

Concern Interactions and Tradeoffs: Preparing Requirements to Architecture

José Magno¹ and Ana Moreira²

¹ Dep. Eng. Informática, Escola Superior de Tecnologia e Gestão
Instituto Politécnico de Leiria, Morro do Lena, 2411-901 Leiria, Apartado 4163, Portugal

magno@estg.ipleiria.pt

² CITI/Dep. Informática, Faculdade de Ciências e Tecnologia
Universidade Nova de Lisboa, 2829-516 Caparica, Portugal

amm@di.fct.unl.pt

Abstract. This paper focus on concern interactions and tradeoffs in the context of an approach to build an aspect-oriented software architecture from an aspect-oriented requirements specification. This approach, the Requirements Engineering to Software Architecture Framework, is composed of several activities, where the first two address the problem domain and constitute the main body for this paper. These two activities will be described and exemplified by means of a case study.

1 Introduction

As Aspect-Oriented Software Development (AOSD) evolves and matures, its main concepts slowly travel up the software development lifecycle and new proposals for Requirements Engineering, Architecture Design and Detailed Design appear. This happened for other paradigms (e.g. structured and object-oriented), and currently is being repeated for AOSD. Several approaches have been proposed for Aspect-Oriented Requirements Engineering (AORE) [3, 4, 6, 10, 12] and for Aspect-Oriented Software Architecture (AOSA) [9, 11, 13]. As identified in [2], it becomes more and more important to present solutions to fill the gaps between AORE and AOSA. The Requirements Engineering to Software Architecture Framework (RESAF) is our first attempt to develop a framework to support the derivation of an aspect-oriented architecture design from an aspect-oriented requirements specification.

This paper discusses concern interactions and tradeoffs at the problem domain level, in the context of RESAF. This constitutes a minimal basis to identify main issues that will constrain the architecture design at the solution domain. The paper starts by explaining the first two activities of RESAF, in Section 2. Section 3, applies these two activities to a case study and Section 4 finalizes by drawing some conclusions and highlighting open issues that need to be addressed in the near future.

2 Analyse Requirements and Tradeoffs

The main steps of the two first activities, “Analyse Requirements” and “Analyse Tradeoffs for each Scenario”, of RESAF are illustrated in Fig. 1.

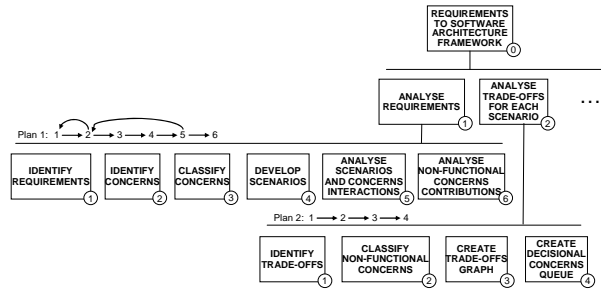


Fig. 1. RESAF activities “Analyse Requirements” and “Analyse Tradeoffs for each Scenario”.

Fig. 2 shows how those two activities interact, and, for each one, its main behavioural components. These activities are explained in detail in the following two sub-sections.

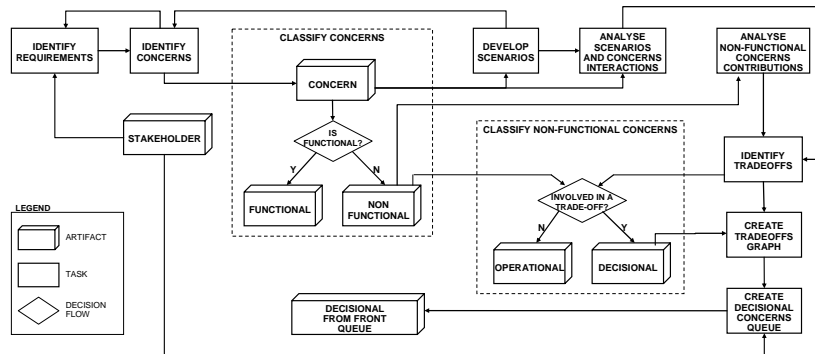


Fig. 2. “Analyse Requirements” and “Analyse Tradeoffs for each Scenario” activities’ interactions.

2.1 Analyse Requirements

The activity “Analyse Requirements” starts by identifying the problem requirements and structuring those requirements into concerns (identifying concerns). These concerns are classified into functional and non-functional. The next step is to develop scenarios from the stakeholders’ perspective [8] and analyse how they interact with the concerns [3, 4, 12].

As illustrated in Plan 1 of Fig. 1, the process of developing scenarios, identifying concerns and their interactions with scenarios is iterative. The initial number of scenarios is small, and therefore each one might address complex functionalities. The

end result is that each scenario interacts with a small number of concerns at a higher level of abstraction. This is good to start with, but might lead to situations where most of the concerns cut across most scenarios. However, it may well be that when analysing these interactions at a lower level of granularity, the number of crosscutting relationships is in reality a lot smaller. The next step is therefore to refine both the scenarios and the concerns. Scenarios are not difficult to refine, but non-functional concerns might be. A good source of information is the Non-Functional Requirements (NFR) Framework [5]. This refinement process is continued until each scenario addresses a single functionality and each non-functional concern is divided into concrete concerns with available specific solutions.

Finally, the last task is to analyse contributions between non-functional concerns [5, 12] to determine the possible contributions (positive or negative) between the non-functional requirements specified, and to create the *negative contributions graph* (see example in Fig. 3.a). NFR catalogues, such as those offered by the NFR Framework [5], can help building this graph.

2.2 Analyse Tradeoffs for each Scenario

RESAF's second activity is applied to each scenario. The idea is to identify tradeoffs, classify non-functional concerns, create the tradeoffs graph and finally build a *decisional concerns queue* (see Plan 2 of Fig. 1).

As illustrate in Fig. 3.b the identification of tradeoffs is achieved by creating, for each scenario, a sub-graph of the *negative contributions graph* built during the activity "Analyse Requirements". Now, for each scenario, we are able to classify the non-functional concerns into *decisional* or *operational* as they appear, or not, in the negative contributions sub-graph, respectively. For instance, in the scenario S1 shown in Fig. 3.b the concerns C1, C2 and C4 are decisional. The operational concerns reinforce the operationalization behaviour of the scenario and can be achieved without any negotiation. In contrast, decisional concerns contribute negatively to one or more non-functional concerns in a scenario. This requires a compromise between the various decisional concerns. This compromise must be obtained through stakeholders' negotiations and will, in general, require prioritisation of the decisional concerns.

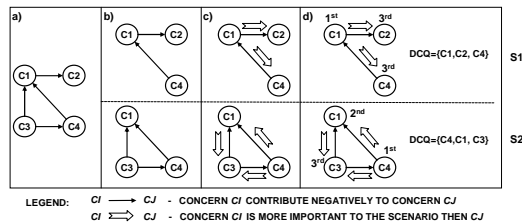


Fig. 3. Tradeoffs prioritization steps for scenarios S1 and S2 of a system represented in a).

A *tradeoffs graph* is created for each scenario by extending its negative contributions graph (see Fig. 3.c), with a new type of edge (\rightleftharpoons) connecting the *tradeoff concerns* (adjacent concerns linked by an edge in a negative contributions graph). The new type of edge represents the relative importance between two concerns in-

volved in a tradeoff. To create a *tradeoffs graph* it is necessary that the stakeholder establishes the relative importance between all decisional concerns in the negative contributions graph of its scenario.

With this new kind of edge, a *tradeoffs graph* can be seen as a *directed acyclic graph* (DAG) [7]. So, to create a *decisional concerns queue* (decisional concerns prioritized) from a *tradeoffs graph* (see Fig. 3.d), an adaptation of the usual algorithm of *topological sorting* [7] is proposed. The input of the *Topological Sorting Adaptation* is a tradeoffs graph *TOG* and the output is a decisional concern queue *DCQ*.

```

Topological Sorting Adaptation (TOG, DCQ)
DCQ ← {}
r ← 1
AQ ← {}
For each node n of TOG without incoming edges
  insert n into AQ
  insert (n, r) into DCQ
  associate r with the node n
while TOG has any edge not visited yet
  r ← r + 1
  if AQ is empty
    throw error exception
  remove a node n from AQ
  for each node m with an edge e not visited from n to m
    mark e has visited
    if m has no other incoming edges not visited
      insert m into AQ
      insert (m, r) into DCQ
      associate r with the node m

```

Table 4 shows a simulation of this algorithm applied to a scenario of our case study. The end result of this activity is a table with all the scenarios and concerns, where each cell indicates the type of concern with a ranking (as in Table 5). From here the order of composition of concerns in a given scenario can be derived. For each scenario *S*, with one functional concern *f*, the operational concerns o_1, o_2, \dots, o_m and the decisional concerns queue $\{(d_1, r_1), (d_2, r_2), \dots, (d_n, r_n)\}$, the order of composition is given by a rule of the type:

$$S = \mathcal{D}(\mathcal{F}(f) + \mathcal{O}(o_1, o_2, \dots, o_m), d_1, d_2, \dots, d_n)$$

meaning that *f* is the base concern over which all the operational concerns o_i can be applied; only then the decisional concerns can be handled according to their ranking. Notice, however, that composition order cannot always be derived automatically, as it might depend on the nature of the concerns. For example, no matter how low is the priority of authentication access in a system, it might need to be guaranteed before other more important concerns in that system, such as auditing or information retrieval.

Being able to separate the concerns that can be operationalized from those that are involved in tradeoffs is a first step towards supporting informed decisions before the architecture design is derived. Moreover, if we know the relative importance of the decisional concerns, we can select, from the various possible architectural choices, the combination that best satisfies the stakeholders' goals, or expectations.

3 Case Study

Let's apply the ideas described above to the Portuguese automatic toll collection system case study [1].

In a road traffic pricing system, drivers of authorized vehicles are charged at toll gates automatically. The gates are placed at special lanes called green lanes. A driver has to install a gizmo in his/her vehicle. The registration of vehicles includes the owner's personal data, bank account number and vehicle details. The gizmo is sent to the client to be activated using an ATM. The toll gate sensors read a gizmo. The information read is stored, and used to debit the respective account. When an authorized vehicle uses the system, a green light is turned on, and the amount being debited is displayed. If an unauthorised vehicle passes through it, a yellow light is turned on and a camera takes a photo of the plate number. There are three types of toll gates: single toll, where the same type of vehicles pay a fixed amount, entry toll to enter a motorway and exit toll to leave it.

The following list of use cases were obtained as a result of applying the activity "Analyse Requirements" to our example:

- *Register Vehicle*: registers vehicle and owners data. It includes the return payment information from the bank and the gizmo activation.
- *Pass Single Toll*: handles usages in a single point toll gate. If the vehicle is registered a light is turned green, the amount to be debited in the owner's account is displayed and the passage is stored. Otherwise, the light is turned yellow and the vehicle plate number is photographed.
- *Enter Motorway*: handles entrances in a motorway. If the vehicle is registered, a light is turned green and the entrance data is stored. Otherwise, the light is turned yellow and the vehicle plate number is photographed.
- *Exit Motorway*: handles exits of the motorway. If the vehicle is registered and entered correctly, a light is turned green, the amount to be debited is displayed and the passage is stored. Otherwise, the light is turned yellow and the vehicle plate number photographed.
- *Pay Monthly Bill*: bills the system users' monthly. Payments are achieved through bank transfers from the vehicle owner's account.

From this set of use cases, several scenarios can be derived. We will use the six scenarios listed in Table 1 for illustration purposes. By applying the iterative process of scenarios development to our initial scenarios, the concerns are identified and classified into functional and non-functional. For instance, "Exit Motorway Authorized" scenario requires the functional concerns: *Identification* (identify automatically the vehicle), *Billing* (calculate the amount to debit), *History* (record information needed when vehicle passes an exit motorway) and *Feedback* (show green light and the amount to debit). This scenario also requires concerns that can assure some quality criteria's to these functionalities. In fact, the "Exit Motorway Authorized" scenario requires the non-functional concerns: *Security* and *Multi Access* (for *Identification*, *Billing* and *History*), *Response Time* and *Availability* (for all other scenario's functionalities) and *Compatibility* (for *identification*).

As we can see in Table 1, the initial scenarios address a large number of functional concerns. Furthermore, the non-functional concerns cut cross almost all scenarios, and therefore need to be refined.

Table 1. Initial subset of scenarios and classified concerns interactions for the green lane system.

SCENARIO / CONCERNS		Registration	Billing	Identification	History	Feedback	Legal Issues	Payment	Validation	Response Time	Security	Availability	Multiple Access	Compatibility
S1	Register Vehicle	F		F	F	F		F	F		NF		NF	
S2	Pass Single Toll Authorized		F	F	F	F				NF	NF	NF	NF	NF
S3	Enter Motorway Authorized			F	F	F				NF	NF	NF	NF	NF
S4	Exit Motorway Authorized		F	F	F	F				NF	NF	NF	NF	NF
S5	Pass Toll Unauthorized			F	F	F	F			NF	NF	NF	NF	NF
S6	Pay Monthly Bill		F			F		F			NF			

LEGEND: F - FUNCTIONAL NF - NON-FUNCTIONAL

In order to exemplify the iterative process of scenarios development we choose to refine the scenarios responsible for handling a vehicle passage in a toll gate (scenarios S2 to S5 of Table 1). Table 2 describes the scenarios and classified concerns interactions after refinement of the scenarios. These refined scenarios were found by subdividing the functionalities of each of the original scenarios S2 to S5 of Table 1 and by refining the non-functional concern *Security* using the NFR Framework catalogue. The scenarios responsible for handling a vehicle passage in a toll gate were refined as follows:

- “Pass Single Toll Authorized” was refined into scenarios a, b, d and e;
- “Enter Motorway Authorized” was refined into scenarios a, b and d;
- “Exit Motorway Authorized” was refined into scenarios a, c, d and e;
- “Pass Toll Unauthorized” can be refined into scenarios a, f and g.

Table 2. Scenarios and concerns interactions after refining the scenarios S2 to S5 of Table 1.

SCENARIO / CONCERNS		C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18
		Registration	Billing	Identification	History	Feedback	Legal Issues	Payment	Web Registration	Validation	Precision	Response Time	Correctness	Availability	Integrity	Confidentiality	Authenticity	Multiple Access	Compatibility
a	Authorize Vehicle			F								NF		NF	NF	NF	NF	NF	NF
b	Record Vehicle Entry				F							NF		NF	NF				NF
c	Record Vehicle Exit				F							NF		NF	NF				NF
d	Show Green Light					F						NF		NF					
e	Show Amount to be Debited in Owner's Account		F		F	F						NF	NF	NF	NF				NF
f	Show Yellow light					F						NF		NF					
g	Take and Record Vehicle's Plate Photographs				F	F					NF	NF	NF	NF	NF	NF			

In the following iterations, scenarios e and g (of Table 2) need to be further refined, so that each refined scenario only handles one functional concern. Besides, we can see that for scenario “Authorize Vehicle”, for instance the non-functional concern *Security* was refined into the sub-concerns *Authenticity* of the vehicle’s gizmo, and *Integrity* and *Confidentiality* of the data transmitted between the gizmo and the toll gate.

Further iterations of the “Analyse Requirements” activity led us to identify 25 scenarios, 9 functional concerns and 9 non-functional concerns. These concerns are listed in Table 3.

Table 3. Concerns description for the green lane system.

ID	CONCERN	DESCRIPTION
C1	Registration	Users register their vehicle in the system
C2	Billing	System calculates the amount to be debited in the users account
C3	Identification	Vehicles must be automatically identified by the system
C4	History	Users' and banks' interactions with the system must be registered
C5	Feedback	System must give appropriated reply information to vehicle drivers
C6	Legal Issues	System takes unauthorized vehicle's plates photos and sends information to legal authorities
C7	Payment	System must provide automatic money transactions
C8	Web Registration	System provides a web interface for users' registration
C9	Validation	System must validate clients' information
C10	Precision	Photographic camera sensor should be accurate to identify the vehicles' plates
C11	Response Time	System should be able to handle many vehicles in a short time span
C12	Correctness	System should guarantee correct calculations
C13	Availability	System has to be on 24/7
C14	Integrity	System must ensure correct data transmissions
C15	Confidentiality	System must ensure that information is accessible only to those authorized to have access
C16	Authenticity	System must verify the identification of their users
C17	Multiple Access	System must handle a large number of users simultaneously
C18	Compatibility	System must ensure the ability to communicate with gizmos and banks

The last task of the activity “Analyse Requirements” is to identify contributions between non-functional concerns. From here, we can then build the negative contributions graph for our case study, as illustrated in Fig. 4.

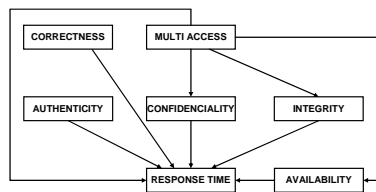


Fig. 4. Negative contributions graph for the green lane system.

We are able now to apply the activity “Analyse Tradeoffs for each Scenario”. Based on the negative contribution graph for green lane, sub-graphs are built, one for each scenario, to address each set of potential tradeoffs at a time. Fig. 5.a shows the negative contribution graph for the “Authorize Vehicle” scenario.

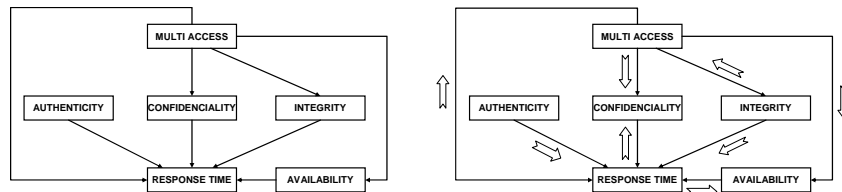


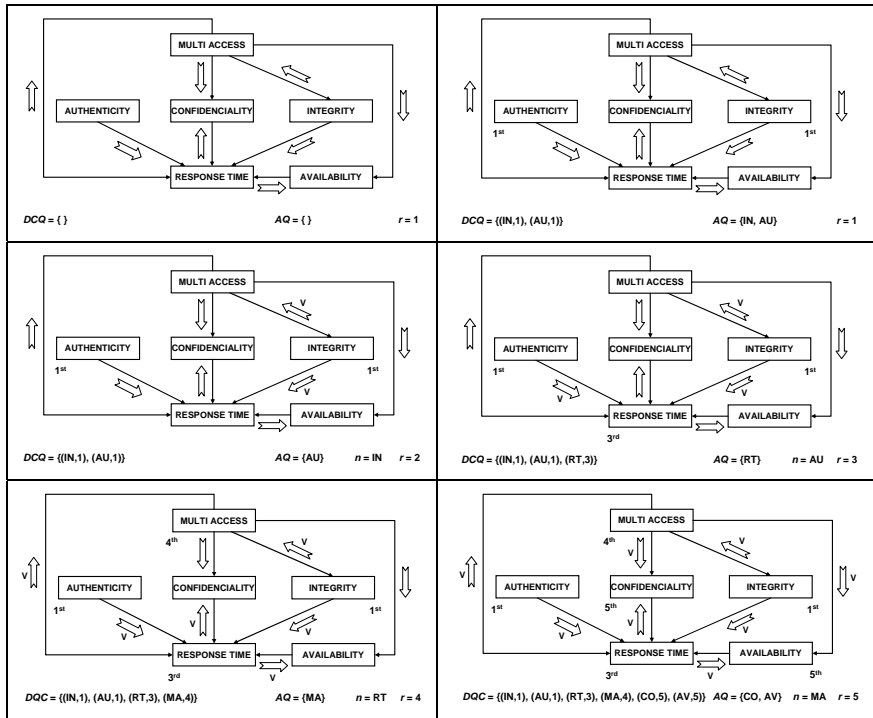
Fig. 5. a) “Authorize Vehicle” negative contributions graph; b) “Authorize Vehicle” tradeoffs graph.

This allows us to classify concerns, for each scenario, into decisional (if they are involved in a tradeoffs graph) or operational (if they are not). So, for instance, the

concerns *Authenticity*, *Multi Access*, *Confidentiality*, *Response Time*, *Integrity* and *Availability* are classified as decisional concerns for the “Authorize Vehicle” scenario, since they are nodes of the negative contribution graph.

Decisional concerns for each scenario can be now prioritized. The prioritization of decisional concerns is achieved by creating the scenario *tradeoffs graph* and applying it to the *Topological Sorting Adaptation*. Table 4 illustrates the resulting simulation by applying the tradeoffs graph of the “Authorize Vehicle” scenario to the algorithm, where the variables *DCQ* represents the “Decisional Concern Queue”, *r* represent the ranking counter, *AQ* represent the queue with the concerns with edges to be visited, and *n* represent the actual concern edges to be visited.

Table 4. *Topological Sorting Adaptation* simulation for scenario “Authorize Vehicle”.



For the scenario “Authorize Vehicle”, the concerns *Integrity* and *Authenticity* are the most important decisional concerns with ranking 1, followed by *Response Time* in 3rd, *Multi Access* in 4th, and finally *Availability* and *Confidentiality* in 5th position.

The application of the activity “Analyse Tradeoffs for each Scenario” to our example results in a table representing the interactions between scenarios and concerns (see Table 5), where decisional concerns are ranked. Notice that the ranking of the decisional concerns only depends on the relative importance that was given to each one by the stakeholders in each scenario.

Table 5. Scenarios and concerns interactions with prioritization of the decisional concerns.

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18
SCENARIO / CONCERNS	Registration	Billing	Identification	History	Feedback	Legal Issues	Payment	Web Registration	Validation	Precision	Response Time	Correctness	Availability	Integrity	Confidentiality	Authenticity	Multiple Access	Compatibility
S1			F								D3		D5	D1	D5	D1	D4	O
S2			F								D2		D4	D1			D3	
S3				F							D2		D4	D1			D3	
S4					F						D1		D2					
S5		F									D4	D1	D3				D1	
S6				F							D4		D3	D1			D2	
S7											D1		D2					
S8					F						D1		D2					
S9						F				O	D4	D1	D1	D1	D5			
S10				F										O				
S11		F										O		O				
S12							F					O		O	O			O
S13				F										O				
S14					F							O						
S15				F										O				
S16								F						D1	D1	O	D3	O
S17									F		O	D1					D2	
S18					F								D1				D2	
S19					F								D1				D2	
S20							F				O			D1	D1	O	D3	O
S21			F											D1	D1		D3	O
S22				F										D1	D1		D3	
S23	F										O				O			O
S24				F										O				
S25				F								O						

LEGEND: F - FUNCTIONAL O - OPERATIONAL D_r - DECISIONAL WITH RANKING r

Analysing the contents in Table 5, it is possible to extract the composition rules for each scenario. For example, for the scenario “Authorize Vehicle” we have:

$$S1 = \mathcal{D} (\mathcal{F} (\text{Identification}) + \mathcal{O} (\text{Compatibility}), \text{Integrity, Authenticity, Response Time, Multi Access, Availability, Confidentiality})$$

4 Related Work

Tradeoffs between conflicting concerns have been previously addressed in [3, 4, 12]. The innovation here is the systematic process to classify and rank concerns that contribute negatively to each other. The Aspectual Software Architecture Analysis Method (ASAAM) [14] introduces explicit mechanisms of interaction between concerns, based on developing scenarios to identify architectural aspects and their related tangled components. However ASAAM does not identify the tradeoff interactions between concerns.

The NFR framework [5] uses Softgoal Interdependency Graphs to represent softgoals decomposition. Tradeoffs between requirements are analyzed. We reuse some of these ideas, applying them in the context of AOSD and proposing a systematic process to rank conflicting concerns, which is not offered by the NFR framework.

5 Conclusions & Future Work

Our approach studies interactions between concerns and proposes a classification that takes into account the type of concern being studied. Decisional concerns, those involved in tradeoffs, are ranked using a systematic process. Based on this, the order of concerns' composition can be derived. This approach provides a better understanding of the interactions between all the elements of a system.

We are currently investigating how to generate a set of architectural solutions, supporting traceability between the problem and the solution domain. In the near future we need to focus on how to verify the satisfaction of the quality criteria and to define composition rules for crosscutting architectural concerns.

6 References

1. Araújo, J., Moreira, A., Brito, I., Rashid, A.: Aspect-Oriented Requirements with UML. In: Kandé, M., Aldawud, O., Booch, G., Harrison, B. (eds.): Workshop on Aspect-Oriented Modeling with UML (2002)
2. Baniassad, E., Clements, P., Araujo, J., Moreira, A., Rashid, A., Tekinerdogan, B.: Discovering Early Aspects. IEEE Software special issue on AOSD (2006)
3. Brito, I., Moreira, A.: Advanced Separation of Concerns for Requirements Engineering. VIII Jornadas de Ingeniería del Software y Bases de Datos - JISBD, Alicante, Spain (2003)
4. Brito, I., Moreira, A.: Integrating the NFR framework in a RE model. Workshop on Early-Aspects at the Third International Conference on Aspect-Oriented Software Development, Lancaster, UK (2004)
5. Chung, L., Nixon, B.A., Yu, E., Mylopoulos, J.: Non-Functional Requirements in Software Engineering, Vol. 5. Springer (1999)
6. Clarke, S., Baniassad, E.: Aspect-Oriented Analysis and Design: The Theme Approach. Addison Wesley (2005)
7. Gross, J., Yellen, J.: Handbook of Graph Theory. CRC Press (2003)
8. Jacobson, I.: Object-Oriented Software Engineering: a Use Case driven Approach. Addison-Wesley, Wokingham, England (1995)
9. Kandé, M.M.: A Concern-Oriented Approach to Software Architecture. Lausanne, EPFL (2003)
10. Moreira, A., Rashid, A., Araújo, J.: Multi-Dimensional Separation of Concerns in Requirements Engineering. International Conference on Requirements Engineering. IEEE Computer Society, Paris, France (2005)
11. Pinto, M.: CAM/DAOP: Modelo y Plataforma Basados en Componentes y Aspectos. Dpto. Lenguajes y Ciencias de la Computación, Universidad de Málaga (2004)
12. Rashid, A., Moreira, A., Araújo, J.: Modularisation and Composition of Aspectual Requirements. In: M.Aksit (ed.): 2nd International Conference on Aspect-Oriented Software Development. ACM Press (2003)
13. Tekinerdogan, B.: Synthesis-Based Software Architecture Design. Enschede, The Netherlands (2000)
14. Tekinerdogan, B.: ASAAM: Aspectual Software Architecture Analysis Method. IEEE/IFIP Conference on Software Architecture, Oslo (2004)