedDate(m)*). Once proposed, the scheduler depends on the participants to indicate whether they agree with the date or not (*resource dependency Agreement(m,p)*). For important participants, the meeting initiator depends critically on their attendance, and thus also on their assurance that they will attend (*softgoal dependency Assured(AttendsMeeting(ip.m))*). The meeting scheduler depends on the meeting initiator to provide a date range (*task dependency EnterDateRange(m)*) for the scheduling.

Fig. 3 shows the strategic rationale model for the actor Meeting Scheduler. This model provides a more detailed level of modeling by looking "inside" the actor to model internal intentional relationships.

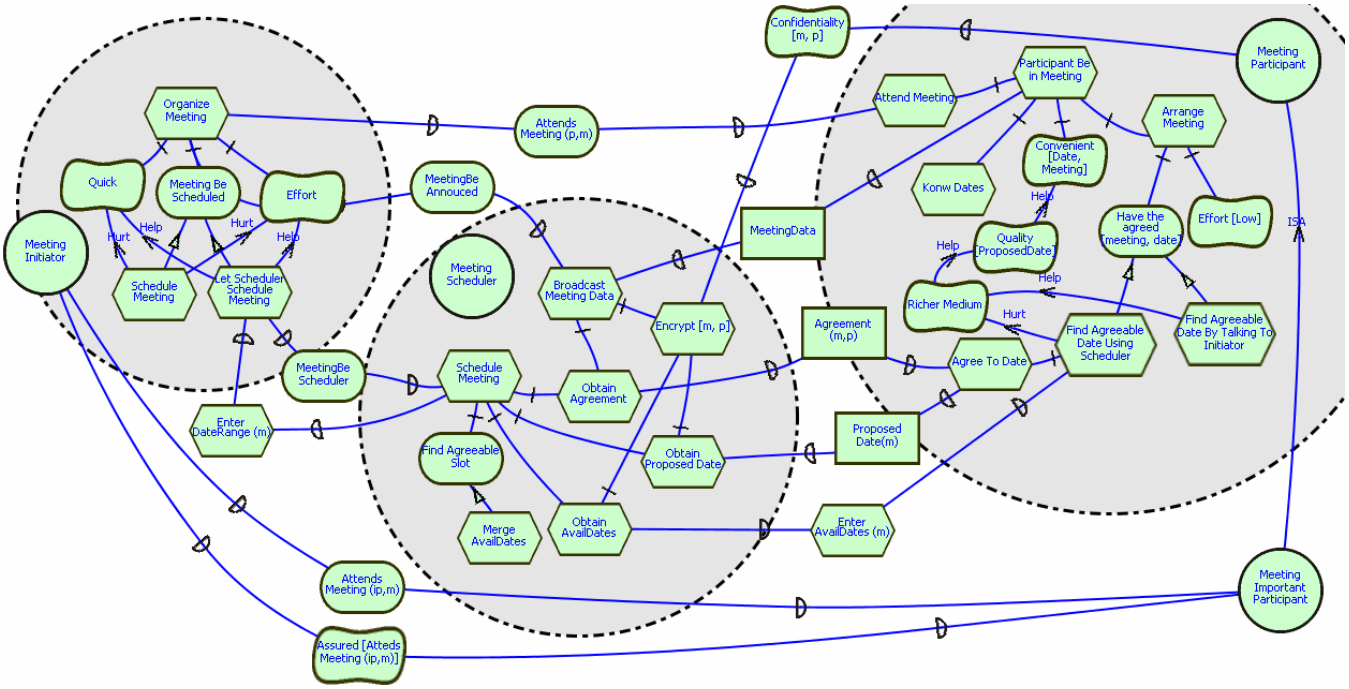

Figure 3. Meeting Scheduler Strategic Rationale Model

The *Meeting Scheduler* actor represents a software system that partially performs the meeting scheduling, while the *Meeting Initiator* and *Meeting Participant* are responsible for providing or receiving information to the system. The *Meeting Scheduler* actor possesses a *Schedule Meeting* task which is decomposed into three sub-components using the task-decomposition relationship: *Find Agreeable Slot, Obtain Agreement and Obtain Avail Dates.* These sub-components represent the work that will be accomplished by the meeting scheduler system. Similarly, the *Meeting Scheduler* uses a *Broadcast Meeting Data* task which is decomposed into two sub-components: *Obtain Agreement* and *Encrypt [m, p]*. These sub-components represent what is necessary to satisfy that task.

4. Identifying Crosscutting Concerns in I*

Our research approach has been carried out according to the steps depicted in Figure 4.

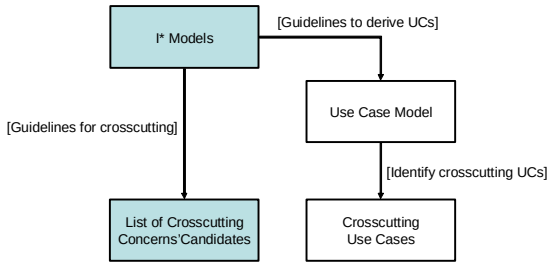

Figure 4. Research approach

Our goal is to propose a set of guidelines to help us identifying a list of crosscutting concerns' candidates directly from the information available on the i* models (left side of Fig. 4). The list of candidates obtained is validated by using existing ideas proposed by different authors. In particular, we know how to identify crosscutting use cases in a use case model [1, 2, 3, 5, 11, 12]. On the other hand, we know how to derive a use case model from the SD and SR models [18]. Therefore, the idea was to first derive a use case model and then identify crosscutting use cases (right hand side of Fig. 4).

4.1 Identifying Crosscutting Concerns

A crosscutting concern can bee seen as a model element that is required by, or affects, several other model elements modifying their original behaviors. In the context of i* a crosscutting concern is identified as follows:

Rule. *If a task that is related to an external dependency is required by (i.e. is a decomposition element) two or more tasks which are also related to other external dependencies, the first task is a candidate crosscutting concern.*

4.2 Case study

Applying our rule to the case study, a list of possible candidate crosscutting concerns can be identified. Let us consider the SD model (Fig.2) and SR model (Fig. 3) of the Meeting Scheduler in the i * approach. We start by selecting the tasks that are related to external dependencies (note that the direction or type of the dependency is not important). For each selected task, we list all its subtasks. For traceability reasons we propose the construction of a table recording all the model elements that satisfy those steps (see Table 1).

Table 1. Tracing model elements to identify crosscutting concerns

Tasks	External Dependencies	Task Decomposition	Has External Dependency
Schedule Meeting	Meeting Be Scheduled Enter Date Range(m)	Obtain Agreement	Agreement(m,p)
		Obtain Avail Date	Enter Avail Dates(m)
		Obtain Proposed Date	Proposed Date(m)
Encrypt[m,p]	Confidentiality[m,p]		
ObtainAgreement	Agreement(m,p)		
Broadcast Meeting Data	Meeting Be Announced Meeting Data	Encrypt[m,p]	Confidentiality[m,p]
		Obtain Agreement	Agreement(m,p)
Obtain Avail Dates	Enter Avail Dates(m)	Encrypt[m,p]	Confidentiality[m,p]
Obtain Proposed Date	Proposed Date(m)	Encrypt[m,p]	Confidentiality[m,p]

The crosscutting concerns are the subtasks (from the *Task Decomposition* column) which have an external dependency (*Has External Dependency* column) which appear in at least two cells of *Task Decomposition*. The resulting list of crosscutting concerns is: *ObtainAgreement* and *Encrypt [m, p]*.

5. Validating our Proposal

5.1 Deriving a Use Case Model from i*

Applying the guidelines summarized in Fig. 1, we start by discovering actors (guidelines 1 to 4). These are: *Meeting Initiator*, *Meeting Participant*, *Important Participant*. The goal of the following step is to discover use cases for the actors (guidelines 5 to 7). The resulting use cases are: *Meeting Be Announced*, *Meeting Be Scheduled*, *Proposed Date*, *Enter Date Range*, *Enter Avail Dates* and *Agreement*. Finally, step 3 is to derive the scenarios for the use cases (guidelines 8 to 10). This step is iterative and allows the identification of <<*include*>> and <<*extend*>> relationships between use cases. From the task-decomposition link of *Schedule Meeting* task we see that the use case *Meeting Be Scheduled* associated to this task will include the use cases *Agreement*, *Proposed Date* and *Enter Avail Dates*. For the task *Broadcast Meeting Date*, responsible for the satisfaction of the goal dependency *Meeting Be Announced*, we see that the use cases associated with this goal will include the use case *Agreement*. The application of these guidelines results in the use case model depicted in Fig. 6.

5.2 Identifying Crosscutting Concerns in a Use Case Model

As described in section 2.3, there is only one crosscutting use case (i.e., Agreement) as it is included by more than one use case (meeting be Announced and Meeting Be Scheduled).

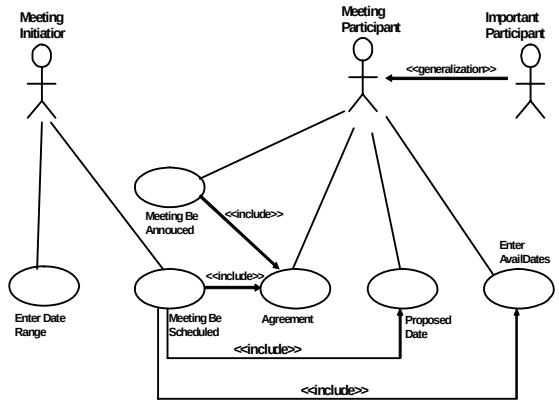

Figure 6. Use Case Diagram for the Meeting Schedule.

5.3 Discussion

The resulting list of crosscutting use cases and the initial list of crosscutting concerns is different. The reason is that the guidelines proposed in [18, 19] do not handle the softgoal-dependency influences. The authors see softgoals as restrictions that are internally applied to several model elements.

However, softgoals correspond, in general, to non-functional requirements. And, according to several authors [13, 14, 15, 16, 17] non-functional requirements are recognized as potential candidate aspects. Therefore, our proposal is to externalize this type of properties, by operationalizing them explicitly in the SR model. For example, to handle the *Confidentiality* softgoal, we created the task *Encrypt*. (Of course, other externalizations are possible). As a result, we identify *Encrypt* as a crosscutting concern.

By generalizing the guidelines proposed in [18, 19] to include the explicit treatment of softgoals, the final use case model is shown in Fig. 7.

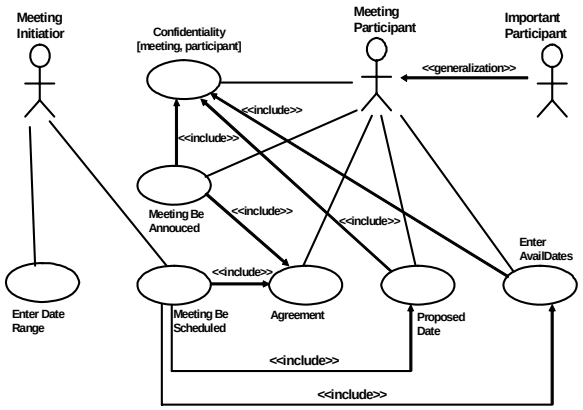

Figure 7. Use Case Diagram for the Meeting Schedule Extended

One limitation of our approach is that, currently, it is dependent on the way the developer models the problem, specially the SR model, where the candidate crosscutting concerns are identified. This means that the end result depends on the model provided by the developer.

6. Related Work

In [23] the RE model chosen is the KAOS framework [9]. The authors claim that satisfying OR-decomposed subgoals in the KAOS model typically leads to tangled implementations, and agents responsible for multiple OR-refined goals should be implemented in the aspect-oriented manner. Instead of proposing requirement engineering (RE) treatments directly mapped to aspects, they examined existing well-established RE models and identified patterns in these models that could be better designed and implemented using aspect-oriented programming. They started from the code level and, through the technique of aspect mining, identified application features existing in a crosscutting fashion. In next step, they performed a consolidated modeling of the application requirements in terms of KAOS concepts. Then they compared the goal decomposition and the actual code decomposition, trying to identify the connection patterns in the goal decomposition graph which would give rise to aspects.

[22] shows that aspects can be discovered during goal-oriented requirements analysis. Their proposal includes a systematic process for discovering aspects from relationships between functional and non-functional goals. The process presents a systematic way for the refinement of a V-graph. During each step of the process, the goal analysis tool is used to detect conflicts and deteriorations. The process ends when all the root goals and softgoals are satisfied. At this stage, they are able to identify candidate aspects by identifying tasks that have a high fan-in. The resulting graph can be further refined if candidate aspects are grouped into what they called goal aspects.

Our approach is to identify crosscutting concerns directly from the organization models, not having to use auxiliary RE techniques to accomplish our goal. The auxiliary RE techniques used here (use cases) served for validation purposes only.

7. Conclusions and Future Work

This work reports an initial investigation on how early aspects modeling can be supported by the i* framework. Firstly, we have proposed a general rule to help identifying a list of candidate crosscutting concerns directly from the information available on the i* models. The proposed approach has been validated using other techniques. A use case model was derived from i* models using [18, 19] and the candidate crosscutting concerns were then identified using [13, 14, 15, 16, 17].

The work was illustrated with an adapted meeting schedule example. Our proposal proved to be more efficient as less effort was spent to identify the candidate aspects: the identification was realized directly from i* models and more candidate aspects were identified (related to the softgoals).

Future work we plan to validate further the ideas discussed here by using larger case studies.

ACKNOWLEDGEMENTS

This work was supported by several research grants (CNPq Proc. 304982/2002-4, CAPES Proc. BEX 1775/2005-7, Proc. BEX 3003/05-1, Proc. BEX 3014/05-3 & CAPES/ GRICES Proc. 129/05).

REFERENCES

[1] Araújo, J., Coutinho, P., "Identifying Aspectual Use Cases Using a Viewpoint-Oriented Requirements Method", Early Aspects 2003: Aspect-Oriented Requirements Engineering and Architecture Design, Workshop of the 2nd International Conference on Aspect-Oriented Software Development, Boston, USA, 17 March 2003.

[2] Araújo, J., Moreira, A., "An Aspectual Use Case Driven Approach", VIII Jornadas de Ingeniería de Software y Bases de Datos (JISBD), Alicante, Spain, 12-14 November 2003.

[3] Araújo, J., Moreira, A., Brito, I. and Rashid, A., "Aspect-Oriented Requirements with UML". Workshop on "Aspect-oriented Modeling with UML", UML 2002, Dresden, Germany, October 2002.

[4] E. Baniassad, P. Clements, J. Araújo, A. Moreira, A. Rashid, B. Tekinerdogan, "Discovering Early Aspects", IEEE Software, Special Issue on Aspect-Oriented Programming, January/February 2006 (To appear)...

[5] Brito, I, Moreira, A., Integrating the NFR framework in a RE model. Early Aspects 2004: Aspect-Oriented Requirements Engineering and Architecture Design, workshop of the 3rd International Conference on Aspect-Oriented Software Development, Lancaster, UK, 22-26 March 2004.

[6] Castro, J. F. B., Alencar, F. M. R., Cysneiro Filho, G. A. A., Mylopoulos, J. "Integrating Organizational Requirements and Object Oriented Modeling**".** In: 5th IEEE International Symposium on Requirements Engineering, 2001, Canada, pp.146 – 153.

[7] Castro, J., Kolp, M. and Mylopoulos, J., "Towards Requirements-Driven Information Systems Engineering: The Tropos Project", Information Systems News, Elsevier, vol 27, pp. 365-89, 2002.

[8] Castro, J. F. B., Silva, C. T. L. L., Mylopoulos, J. "Modeling Organizational Architectural Styles In UML". In: The 15th Conference On Advanced Information Systems Engineering (Caise03), 2003, Klagenfurt/Velden. Lecture Notes in Computer Science- LNCS Vol 2681, 2003, pp.111 – 126.

[9] Dardenne, A., van Lamsweerde, A. and Fickas, S., “Goal-directed requirements acquisition”. Science of Computer Programming, Vol. 20, North Holland, pp. 3-50, 1993.

[10] Giorgini, P., Kolp, M., Mylopoulos, J. and Castro, J. Tropos: “A Requirements-Driven Methodology for Agent-Oriented Software”, Book Chapter in Agent-Oriented Methodologies. ed.: Idea Group, pp. 20-45, 2005.

[11] Jacobson, I., P. Ng, Aspect-Oriented Software Development with Use Cases, Addison-Wesley, 2005.

[12] Moreira, A, Araújo, J., Brito, I., "Crosscutting Quality Attributes for Requirements Engineering", Software Engineering and Knowledge Engineering Conference (SEKE), Ischia, Italy, 15-19 July 2002.

[13] Moreira, A., Araújo, J.. “Handling Unanticipated Requirements Change with Aspects”. Software Engineering and Knowledge Engineering Conference (SEKE'04), Banff, Canada, June 2004.

[14] Moreira, A., Araújo,J., Rashid, A. "A Concern-Oriented Requirements Engineering Model", 17th Conference on Advanced Information Systems Engineering (CAiSE 2005), Porto, Portugal, Lecture Notes in Computer Science, Springer, 13-17 June 2005.

[15] Moreira, A., Rashid, A., Araújo, J. "Multi-Dimensional Separation of Concerns in Requirements Engineering", 13th IEEE International Requirements Engineering Conference (RE 2004), Paris, France, IEEE Computer Society, 29 Augusto - 2 September, 2005.

[16] Rashid, A., Moreira, A., Araújo, J., "Modularisation and Composition of Aspectual Requirements", AOSD 2003, Boston, USA, ACM Press, 17-21 March, 2003.

[17] Rashid, A., Sawyer, P., Moreira, A., Araújo, J., "Early Aspects: a Model for Aspect-Oriented Requirements Engineering", Requirements Engineering 2002 (RE'02), Essen, Germany, 9-13 September 2002.

[18] Santander, V. F. A., “Integrating Organizational Modeling with Functional Modeling”(in portuguese: Integrando Modelagem Organizacional com Modelagem Funcional). Doctoral Thesis, Universidade Federal de Pernambuco, Recife, Brazil, December, 2002.

[19] Santander, V. F. A., Castro, J. F. B. “Deriving Use Cases from Organizational Modeling”. RE02 - IEEE Joint Conference on Requirements Engineering, Essen, Germany, 2002.

[20] Yu, E., “Modeling Strategic Relationships for Process Reengineering”. Ph.D. thesis, Department of Computer Science, University of Toronto, Canada, 1995.

[21] Yu, E., “Agent Orientation as a Modeling Paradigm”. Wirtschaftsinformatik. 43(2), April- pp.123-132, 2001.

[22] Yu, Y., Leite, J. C. S. P.and Mylopoulos, J. “From goals to aspects: Discovering aspects from requirements goal models”. In RE ’04: Proceedings of the Requirements Engineering Conference, 12th IEEE International (RE’04), IEEE Computer Society, Kyoto, Japan, September 2004, pp 38–47.

[23] Zhang, C., Jacobsen, H. and Yu, Y.. "Linking Goals to Aspect”. In RE’05: Proceedings of the Requirements Engineering Conference, 13th IEEE International. (RE'05). IEEE Computer Society, Paris, France, September 2005.