



Open Archive Toulouse Archive Ouverte

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible

This is an author's version published in:

<http://oatao.univ-toulouse.fr/22378>

Official URL

DOI : <https://doi.org/10.1016/j.ijar.2018.05.004>

To cite this version: Amgoud, Leila and Ben-Naim, Jonathan
Evaluation of Arguments in Weighted Bipolar Graphs. (2018) International
Journal of Approximate Reasoning, 99. 39-55. ISSN 0888-613X

Any correspondence concerning this service should be sent
to the repository administrator: tech-oatao@listes-diff.inp-toulouse.fr

Composition of Aspectual Requirements: A Multi-criteria Process for Conflict Resolution

Mohamed Amroune, Nacereddine Zarour, Piere Jean Charrel and Jean Michel Inglebert

University of Constantine 2, Algeria

Corresponding Author: Mohamed Amroune, University of Constantine 2, Algeria

ABSTRACT

In Aspect Oriented Software Development, aspects are not only used at the programming level but also tend to arise at the requirements analysis and software architecture design. We previously proposed an approach named AspeCiS (An aspect-oriented Approach to Develop a Cooperative Information System) to develop a Cooperative Information System from existing Information Systems by using their artifacts such as existing requirements and design elements. This approach include an important step in which the aspectual requirements composition problem is considered to be one of the remaining challenges. So, when multiple aspectual requirements share the same join point, undesired behavior may emerge and a conflict resolution process must be triggered. This study presents a conflict resolution process among aspects during the requirements engineering level: A priority value is computed for each aspect and it allows identifying a dominant aspectual requirement on the basis of stakeholder priority. This process is more formal than those currently proposed, which requires a trade-off negotiation to resolve conflicts.

Key words: Requirements engineering, aspect, priority, conflict resolution

INTRODUCTION

During the last years, several researchers have been proposing the use of aspect-oriented concepts in the earlier phases of software development, such as design (Clarke and Walker, 2001; Jose *et al.*, 2000; Sampaio *et al.*, 2007) and requirements analysis (Moreira *et al.*, 2002; Arajo *et al.*, 2004; Jose *et al.*, 2000). This activity is known as Aspect Oriented Requirement Engineering (AORE) and it is the first phase in Aspect Oriented Software Development (AOSD). It is used for the identification, separation, representation and composition of crosscutting concerns. In AORE, the conflict resolution activity is a compulsory process (Sampaio *et al.*, 2007), so it is of interest to achieve it. A conflict may occur when two or more aspectual requirements try to introduce alternatives to a concern through a conflicting behavior (Schauerhuber *et al.*, 2006). In other words conflicting behavior is due to the fact that one aspect tries to vary the concern following a given way while another aspect tries to vary the same concern in a way which is totally opposite to the first aspect.

We earlier presented (Amroune *et al.*, 2011) an approach named AspeCiS (An Aspect approach to develop Cooperative Information Systems), which develops Cooperative Information System (CIS) by reusing the most existing Information Systems artifacts as possible. This approach can

be justified by the fact that it is often difficult for a single Information System (IS) to accomplish all the requirements. So, one solution is to combine several different existing ISs and make them collaborate in order to realize these requirements.

Furthermore, when a new requirement cannot be achieved directly by existing ISs, AspeCiS composes requirements in order to fulfill this new requirement. AspeCiS defines four kinds of requirements, namely: Existing Requirements (ERs), Additional Requirements (ARs), Aspectual Requirements (cf. Section 2.5) and Cooperative Requirements (CRs). To simplify, aspectual requirements are called aspects.

The ERs are requirements provided by existing ISs. However, the ARs are requirements that are not supported by any existing IS. Amroune *et al.* (2011) studied how to elicit and analyze the CRs. The ERs, ARs and aspects are used by a composition process to define CRs related to the CIS. This composition needs a conflict resolution process in order to deal with conflicting situations. So, in AspeCiS, a conflict occurs when an aspect has a negative contribution with another requirement involved in the definition of the same CR. However, each method of software development must support the resolution of conflict situations that may occur during the composition process. AspeCiS not only aims to provide improved support for separation of crosscutting and non-crosscutting requirements but also to provide a better means to identify and manage conflicts due to tangled representations of crosscutting requirements. The problem occurs when several stakeholders are concerned with the conflicting situation (Rashid *et al.*, 2002; Mitchell *et al.*, 1997). The problem is even more difficult in a distributed environment where each stakeholder, may have different roles, responsibilities and concerns. This situation increases the difficulty to negotiate a trade-off with stakeholders. So we put forward the idea to compute a weight for each stakeholder and to identify and resolve conflict at early stages on the basis of these weights.

The main objective of this study is to propose a process, based on the stakeholder's prioritization to resolve the conflict during the composition of aspects at the requirements level of AspeCiS. We make use of an existing formal method.

CONFLICT RESOLUTION TECHNIQUES: AN OVERVIEW

A conflict occurs when two or more aspects try to introduce variation to a concern through conflicting behavior. This conflicting behavior is due to the fact that one aspect tries to vary the concern in a given way while the other aspect tries to vary the same concern in a way which is totally opposite to the first aspect.

In this section we describe several conflict resolution techniques used in AORE.

Pair wise comparison (PCM): Pair Wise comparison Method (PCM for short) (Rashid *et al.*, 2003), is the technique for conflict resolution, to the best of our knowledge, this is the simplest technique proposed up till now. The following steps are performed in order to resolve conflicts according to this technique (Rashid *et al.*, 2003):

- Building a contribution matrix where each aspect may contribute negatively (-) or positively (+) to the others
- Assigning weights to candidate aspects with respect to the importance of that aspect according to a particular stakeholder. Each stakeholder requirement (viewpoint) assigns weight to each candidate aspect which is involved in the contribution matrix. The weight assignment is made according to a scale ranging from 0-1 and any real number is assigned in this range to a particular aspect according to stakeholder requirements

- The conflict with the stakeholders, using step 1 and 2 above for prioritization approach to help communication
- This technique neither uses a method to attribute a weight of actors nor propose a method to assign weights to aspects

MRAT: The use of temporal conflicts in concern composition leads to temporal conflicts which are resolved using the technique known as MRAT (Chitchyan *et al.*, 2007). The operators which are used during composition are “before”, “concurrent”, “after”, etc., which are used for requirement structuring in time with respect to each other (Chitchyan *et al.*, 2007). For instance, a requirement from security concern can demand that data must be encrypted then it should be sent to the server and another requirement can come from a log concern that the request from update should be logged. According to this scenario we have two compositions: encrypt before send and log after send. In this situation the ordering between aspects is simple. But if we had composition such as encrypt before send and log before send, then such undermined ordering leads to conflicts.

Compositional intersection with PCM: Composition interaction (Moreira *et al.*, 2005) is a technique used to find out the trade-off points in a multi dimensional approach to separation of concern. In two dimensional approaches, as stated in Rashid *et al.* (2003) and Niu and Easterbrook (2006), the effect of concern over other concerns is recorded with respect to functional requirements. In this case functional requirements play the role of trade-off points. But, in a multidimensional approach, the functional requirement itself can influence other non functional concerns, so it cannot be taken as a trade-off point (Moreira *et al.*, 2005). The nature of trade-off analysis for multidimensional separation of concerns must be multi dimensional. This can be accomplished by choosing every possible combination of concerns as a basis to study interaction among two concerns and then synthesizing the results of such an analysis. However, this poses a significant overhead, even in the presence of tool support, as the number of potential combinations of concerns, to be used as a base may be extremely high. Therefore, the notion of compositional intersection is introduced to constrain the potential combinations of concerns to be used as a base to those combinations that have real value to offer in term of requirements-level trade-off analysis.

ASPECIS: AN ASPECT APPROACH TO DEVELOP COOPERATIVE INFORMATION SYSTEMS

Before presenting in detail the conflict resolution process, we present the overview of AspeCiS from our previous work (Amroune *et al.*, 2011). First, we define the following concepts; ERs, ARs and CRs.

Definitions:

- **Existing requirement (ER):** It is a statement of services or constraints provided by an existing IS which defines how the system should react to particular inputs and how the system should behave in particular situations
- **Additional requirement (AR):** It is a requirement which is not supported by any existing IS. In this case, other external information systems will be solicited to fulfill such ARs
- **Cooperative requirement (CR):** It is a requirement that will be refined to relate on ERs and eventually ARs, exhibiting what parts of existing systems requirements will be reused and composed and what parts should be newly developed

The main objectives of AspeCiS are: (1) To separate existing requirements from new requirements in the CIS; (2) To provide a high degree of functional reuse, which helps to build again the same requirements on other existing ISs; (3) To propose an aspect approach, which allows weaving existing and additional requirements on join points at the model level. AspeCiS includes three main phases which are:

- Discovery and analysis of CRs; this phase includes a conflict resolution task that can appear during the requirements composition process
- Conception of CRs models
- Preparation of the implementation phase

AspeCiS includes three main phases. In this study, we present only the first phase, because the conflict can appear in this phase.

Phase I: Elicitation and analysis of CRs: This phase is composed of four steps which are: (1) The definition of CRs, (2) The refinement of CRs, (3) The formulation of CRs depending on the ERs and possibly with the definition of some ARs, (4) The selection of a set of **OperatorRequirement** (unary and composite) to be applied to the ERs and the ARs to define the CRs

Definition of CRs: The requirements elicitation activity offers means for the identification and analysis of all requirements. In the literature, several techniques for requirements elicitation are defined (Chitchyan *et al.*, 2007). The choice of an elicitation technique depends on the time and resources available to the engineering requirements and, of course, the kind of information that needs to be elicited (Chitchyan *et al.*, 2007). Nuseibeh presents an interesting classification of elicitation techniques (Nuseibeh and Easterbrook, 2000). This phase must be followed by a refinement process.

Refinement of CRs: The refinement process is composed by two actions, which are: The decomposition of CRs into a set of basic requirements and the use of the inference relation.

- **Decomposition of CRs:** This action aims to decompose these requirements qualified as high-level requirements into a set of ERs and ARs (not decomposable). These requirements are connected by conjunctions or disjunctions nodes. We can distinguish ERs that can be used without any change in the definition of CRs and ERs that must be changed by means of appropriate **OperatorRequirements**. Here, we must apply some **UnaryOperatorRequirements** to these requirements in order to make appropriate changes
- **Cooperative requirements inference:** Within the refinement phase, we can use the inference relation mentioned in Jureta *et al.* (2010). The inference brings the following benefits: (1) It allows dealing with the redundancy of CRs. So, this relation avoids the definition of CRs that can be obtained simply by an inference relation. (2) It allows the requirements engineer dealing with the problem of ambiguity that occurs when one CR has several possible interpretations. (3) It allows reducing development cost and project schedule

Expression of CRs in terms of ERs and ARs: In this sub-phase we determine ERs and ARs involved in the definition of every CR which could not be inferred from others, i.e., we express CR

using a combination of ERs and/or ARs. A set of ERs must be modified in the intention to clearly define the CRs. This modification is provided by a set of OperatorRequirement.

Selection of the Operator Requirements: Usually, the reuse of ERs needs some modifications in order to reach CRs. These changes are ensured by OperatorRequirement. A modification of an ER is the result of the weaving of a new behavior on this ER.

This change is provided by the UnaryOperatorRequirement. So, we consider the UnaryOperatorRequirement as a specific kind of requirement which appears in the case of the definition of CRs. It must be woven with ERs in order to define CRs.

In addition, a UnaryOperatorRequirement can be used several times in the definitions of the CRs. Thus, they are also transverse. On the basis of the fact that these OperatorRequirements are transversal to the ERs&ARs, they are considered as aspects. At this level, several aspects can be composed to define the same CR and the conflict resolution process must be used. The next section presents this process.

CONFLICT RESOLUTION PROCESS TO RESOLVE THE ASPECTS' COMPOSITION

We present a conflict resolution process to resolve the conflict between aspects during composition activity at requirements level of AspeCiS. This process is mainly based on the priorities of aspects and the weights of the stakeholders. To resolve conflicts, we suggested giving a degree of priority for each aspect. For this we propose to use a mathematical function which returns a priority value for each aspect, on the basis of stakeholders' priorities. To prioritize stakeholders we have used the model of Mitchell *et al.* (1997) presented in the next sub-section. Therefore, if the use of this function is unable to resolve the conflict, we also propose two other criteria for conflict resolution that are the sum of the weights and the number of stakeholders involved in the aspect.

Prioritize stakeholders: Mitchell *et al.* (1997) developed a model that includes the attributes of power, legitimacy and urgency. A stakeholder has power when it can influence other stakeholders to make decisions these stakeholders would not have otherwise made. Mitchell *et al.* (1997) relied on Berander and Andrews (2005) categorization of power: coercive power, based on the physical resources of force, violence, or restraint; utilitarian power, based on material or financial resources and normative power, based on symbolic resources.

Legitimacy is determined by whether the stakeholder has a legal, moral, or presumed claim that can influence the organization's behavior, direction, process or outcome. Stakeholders are risk-bearers who have "invested some form of capital, human or financial, something of value, in a firm". Mitchell *et al.* (1997) used the notion of risk to narrow stakeholders with a legitimate claim. These stakeholders are often dependent on the organization. The combination of power and legitimacy is authority.

Urgency exists under two conditions "(1) When a relationship or claim is of times sensitive nature and (2) When that relationship or claim is important or critical to the stakeholder". Urgency, then, requires organizations to respond to stakeholder claims in a timely fashion. Urgency alone may not predict the priority of a stakeholder, especially if the other two attributes are missing.

This model uses the combination of the three attributes to develop a prioritization strategy. Accordingly, latent stakeholders possess only one of the attributes; expectant stakeholders possess two attributes and definitive stakeholders possess all three attributes. If individuals or groups do not possess any of the attributes, they are not considered as stakeholders (Gruning and Hunt, 1984):

- The latent stakeholders have lower salience to an organization because they only have one attribute. They are identified as dormant, discretionary and demanding
- The dormant stakeholder has power but no legitimacy or urgency in his claim. Therefore his power remains unused
- The discretionary stakeholder possesses legitimacy but no power to influence and no urgency in the claim and therefore is reliant on the good will of the organization rather than through any other pressure
- The demanding stakeholder has urgency but no legitimacy or power
- These groups could be bothersome but not dangerous
- The expectant stakeholders possess two attributes and are organized into dominant, dependent and dangerous stakeholders
- The dominant stakeholders have power and legitimacy and because they can act on their claims, they receive much of management's attention
- The dependent stakeholders have legitimacy and urgency. Organizations should be socially responsible to stakeholders that have a legitimate and urgent claim and who depend on the organization to address and resolve the claim. The inclusion of a dependent relationship is important because it recognizes that the stakeholder priority is not limited to influence over the organization
- The Dangerous stakeholders have urgency and power but lack legitimacy

Most of the time, these stakeholders use formal channels to affect change but may they become violent or coercive to achieve their claims. Social activist groups sometimes engage in forms of protests, boycotts and (in extreme cases) damage to property and lives. Stakeholders who have all three attributes are definitive stakeholders and have the highest priority. An important tenet of this model is that each attribute is variable and not constant. In other words, any group can acquire (or lose) power, legitimacy, or urgency depending on the situation. Therefore, an expectant stakeholder group can become a definitive stakeholder if it acquires the third attribute. A dangerous stakeholder group can acquire legitimacy, as has been the case with many nongovernmental organizations over the last few years. A dependent stakeholder group, such as a community affected by the irresponsible corporate behavior, can acquire the power by appealing to governmental agencies.

Different steps of the conflict resolution process: The proposed process is composed primarily of four major Steps. The first determines for each CRs the stakeholders involved. In the second the process determines the priority values assigned by each stakeholder to a different Aspect used to define CRs. The third step of the process determines the mutual influences between Aspects. However the fourth step concerns the conflict resolution. Our process about conflict resolution is described as follows:

Step 1: The (Stakeholders*CRs) matrix is defined (Fig. 1). In this step, the process determines the stakeholders involved in the definition of each CRs. The cells containing the tick sign indicates that the stakeholder is in someway related to the CRs of this column. The cell which contains nothing indicates that the stakeholder don't care about the CRs of these column

CRs stakeholders	CR ₁	CR ₂	...	CR _i	CR _n
Stakeholder ₁	x				
Stakeholder ₂		x		x	x
Stakeholder _i	x	x			
		x		x	x
Stakeholder _m				x	

Fig. 1: (Stakeholders*CRs) matrix

Aspect (A) Stakeholders	Stakeholder's priority	A1	A2		A _n
Stakeholder ₁	Ps ₁	A _{1,1}	A _{2,1}		A _{n,1}
Stakeholder ₂	Ps ₂				
Stakeholder _i	Ps _i	A _{1,i}	A _{2,i}		A _{n,i}
P(A _i)		W1	..	W _j	W _n
SW(A _i)		SW1	..	Sw _j	SW _n
NS(A _i)		NS1		Ns _j	NS _n

Fig. 2: (Stakeholders*Aspects) matrix (MS_A)

Step 2: In this Step, we build for each CRs the (CRs* Aspects) matrix (MS_A for short), for relationship between the aspects involved in the CR expressed by each stakeholder (Fig. 2). In this matrix each stakeholder assigns weight to each Aspect. The weight assignment is made according to a scale. The scale is ranged from 1-5. In this step, stakeholders assign a priority value for each aspect involved in the definition of CR

This step aims to prioritize stakeholders identified according to their roles in the CRs. To prioritize them we have adopted Mitchell *et al.* (1997) model.

Last three lines of the MS_A matrix denote:

- **P(A_i):** Describes how to give a degree of priority of a specific aspect (A) on the basis of stakeholders` priorities and their evaluation of a specific Aspect. The weights of Aspects are calculated by the function P as described below.

We define a following prioritization function P by the following equation:

$$P(\{(S_p, SP_p, A_p, AP_{ij})\}, i = 1..n, j = 1..m) = (A_p, P(A_j))$$

Aspect Aspect	A ₁	A ₂	...	A _i	A _n
A ₁	+			-	-
A ₂				+	-
:					
A _i					+
:					
A _n					

Fig. 3: (Aspect*Aspect) matrix

Where:

- S_i = 1..n are the stakeholders
- Sp_i = Stakeholder's S_i degree of priority
- A_{i,j} = 1..n denotes the current Aspect (A)
- Ap_{i,j} = Denotes the degree of priority given by the stakeholder S_i to the Aspect A_j
- Ap_{i,j}^a = {1 (very low); 2 (low); 3 (medium); 4 (high); 5 (very high)}
- P(A_i) = Denotes the A_i priority resulting from the function P, P(A_i) ∈ [1,5]

Where:

$$P(A_i) = \frac{\sum_{j=1}^n A_{i,j} * PS_i}{\sum_{j=1}^n PS_i}$$

- **SW(A_i):** Denotes the sum of the weights of the stakeholders how have expressed the aspect (A_i):

$$SW(A_i) = \sum_{j=1}^m PS_i$$

- **NS(A_i):** Denotes the number of the stakeholders how have expressed the Aspect (A_i)

Step 3: The (Aspect (A)*Aspect (A)) matrix (MA_A for short) is built, to express the relationship between the Aspects, where each aspect which may contribute negatively (-) or positively (+) to the others. The negative contribution means that the aspects do not complement each other rather they intersect. On the other hand the positive contribution among aspects means that both aspects complement each other. MA_A is a symmetrical matrix. This step is inspired from (Rashid *et al.*, 2003) (Fig. 3)

Step 3.1: The Conflict Matrix (MC) deduct from the matrix MA_A another so-called conflict matrix denoted MC. The elements of this matrix must verify the following conditions:

$$\begin{cases} C[i,j]=MA[i,j]^1 & \text{and} \\ MA[i,j]^2 = "-" & \text{and} \\ NS[i] > 0 \text{ (Fig. 1)} \text{ and} \\ NS[j] > 0 \text{ (Fig. 1)} \end{cases}$$

- MC elements belong to $\{-, " "\}$
- $NS[i] = 0$ means that any stakeholder has expressed this $A(i)$. So $NS[i]$ denotes the number of the stakeholders how have expressed the $A(i)$ in the (Stakeholders*CRs) matrix (Fig. 1)

Step 4: The algorithm described in this section presents the process of the conflict resolution. This process is used for each case of conflict (detected in the matrix of conflict MC, presented in the stage 3.1). Furthermore, four criteria are proposed in order to resolve conflict situations

Criteria of conflict resolution: In the composition process several conflict situations can be appeared. So, to adjust these situations, the proposed process uses four criteria which are.

First criterion: If two Aspects are in conflict, the algorithm compares the two values of priority calculated by the function P described in step 2. The Aspect dominating is the one that has the greatest value of P.

Second criterion: If the first criterion did not solve the problem. So for each Aspect, the algorithm compares the sum of the weights of the involved stakeholders, this value is calculated by the function SW (step 2). The Aspect (A) dominating is it the one that has the greatest value of SW.

Third criterion: In this case, the priority is given to the Aspect, which has the greatest number of stakeholders. This value is calculated by the function NS (step 2).

Fourth criterion: In this case, two other parameters are defined. These parameters are: the aspect's frequency (FreqA) and the number of conflicts (NbreConfA).

The Aspect's frequency (FreqA_i) denotes the number of participation of the aspect (A_i) in the definition of CRs. Whereas, (NbreConfA_i) denotes the number of conflict cases in which this aspect is present.

For these reasons, another matrix is defined, is called the frequency matrix (Fig. 4). It is defined as follows:

Aspect	FreqA _i	NbreConfA _i
A ₁	FreqA ₁	NbreConfA ₁
A ₂	FreqA ₂	NbreConfA ₂
A _n	FreqA _n	NbreConfA _n

Fig. 4: Frequency matrix

In this case the aspect priority is defined by the P1 equation:

$$P1(A_i) = \frac{FreqA_i}{NbConfA_i} \times \frac{\sum_{j=1}^n A_{i,j} * PS_j}{\sum_{j=1}^n PS_j}$$

It is important to note that:

$$P1(A_i) = \frac{FreqA_i}{NbConfA_i} \times P(A_i)$$

i.e., that the priority of aspect "A_i" is recomputed based on the factor of frequency obtained by the equation:

$$\frac{FreqA_i}{NbConfA_i}$$

This factor plays a significant role in the conflict resolution process; it attributes the priority to the aspect, which contributes more in the definition of CRs, with a minimum of conflict.

The result of this algorithm is a matrix called matrix of conflict resolution (MCR for short). The elements of this matrix belong to {"<<", ">>", "- "}. So if the MCR[i,j] equal at "<<" this means the Aspect (A_i) dominates the Aspect (A_j) but if it equal at ">>" means that the Aspect (A_j) dominates the (A_i).

If the proposed process could not resolve the conflict between (A_i and A_j) in this case MCR [i, j] is equal to.

"- ". Below we present the proposed process to resolve the conflict by using an algorithmic notation.

Algorithm ResolveConflict

Input

MC: Matrix /* the conflict matrix deduced in stage 3.1.* /

Output

MCR³: Matrix /* the conflict resolution matrix.* /

Begin

For I := 1 to N Do

 For J := 1 to (I-1) Do

Begin

 If MC[I,J] = "-" Then

 If P(A_i) > P(A_j)

 Then MCR[I,J] := "<<"

/* ("CA_i is the dominant Aspect")* / Else

 If P(A_i) < P(A_j) Then MCR[I,J] := ">>"

Else /* ("Use of the second criterion")* /

 If SW(A_i) > SW(A_j) Then

 MCR[I, J] := "<<" Else

 If SW(A_i) > SW(A_j) Then

 MCR[I,J] := "<<" Else

³The matrix of conflict resolution ("MCR") contains all cases of conflict have been resolved by the proposed process,

and cases of conflict which require a trade off

```

If SW(Ai) < SW(Aj) Then MCR[I,J]:= ">>"
Else/* ("Use of the third criterion")*/
If NS(Ai) > NS(Aj) Then
MCR[I,J]:= "<<"           Else
If NS(Ai) < NS(Aj) Then    MCR[I,J]:= ">>"
Else /* use of the frequency factor*/
If P1(Ai) > P1(Aj)
Then    MCR[I,J]:= "<<"
Else    If P1 (Ai) < P(Aj) Then    MCR[I,J]:= ">>"
ELSE    MCR:= "-" /* ("Trade Off")*/
EndIF; End;
End

```

Table 1: CRs and stakeholders matrix

CRs stakeholders	Register vehicle	Pay vehicle	Enter motorway	Motorway	Pass single tool	Register vehicle
Bank	x	x				x
Vehicle owner	x	x				x
Vehicle driver			x	x		

A motivating example: In this section we give an example of conflict resolution to illustrate the described process. It is inspired from the toll collection system on the Portuguese highway (Rashid *et al.*, 2003). The following explanation just shows the conflict resolution by the proposed function. Since our work, is only based on conflict resolution, so the identification, representation and separation of concern will be assumed to be the same as mentioned in our previous work (Amroune *et al.*, 2011):

The stakeholders are: Bank, Vehicle owner and vehicle driver (Table 1). The CRs are:

- **(CR1):** Register vehicle, (CR2): Pay vehicle
- **(CR3):** Enter motorway, (CR4): Exit motorway and
- **(CR5):** Pass single tool

The aspects are:

- **A1:** Security; **A2:** Response time; **A3:** Multi access system

Step 2: The (Stakeholders*Aspect) matrix (MSCA) is as follows: Attributing weight for every aspect according to every stakeholder and then composing it in a form of a matrix. For the CR1 (Table 2)

Step 3: For the CR1, the (Aspect* Aspect) matrix (MA_A) is as follows (Table 3)

- The (Aspect * Aspect) conflict matrix (MC) is as follows (Table 4)

Step 4: Detection and resolution of conflict: the following matrix that described in step 2, is regarded as an input parameter to our algorithm for conflict resolution. To this we add the last two columns to refer some remarks (Table 5)

Table 2: Stakeholders*aspect matrix

Aspect (A) stakeholders	Stakeholder's priority	A1	A2	A6
Bank	2	2	3	
Vehicle owner	3	3	2	2
P(Ai)		2,6	2,4	2
SW(Ai)		5	5	3
NS(Ai)		2	2	1

Table 3: Aspect*aspect matrix for CRI

Aspects aspects	A1	A2	A6
A1: Security		-	+
A2: Response time	-		-
A6: Correctness	+	-	

Table 4: Aspect*aspect conflict matrix for CRI

Aspect Aspect	A1	A2	A6
A1: Security		-	
A2: Response time	-		-
A6: Correctness		-	

Table 5: Detection and resolution of conflict matrix

Aspect stakeholders	Stakeholder's priority	A1	A2	A6	Conflict situation
Bank	2	2	3		A2
				<<	A1* ⁴
Vehicle owner	3	3	2	2	A1
				<<	A2
P(Ai)		2,6	2,4	2** ⁵	
SW(Ai)		5	5	3	
NS(Ai)		2	2	1	

We can easily see the two cases of conflict between A1 and A2 (i.e.,: $A2 << A1$ and $A1 << A2$) and also between A2 and A6 (i.e., these aspects have the same value of priority (2)). Therefore, the application of the algorithm (described in step 4) gives the following results.

- **For the first case of conflict:** The conflict between (A1, A2) is solved with the first criterion of the algorithm, then through the function P, in this case ($A1 << A2$) means that A1 dominates A2, i.e., that A1 holds the greatest priority ($W(A1) = 2,6 > W(A2) = 2,4$)
- **For the second case of conflict:** In this case the conflict is resolved by the second criterion so, the algorithm compares the sum of the weights of the involved stakeholders, this value is calculated by the function SW:

$$(SW(A1) = 5 > SW(A2) = 3)$$

Consequently the matrix of conflict resolution (MCR) is as follows (Table 6)

⁴Conflict inter stakeholders, ⁵Result of the Composition : $A1 << A2$ and $A2 << A6$

Table 6: Matrix of conflict resolution (MCR)

Aspect Aspect	A1	A2	A6
A1:Security		<<	
A2:Response time	>>		<<
A6:Correctness		>>	

CONCLUSION

In Aspect oriented software development approaches, the importance of identifying and resolving aspect interactions is widely recognized. This paper has proposed a process for conflict resolution between aspects during the requirements engineering level, especially during the analysis phase of AspCiS (Amroune *et al.*, 2011).

The Important ideas presented here are those of prioritization of stakeholders (inspired from Mitchell *et al.* (1997) model), the use of stakeholders weights to compute the weights of aspects, where each one may contribute negatively to the others at this stage. We have proposed a mathematical function the parameters of which are a set of quadruples. Each quadruple encapsulates essentially the priority value of the corresponding stakeholder and the weight she/he gives to the current Aspect. We also proposed two other criteria for conflict resolution that are the sum of the weights and the number of stakeholders involved in an aspect.

The planned extension to this work includes an investigation of the following research question: "how to resolve the conflict when the use of stakeholders weights could not resolve the issue of conflict ».

REFERENCES

- Amroune, M., J.M. Inglebert, N. Zarour and P.J. Charrel, 2011. AspeCiS: An aspect-oriented approach to develop a cooperative information system. Proceedings of the 1st International Conference on Model and Data Engineering, September 28-30, 2011, Obidos, Portugal, pp: 122-132.
- Arajo, J., J. Whittle and D. Kim, 2004. Modeling and composing scenario-based requirements with aspects. Proceedings of the 12th IEEE International Requirements Engineering Conference, September 6-11, 2004, Caparica, Portugal, pp: 58-67.
- Berander, P. and A. Andrews, 2005. Requirements Prioritization. In: In: Engineering and Managing Software Requirements, Aurum, A. and C. Wohlin (Eds.). Springer, Berlin, Heidelberg, pp: 69-94.
- Chitchyan, R., A. Rashid, R. Wathers, I. Brito, A. Moreira and J. Araujo, 2007. Requirements-level aspectual trade-off analysis approach. AOSD-Europe. <http://scc-aosd.lancs.ac.uk/deliverables/d74.pdf>
- Clarke, S. and R.J. Walker, 2001. Composition patterns: An approach to designing reusable aspects. Proceedings of the 23rd International Conference Software Engineering, May 12-19, 2001, IEEE Xplore, London, pp: 5-14.
- Gruning, J.E. and T.T. Hunt, 1984. Managing Public Relations CL. Holt, Rinehart and Winston, USA., pp: 141.
- Jose, H., F. Sanchez, F. Lucio and M. Toro, 2000. Introducing separation of aspects at design time. Proceedings of the 14th European Conference on Object-Oriented Programming, June 11-12, 2000, NY., USA.

- Jureta, I.J., A. Borgida, N.A. Ernst and J. Mylopoulos, 2010. Techne: Towards a new generation of requirements modeling languages with goals, preferences and inconsistency handling. Proceedings of the 18th International Conference on Requirements Engineering (RE), September 27-October 1, 2010, Sydney, NSW, pp: 115-124.
- Mitchell, R.K., B.R. Agle and D.J. Wood, 1997. Toward a theory of stakeholder identification and salience: Defining the principle of who and what really counts. *Acad. Manage. Rev.*, 22: 853-886.
- Moreira, A., A. Rashid and J. Araujo, 2005. Multi-dimensional separation of concerns in requirements engineering. Proceedings of the 13th IEEE International Conference on Requirements Engineering, August 29-September 2, 2005, Paris, France, pp: 285-296.
- Moreira, A., J. Araujo and I. Brito, 2002. Crosscutting quality attributes for requirements engineering. Proceedings of the 14th international Conference on Software Engineering and Knowledge Engineering, July 15-19, 2002, Italy, pp: 167-174.
- Niu, N. and S. Easterbrook, 2006. Discovering aspects in requirements with repertory grid. Proceedings of the International Workshop on Early Aspects at ICSE, May 20-28, 2006, Shanghai, China, pp: 35-42.
- Nuseibeh, B. and S. Easterbrook, 2000. Requirements engineering: A roadmap. Proceedings of the Conference on The Future of Software Engineering, June 4-11, 2000, Limerick, Ireland, pp: 35-46.
- Rashid, A., A. Moreira and J. Araujo, 2003. Modularization and composition of aspectual requirements. Proceeding of the 2nd International Conference on Aspect-Oriented Software Development, March 17-21, 2003, Boston, pp: 11-20.
- Rashid, A.P. Sawyer, A. Moreira and J. Araujo, 2002. Early aspects: A model for aspect-oriented requirements engineering. Proceedings of the Joint International Conference on Requirements Engineering, September 9-13, 2002, Essen, Germany, pp: 199-2002.
- Sampaio, A., P. Greenwood, A.F. Garcia and A. Rashid, 2007. A comparative study of aspect-oriented requirements engineering approaches. Proceedings of the 1st International Symposium on Empirical Software Engineering and Measurement, September 20-21, 2007, Madrid, pp: 166-175.
- Schauerhuber, A., W. Schwinger, E. Kapsammer, M. Wimmer and W. Retschitzegger, 2006. Towards a common reference architecture for aspect-oriented modeling. Proceedings of the 8th International Workshop on Aspect Oriented Modeling, March 2006, Bonn, Germany.