i* with Aspects: Evaluating Understandability

Ricardo A. Ramos[1], Fernanda Alencar[1], João Araújo[2], Ana Moreira[2],
Jaelson Castro[1] and Rosangela Penteado[3]
[1] *Universidade Federal de Pernambuco - Brasil,*
{rar2, jbc}@cin.ufpe.br, fmra@ufpe.br
[2] *CITI/FCT, Universidade Nova de Lisboa - Portugal,*
{amm,ja}@di.fct.unl.pt
[3] *Universidade Federal de São Carlos - Brasil.*
rosangel@dc.ufscar.br

Abstract

The applicability of the i approach for organizational modeling has been compromised by the complexity of the resulting models. To solve this problem an approach that use i* with Aspect Orientation is proposed. In this work we evaluate the understandability of five selected concerns of two requirements documents. The first requirements document was modeled by the i* approach and the other one by i* extended with aspects. To do this evaluation it was used abstract metrics that needed to be instantiated in both requirement documents.*

1. Introduction

With the increase use of the multi-agent system (MAS) in last few years, the need for applicable and broadly accepted development methodologies has resulted in a large number of proposals such as: GAIA [19], MaSE [18], AALAADIM [7] and TROPOS [5]. The TROPOS adopts concepts offered by the i* (i-star) organizational modeling framework [20, 21] to provide a framework for understanding the organizational environment and goals. The participants of the organizational setting are actors with intentional properties such as goals, beliefs, abilities and commitments.

The goal models, like i*, may increase the complexity when they deal with the large multi-agent systems. This complexity compromises the understandability of these models and consequently makes difficult the following phases of the software development. In order, to improve the understandability of the goal models, Silva and Leite [16] present a method for integrating crosscutting concerns in the requirements engineering process. This approach uses the V-Graph goal model [22] and the concepts are defined in aspect-oriented languages to provide separation, composition and visualization of crosscutting concerns.

[1, 2, 3] investigate if the concepts of early aspects [14] can help dealing with the complexity which may arise when using i* and improve the understandability. The basic idea is (i) to identify and modularize crosscutting concerns (model elements that affect several other elements in the same model), and (ii) to define composition rules for these crosscutting elements (or aspectual elements) that allow us to recover the original model.

When new approaches or methods are proposed, it is necessary to evaluate them, and this evaluation can focus different quality attributes, such as understandability, modularization, and etc. This evaluation will encourage the software engineers to use them with some support about its quality [9].

In this paper we evaluate the understandability of i* models. For that we select five concerns of two excerpts of requirements documents. The first requirements document was modeled by the i* approach [21] and the other one by i* with aspects [1, 2, 3]. These evaluations use an instantiation, for each requirements document, of the same abstract metrics described in [12]. An analysis of the results is presented.

The rest of the paper is organized as follows. Section 2 presents some background that has been used for this work. In Section 3 we present the two evaluations. Section 4 discusses the data analysis. In Section 5 some related works are described. Finally, Section 6 presents our conclusions.

2. Background

2.1 The i* Framework

The purpose of the i* framework [20, 21] is to make easier the earlier comprehension of the intended system in the development process of agent oriented software emphasizing social factors. The social feature of i* is represented through actors and relationships between them. This approach provides a graphical view of both system's actors (agents, roles and positions) and their intentions, dependencies, responsibilities and vulnerabilities. i* proposes developing two models: the Strategic Dependency (SD) model and the Strategic Rationale (SR) model.

Strategic Dependency Model. The SD model consists of a set of nodes and links connecting them, where nodes represent actors and each link indicates a dependency between two actors. The depending actor is called depender, and the actor who is depended upon is called the dependee. Hence, an SD model consists of a network of actors, capturing the motivation and the rationale of activities. i* distinguishes four types of dependencies. Three of these are related to existing intentions – goal dependency, resource dependency and task dependency, while the fourth is associated with the notion of non-functional requirements, the so called softgoal dependency. In i* we can also model different degrees of dependency commitment on the part of the relevant actors (e.g. open, committed, or critical). We can also classify actors into agents, roles and positions. An agent is an actor with concrete physical manifestations (a person or a system). A role is an abstract characterization of the behavior of a social actor within some specialized context, domain or endeavor. A position is a set of roles typically played by one agent. Finally, i* supports the analysis of opportunities and vulnerabilities for different actors [21].

Strategic Rational Model. The SR model provides a more detailed level of modeling by looking "inside" actors to model internal intentional relationships. It is used to: (i) describe the interests, concerns and motivations of participant process; (ii) enable the assessment of the possible alternatives in the definition of the process; and (iii) provide the rationale behind the dependencies between the various actors. Nodes and links also compose this model. It includes the previous four types of nodes: goal, task, resource and soft-goal. However, three new types of relationship are incorporated: (i) means-end that suggests that there could be other means of achieving the objective (alternatives), (ii) task-decomposition that describes what should be done in order to perform a certain task and (iii) the means-end contributing for softgoals links that will represent a partial contributions of a means (task or softgoal) to an end (softgoal).

In this paper we selected the requirements document excerpts of the Meeting Schedule system [21] modeled in the SR level; in the section 2.3 more details will be given.

2.2 Integration of Aspects with i* Models

By addressing crosscutting concerns earlier on agent-oriented software development, we will avoid tangled and scattered software artefacts. These crosscutting concerns are responsible for producing tangled representations that are difficult to understand and maintain. The identification and specification of crosscutting concerns in early phases will result in better support for modularization. In [1, 2, 3] it was introduced an approach to support early aspects identification in agent-oriented software development. In particular, it is proposed a set of guidelines to help the identification and representation of crosscutting concerns from the i* models. In fact, the SD and SR models were extended to represent aspectual concepts and have its graphical complexity reduced.

The general proposed approach is illustrated in Figure 1. This approach takes as inputs the i* models and consists of four steps: (1) identification and representation of candidates aspects; (2) identification of the relationship among candidate aspects; (3) composition; and (4) trade-offs analysis, which can initiate a new iteration. The first and second phases are done in parallel way.

In the first step three guidelines are proposed to identify candidate aspects. These three guidelines are as follows:

Guideline 1 (in the SD model): if the same dependum is provided by at least two dependee actors, them the operationalization corresponding to that dependum is a candidate crosscutting concern.

The Meeting Initiator actor (Figure 4) depends to the Meeting Participant (dependee actor) in a softgoal dependency Response Time. On the other hand, the Meeting Participant actor is related with the actor Meeting Scheduler (dependee actor) through the same dependency. The two distinct dependee actors operationalize the same dependum element (Response

Time), them its operationalization will be a candidate crosscutting concern.

Guideline 2 (in the SR model): if a task, that is directly or indirectly related to an external dependency, is required by (i.e. is a decomposition element of) two or more tasks (which are also related to other external dependencies), then that task is a candidate crosscutting concern. Observe that a task is indirectly related to an external dependency if, in the hierarchy where it belongs, at least one of its parents is connected to an external dependency.

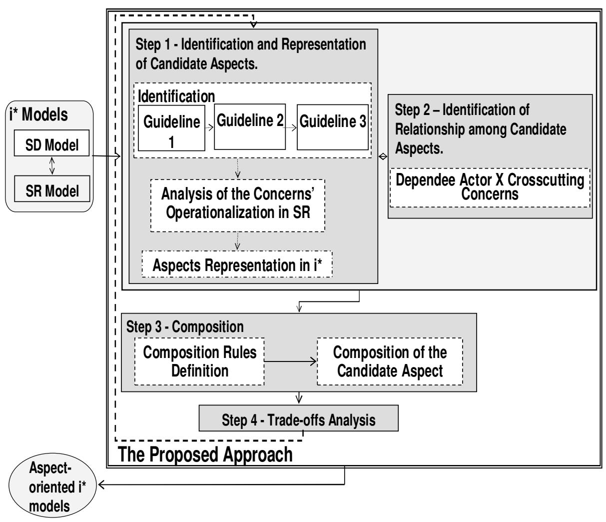

Figure 1 - The proposed approach to i* models with aspects [3].

For example (Figure 4), the task Obtain Proposed Date is simultaneously required by the task Broadcast Meeting Data and Schedule Meeting. Then this task is a candidate crosscutting concern.

Guideline 3 (remove redundancies): the list of crosscutting concerns identified by Guideline 2, which correspond to operationalizations of crosscutting concerns identified by Guideline 1, need to be merged together. The final crosscutting concerns are those that correspond to the operationalizations.

According to guideline 2, the task Handle Response Time (Figure 4), that operationalizes the softgoal dependum Response Time, will be a candidate since this task is simultaneously required by the task Schedule Meeting and Announce Meeting. But this same task was captured by the guideline 1.

The final crosscutting concerns are, in fact, particular kinds of tasks. For modularization purposes and following the principles of AOSD we should externalize and modularize these tasks, taking them away from the original actors, and place each of them in a new kind of model element, the aspect. Initially the dependency between the depender and the dependee actors involved with the crosscuting concern is replaced by an aspect that modularizes the crosscutting concern, A "Crosscuts" relationship between the aspect and the dependee it affects is defined. The direction indicated by the triangles suggests the composition direction, meaning that the behvaiour of the source element needs to be transferred to the behaviour of the target elements. The relationship between the depender and the dependum is specified by the compositions rules. The candidate aspect is represented by a star (see Response Time in Figure 2). If this candidate aspect will be, really, an aspect in the late stages of development, it will be attributed to some agent that has the capability to implement this.

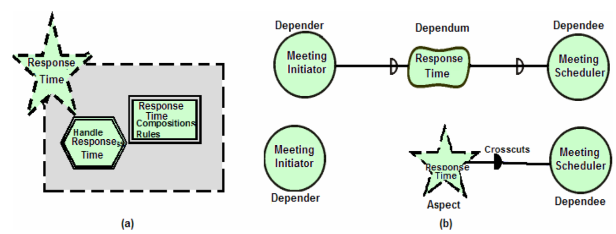

Figure 2 - An aspect element**.**

2.3. The Meeting Schedule System

The system we have chosen to evaluate is the Meeting Schedule system [21]. 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 finds dates and locations that are as convenient as possible. The meeting initiator asks potential participants for information about their availability to meet during a date range, based on their personal agendas, as well as, an exclusion set of dates. 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.

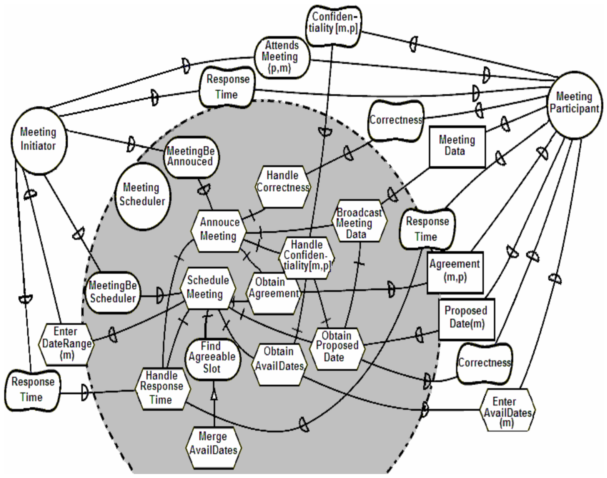

Figure 3 - The Meeting Schedule system (partial) SR model.

The Figure 3 shows the strategic rationale model for the Meeting Scheduler actor. This model provides detailed level of modeling looking "inside" the actor to model internal intentional relationships. In the model level, task-decomposition (like Schedule Meeting that is decomposed in a sub-goal, *Find Agreeable Slot*, and three sub-tasks: *Handle Response Time, Obtain AvailDates, Obtain Agreement*) provides a hierarchical description on intentional elements. Each element or sub-element in a task is needed for the success of this task. The means-end link in the SR model provides understanding about why an actor would engage in some tasks, pursue a goal, need a resource or want a softgoal. In this example, we can see a means-end link between the *Find Agreeable Slot* goal (the end) and the *Merge Avail Dates* task (the means through which the end is reached).

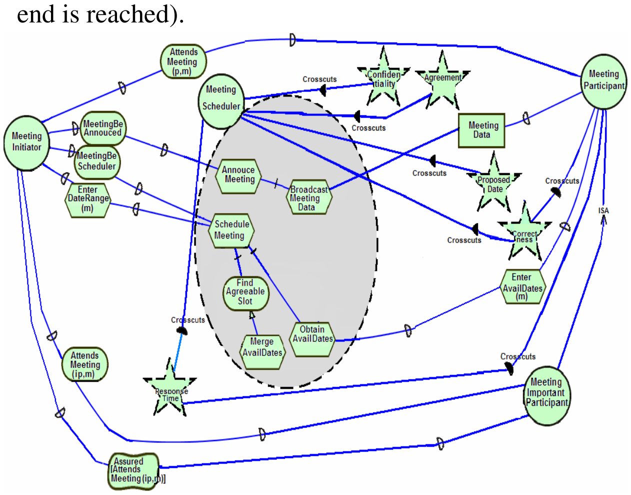

Figure 4 - The Meeting Schedule system aspectual SR model (partial) [1]

The Figure 4 shows a version using the approach of Alencar [1, 2, 3] of the same strategic rationale model for the Meeting Scheduler actor showed in Figure 4. In the aspect oriented version all concerns that was scattered and appear repeatedly were modularized to simplify the original representation. Aspects are represented by stars. The crosscutting relationships between each aspect and other model elements are shown as arcs, with a dark semi-circle. The direction indicated by the semi-circle suggests the direction of the composition, meaning that the source element's behaviour needs to be "injected" into the target elements' behaviours.

2.4 The Abstract Approach to Measure Requirement Documents

This approach describes an instantiation process to abstract metrics (Figure 5) for requirements documents. A set of metrics which, together with some guidelines, can be instantiated to measure requirements documents, obtained using any approach[12].

The guidelines in Figure 5 part 'a' are used to assist the identification of structures and substructures used by an approach to specify and/or model a requirements document, for example, in i* models: goal, task, resource, softgoal. These structures are called artifacts. The guidelines help the software engineer describing the general characteristics of the artifacts.

In the first stage, Figure 5a, the software engineer uses the guidelines to find artifacts in the requirements document. When the artifact is found, the software engineer will have to fill a template named "artifact description". Each artifact has to be described in details. This have the objective to delimit the artifact scope, this action will be useful to take decisions in case it has some conflict.

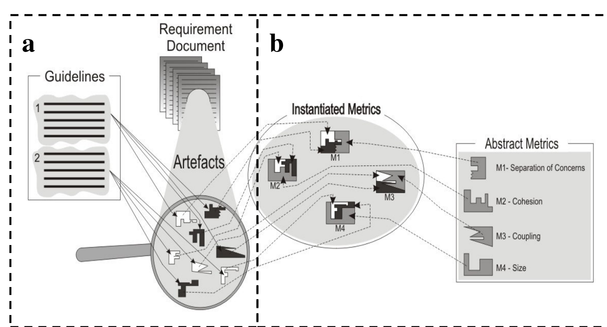

Figure 5 – Overview of Metrics Instantiation Process [12].

In the next stage, Figure 5b, the abstract metrics must be instantiate using the template “artifact description” (filled in the first phase). The Table 1 shows partially two abstract metrics sets and their descriptions, these two metrics will be used in our evaluation. Currently the abstract metrics template has 4 sets of metrics: **M1** - Separation of Concerns (Concern Diffusion over Components and Concern Diffusion over Elements), **M2** – Cohesion (Cohesion between Components), **M3** – Coupling (Coupling between Components), **M4** - Size (Vocabulary Size, Number of Elements and Number of Conditional Elements) [12].

Table 1 - Metrics template (partially).

- All variables (words between the symbols ‘<’ and ‘>’) of this template must be instantiated with the use of the filled data in the “artifact identified template”. - Words between the symbols ‘[’ and ‘]’ means that only one must be selected. * Before the instantiation, the software engineer must decide “which the specific concern that will be measure”.			
M 1	**Separation of Concerns Metrics**		
Concern Diffusion over Components (CDC) - is a requirement metric that counts the number of primary components (main decomposition structures) whose main purpose is to contribute to the speciation of a specific concern.			
How many	**<Artifact>**	whose main purpose is to contribute to the specification of a	***<Concern (specific)>**
M 4	**Size Metrics**		
Vocabulary Size (VS). VS measures the requirement vocabulary size. This metric counts the number of main decomposition structure that exist in the requirements document.			
How many	**<Artifact>**	exist in the **[selected part, full requirement document]**	

3 The Evaluation

The main goal of this study is evaluate the understandability of five selected concerns in two excerpts of requirements documents. The first is modeled by the original i* model [20] and another one with the aspect oriented i* approach [1].

The abstract approach to measure requirement documents [12] was used to conduct this evaluation. The evaluation was divided in three steps: the evaluation planning, the metrics instantiation and the perform metrics collection.

i) The evaluation planning:

i.1) what it will be measured - we decide to measure only the excerpt of the requirements document, modeled in SR level, which was available at [1]. In this selected excerpt the concerns in focus are: confidentiality, response time, correctness, agreement and proposed date. The internal structures and the compositions rules are not evaluated.

i.2) how it will be measured - we are starting with the principle that the number of a structure can indicate the effort needed to understand the system [6]. Thus, in this evaluation, we assume that the system with fewer structures can be easier to understand than a system that has many structures. To measure the number of structures we select the Vocabulary Size metric from the metrics template (Table 1) [12].

At the requirements level, the understanding of a concern is directly associated with its modularity and the number of parts that compose this concern [17]. This is another principle that we are using in this evaluation. Thus, we state that a high number of structures to describe a concern could increase the complexity of it. To measure the number of structures that compose a concern we used the Concern Diffusion over Components metric from the metrics template (Table 1) [12].

i.3) the hypotheses - we define the two hypotheses:

H0 - The understandability quality of the five selected concerns was improved in the aspect oriented requirements documents, according to the principles defined in item i.2 .

Ha - The understandability quality of the five selected concerns was not improved in the aspect oriented requirements documents, according to the principles defined in item i.2 .

ii) The metrics instantiation:

ii.1) use the guidelines to find the measurable artifacts - we found four artifacts in the original i* approach, goal, softgoal, task, resource. In the aspect oriented i* approach we found five artifacts, the same four ones of the original document plus the aspect artifact.

ii.2) instantiate the metrics template - one metric instantiation was performed to each artifact found and to each concern in focus. For example, the Table 2 shows the five instantiated metrics to the Confidentiality concern in the aspect oriented i* approach.

Table 2 - Instantiated metrics to the Confidentiality comcern.

*Concern Diffusion over Components (CDC).-- Evaluation to the **Confidentiality** concern*			
How many	Softgoal	whose main purpose is to contribute to the specification of	Confidentiality
	Task		
	Goal		
	Resource		
	Aspect		

iii) The perform metrics collection:

iii.1) the metrics are manually collected and all results are summarized to each requirement document. For example, the Table 3 shows the metrics values for the original document and with aspect oriented requirement document version. In this table we can see that in the original requirements document there are 1 `softgoal` and 3 `tasks` that contribute to the *Confidentiality* concern specification. In the aspect oriented version only 1 `aspect` that contributes to this specification.

Table 3 - Metrics values to the Confidentiality concern.

Metric Values		*Concern Diffusion over Components (CDC).*		
Or.	**Asp.**	*Evaluation to the **Confidentiality** concern*		
1	0	`Softgoal`	whose main purpose is to contribute to the specification of	*Confidentiality*
3	0	`Task`		
0	0	`Goal`		
0	0	`Resource`		
-	1	`Aspect`		

4 Data Analysis

In the aspect oriented version, all structures that contribute to describe a concern, in the original version, were encapsulated in an aspect. Consequently, all the resulting metric values of the aspect oriented version are '0', with exception of the aspect structures that are '1'. The Figure 6 shows the metrics values for each one of five concerns in focus. All selected concerns are more modular in the version with aspects than in the original version.

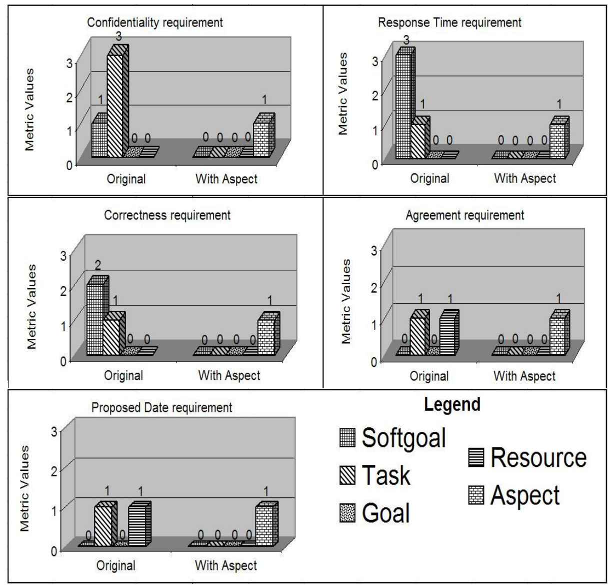

Figure 6 - Metric values to five concerns in focus.

In the Figure 7 we can see that the total number of structures decreases. For example, in the original requirements document (without aspects) there are 12 Task structures and in the version with aspects there are 6 -, a 50% reduce. The two metrics set indicate that the understandability was improved.

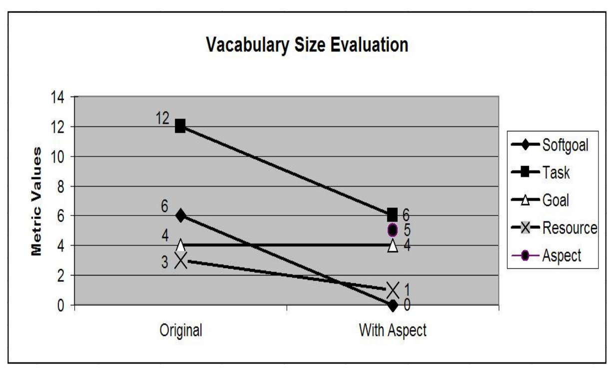

Figure 7 - The vocabulary size evaluation**.**

Using the insights gained during the data analysis we infer that the understandability of five selected concenrs was improved. Thus, we consider that while the hypothesis H0 is supported by the results shown in this section, the hypothesis Ha not be supported, accepted nor refuted.

It is important to note that the work described in this paper has the following limitations: i) the perform metrics did not evaluate the composition mechanisms nor the internally structures, ii) only a selected part of the requirements document was evaluated, iii) it was not considered the time to learn a new paradigm such as the aspect orientation.

5 Related Work

Hannemann and Kiczales evaluate Java implementations and AspectJ implementations of the GoF design patterns in terms of composability and pluggability [8]. Sant'Anna et al propose an assessment framework to evaluate aspect oriented code. In their work, it was performed a case study to evaluate the reuse and the maintenance quality of the GoF design patterns implementation in two versions, aspect and object oriented [15]. Both works performed the evaluation in the code level, in our work the evaluation was performed at the requirements level.

Anda et al [4] describe an explorative study where three different sets of guidelines were used for constructing and documenting use case models. A use case understandability evaluation was performed to indicate what the better guidelines are. In our work the understandability evaluation was performed in two versions of the same requirements document system produce by i* and aspect oriented i* approaches.

In [11] we proposed some additional metrics to extend an evaluation methodology that measure web based requirements document. In this work a case study was performed to evaluate the completeness in a Web based aspect oriented requirement document. In our work the evaluation is focused on five pre-selected concerns and the main objective is to measure the understandability using an abstract approach that measures requirements document modeled by any approach.

6 Conclusion

This paper shows an understandability evaluation of five selected concerns of two requirements documents. The first was modeled using the i* approach and the second one was specified using the aspect-oriented i* approach. This evaluation helps us to infer that the aspect-oriented i* approach could improve the modularity and decrease the total number of requirements structures. The identification of candidate aspects at early stages helps us to reduce the complexity of the i* models, promoting understandability, and to modularize the concerns that are scattered and tangled in the system specification.

Moreover, it was evaluated the applicability of the abstract approach to measure requirements document. This work is part of a project whose objective is to evaluate the requirements document quality, finding pieces that can be improved with the application of requirements patterns and refactoring [11, 12, 13].

With the lessons learned in this work we expected to improve the aspect oriented i* approach and the abstract approach to measure the requirements documents. Also, tool support for the abstract metrics approach will be developed. Both works are being developed by the Requirement Engineering Laboratory - LER [10].

Acknowledgments. This work was supported by several research grants (CNPq Proc. 304982/2002-4, Proc. 142248/2004-5, CAPES Proc. BEX 1775/2005-7, Proc. BEX 3014/05-3 & CAPES/ GRICES Proc. 129/05).

References

1. Alencar, F., Moreira, A., Araújo, J, Castro, J., Silva, C.,Mylopoulos, J.: Towards an Approach to Integrate i* with Aspects. In: Proc. of 8th International Bi-Conference Workshop on Agent-Oriented Information Systems (AOIS-2006), in conj. with CAiSE'06. Luxembourg, June (2006).
2. Alencar, F., Silva, C., Moreira, A., Araújo, J., and Castro, J.: Identifying Candidate Aspects with I-star Approach. In: Work. on Early Aspects:, Bonn, Germany, March 2006
3. Alencar, F., Moreira, A., Araújo, J., Castro, J., Ramos, R., and Silva, C.: Proposal to deal with the Complexity of i* Models with Aspects. In: the First International Conference on Research Challenges on Information Science – RCIS'07. Ouarzazate, Morocco, April, 2007 (to appear).
4. Anda, B., Sjoberg, D. I. K., Jorgensen, M. Quality and Understandability of Use Case Models, In: Proc of the 15th European Conference on Object-Oriented Programming, p.402-428, June 18-22, (2001).
5. Castro, J., Kolp, M. and Mylopoulos, J.: "Towards Requirements Driven Information Systems Engineering: The Tropos Project". In Information Systems, Vol. 27. Elsevier, Amsterdam, The Netherlands 365–389 (2002).
6. Fenton, N.E., Pfleeger, S.L.: Software Metrics: A Rigorous and Practical Approach. PWS Publishing Company (1997)
7. Ferber, J. and Gutknecht O.: "A meta-model for the analysis and design of organizations in multiagents systems". In Demazeau, Y., editor, ICMAS' 98, pages 128–135, Paris (1998).

8. Hannemann, J., Kiczales, G. "Design Pattern Implementation in Java and AspectJ", Proceedings of OOPSLA'02 , November, pp. 161-173(2002).
9. Kitchenham, B., Pickard, L., Pfleeger, A. L. Case Studies for Method and Tool Evaluation. In: IEEE Software, Vol. 12 Nr. 45 (1995).
10. LER – Laboratório de Engenharia de Requisitos, www.cin.ufpe.br/~ler, link visited in January (2007).
11. Ramos, R. A, Castro, J. F. B. "Avaliação de uma Metodologia de Medição da Qualidade em um Documento de Requisitos Orientado a Aspectos". In: WER, Porto- Portugal, 2005.
12. Ramos, R. A., Araújo, J., Castro, J. F. B., Moreira, A., Alencar, F., Silva, C. "Uma Abordagem de Instanciação de Métricas para Medir Documentos de Requisitos Orientados a Aspectos". In: III Workshop Brasileiro de Desenvolvimento de Software Orientado a Aspectos WASP´2006. Florianópolis, Santa Catarina - Brasil (2006)
13. Ramos, R. A., Araújo, J., Castro, J. F. B., Moreira, A., Alencar, F., Silva, C. "Um Modelo de Qualidade para Avaliar Documentos de Requisitos Orientados a Aspectos". In: Desarrollo de Software Orientado a Aspectos, DSOA 2006, Asociado a XV Jornadas de Ingeniería del Software y Bases de Datos. Sitges - Barcelona (2006).
14. Rashid, A., Moreira, A., Araújo, J.: Modularization and Composition of Aspectual Requirements. In: Proceedings of the International Conference on Aspect-Oriented Software Development (AOSD'03), USA (2003).
15. Sant'Anna, C., Garcia, A., Chavez, C., Lucena, C., and von Staa, A. On the Reuse and Maintenance of Aspect-Oriented Software: An Assessment Framework. In XVII Brazilian Symposium on Software Engineering, (2003).
16. Silva, L., Leite; J. "Integração de Características Transversais Durante a Modelagem de Requisitos". In: Proceedings of 20º Simpósio Brasileiro de Engenharia de Software,SBES'05. October, Uberlândia-MG, Brasil (2005).
17. Sommerville, I.: Software Engineering, 7th edition. Pearson Education (2004).
18. Wood, M.F. and DeLoach, S.A.: "An overview of the multiagent systems engineering methodology". First international workshop, AOSE 2000 on Agent-oriented software engineering, p.207-221, January, Limerick, Ireland (2001).
19. Wooldridge, M. and Ciancarini, P.: "Agent-Oriented Software Engineering: The State of the Art". In Agent-Oriented Software Engineering. Springer-Verlag Lecture Notes in AI Volume 1957 (2001).
20. Yu, E.: "An organization modeling framework for information systems requirements engineering". In Proc. Of the 3rd Workshop on Information Technologies and Systems WITS'93, Orlando FL, December 4-5 (1993).
21. Yu, E.: Modeling Strategic Relationships for Process Reengineering. Ph.D. thesis, Department of Computer Science, University of Toronto, Canada (1995).
22. Yu, Y., Leite, J., Mylopoulos, J. From goals to aspects: discovering aspects from requirements goal models; Proceedings of Requirements Engineering Conference, 12th IEEE International (RE'04); pp. 38-47; Japan (2004).