

A Fuzzy Interval based Approach for Handling Conflicts in AORE

Narender Singh

Department of Computer Science & Applications,
Maharshi Dayanand University, Rohtak, India.

Nasib Singh Gill

Department of Computer Science & Applications,
Maharshi Dayanand University, Rohtak, India.

ABSTRACT

In AORE, a conflict occurs when two or more crosscutting concerns i.e aspects having the same priority contribute negatively to each other, need to be composed in the same match point. Conflict resolution is a process that establishes a critical trade-off among such kind of aspects. So, the conflict resolution process is a compulsory process and need to achieve it. Over the last few years, several research efforts have been devoted to resolve conflict in AORE but, still a lot of work is needed. The use of fuzzy logic to conflict resolution is an emerging area that will incorporate one domain in other. In this paper, an attempt is made to apply fuzzy logic for the conflict resolution in AORE.

Keywords: Aspect-Oriented Software Development, Aspect-Oriented Requirements Engineering, Conflict Resolution, fuzzy interval.

1. INTRODUCTION

Aspect-Oriented Software Development (AOSD) provides a systematic mechanism to handle crosscutting concerns for improved modularization and composition. It focuses on crosscutting concerns by providing means of their systematic identification, separation, representation and composition [1]. It encapsulates crosscutting concerns in separate modules, known as aspects. It later uses composition mechanism to weave them with other core modules at loading time, compilation time, or run-time [2].

Aspect-Oriented Requirements Engineering (AORE) [1] is an early phase in AOSD that supports separation of crosscutting concerns at requirements level. It not only aims to provide improved separation of crosscutting concerns during requirements engineering, but also to provide a better means to identify and manage conflicts arising due to tangled representations of crosscutting concerns. Since during requirement phase, a more direct contact is maintained with stakeholders [3], which assists negotiation and decision-making among stakeholders. Such means of early conflict resolution help to establish critical tradeoffs even before the architecture is derived [4].

A conflict occurs among two or more concerns when one of them tries to vary in a way that is totally opposite i.e contribute negative to others [4A]. In AORE, a conflict occurs when two or more crosscutting concerns i.e aspects having the same priority contribute negatively to each other, need to be composed in the same match point. A match point identifies specific locations in the base concerns where other concerns' behaviour should be composed, or satisfied. Conflict resolution is a process that establishes a critical trade-off among such kind of aspects. So, the conflict resolution process is a compulsory process [5] and need to achieve it.

Fuzzy logic was invented by Zadeh [6] in 1965 for modelling the uncertain and imprecise knowledge in human reasoning. It

is suitable for the representation of vague data and concepts on an intuitive basis, such as human linguistic description, e.g. the expressions *very important*, *less important*, *large*, *young etc.* hence, has proved to be a powerful tool for decision-making in human reasoning, and to handle and manipulate imprecise and noisy data.

Over the last few years, several research efforts have been devoted to resolve conflict in AORE but, still a lot of work is needed. The use of fuzzy logic to conflict resolution is an emerging area that will incorporate one domain in other. In this paper, an attempt is made to apply fuzzy logic for the conflict resolution in AORE, but the work still to be modified and need some automation.

2. RELATED WORK

As mentioned above, a conflict occurs among concerns when they contribute negative to each others, and conflict resolution is a process that establishes a critical trade-off among such kind of concerns. Various approaches like [7], [8], [9], [10], [11], [12], [13] etc have been proposed to resolve conflicts in requirements engineering.

AORE is still a hot research area, which focuses on identifying, analyzing, specifying, verifying, and managing the crosscutting concerns at the early stages of software development. Over the last few years, several research efforts have been devoted to developing AORE models that can help in extracting, identifying, and modelling aspects in the early phase of requirements analysis. Some success in this direction has been achieved. But, still managing conflicts among multiple viewpoints or stakeholders is immature and needs more attention with many research issues.

In last few years, many researchers contributed their significant work in this area for resolving conflicts but most of them have used simple reasoning techniques based on intuitive that do not allow a rigorous engineering approach to the problem. In this section, we briefly discuss these efforts.

Brito *et al.* [14] proposed an approach to resolved conflicts by identifying dominant concerns. This is accomplished analyzing the *Priority* row of the template to describe concerns. A concern is dominant that has higher priority in a situation where each concern has different priority. Hence, it is not difficult to solve. However, the problem becomes more complicated if at least two crosscutting concerns have the same priority for a given match point with negative contribution to each other. Here, a negotiation is required with respective stakeholders so that the concerns can be prioritized differently to obtain the dominant crosscutting concern. The process begins with analyzing the dominant concern among two crosscutting concerns, further analyzing this dominant concern with third concern and so on until considering all concerns for a match point. Here, we get a concern with higher priority among all concerns for a match point. Similarly, we can get other dominant concerns and hence resolve conflicts.

Rashid *et al.* [1] and Moreira *et al.* [15] proposed almost similar approaches for conflict resolution. They assigned weights to those concerns that contribute negatively to each other and used Pair wise Comparison Method for conflict resolution, which is the simplest method in literature. The process begins with building a contribution matrix where each aspect may contribute negatively (-) or positively (+) to other concerns. Next, weights are assigned to each concern in relation to a set of stakeholders' requirements. Finally, resolve the conflicts by pair wise comparing weights for prioritization.

Chitchyan *et al.* [16] presented a requirements description language (RDL) that uses Multi-Dimensional Requirements Analysis Tool [17] to support analysis of crosscutting requirements. It enhanced the existing natural language requirements specification with semantic information derived from the semantics of the natural language itself. Composition specifications are written based on these semantics rather than requirements syntax hence providing improved means for expressing the intentionality of the composition, in turn facilitating semantics-based reasoning about aspect influences and trade-offs.

Hameed *et al.* [18] proposed scenario-based priority comparison method for resolving conflict with in AORE. This method is an extension of pairwise comparison method [1]. The extended approach consists of the steps like building the (concern*concern) matrix, prioritization of every concern on the basis of scenario modelling, building the modified (stakeholder*concern) matrix for representing weights for concerns with respect to each stakeholder, and resolving the conflicts on the basis of all earlier steps.

Sardinha *et al.* [19] proposed an approach called *Utilizing Search-Based Techniques to Resolve Conflicts in AORE*. This approach presents a mathematical formulation that can be implemented with a popular search-based technique called Genetic Algorithm, which is composed of an aspect-oriented specification, stakeholders' importance and priorities, and available resources for implementing the requirements. Here, they aimed to select requirements for resolving conflict that can maximize the stakeholders' satisfaction and respect the availability of resources.

3. OUR PROPOSAL

Our approach is an extension to already existing approach in [18] for resolving conflicts. Here, we apply the concepts of fuzzy logic which is most suitable for representing vague and imprecise data such as human linguistic description, e.g. the expressions *very important*, *important*, *less important*.

Fuzzy logic was invented by Zadeh [6] in 1965 modelling the uncertain and imprecise knowledge in human reasoning. It is suitable for the representation of vague data and concepts on an intuitive basis, such as human linguistic description, e.g. the expressions *very important*, *less important*, *large*, *young* etc. hence, has proved to be a powerful tool for decision-making in human reasoning, and to handle and manipulate imprecise and noisy data.

Hence our proposal for conflict resolution with in AORE using fuzzy logic will prove to be a powerful tool in this area. The proposed approach consists of the following steps:

Step 1: *Building a contribution matrix (Concern * Concern)* where each concern may contribute negatively or positively to the others. This step is same as described by [1].

Step 2: *Prioritization of scenarios* is achieved by scenarios modelling where two scenarios are generated for each

conflicting concern and further a utility tree is derived for these scenarios. Utility tree is a mechanism used with in "Architecture Tradeoff analysis method" [20] to assign a priority to the scenarios. We have considered two aspects to prioritize the scenarios which are: importance level of stakeholder for the scenario, and priority assigned by stakeholder to the scenario. The values are given according to the importance each aspect has for each stakeholder.

The scales we are using are based on ideas from fuzzy logic as described in [1] and have the following meaning:

- *Very important* takes values in the interval [0.8, 1.0]
- *Important* takes values in the interval [0.5, 0.8]
- *Average* takes values in the interval [0.3, 0.5]
- *Not so important* takes values in the interval [0.1, 0.3]
- *Do not care* much takes values in the interval [0, 0.1]

Considering the fuzzy values for both stakeholders' importance (e.g. very important, important, average, not so important, and do not care) and stakeholders' priority (e.g. very important, important, average, not so important, and do not care) facilitates the prioritization of scenarios. This prioritization is done by using modified fuzzy interval arithmetic [21] defined as:

For any two given fuzzy intervals $[x_1, x_2] \in \mathbb{R}$ and $[y_1, y_2] \in \mathbb{R}$; and also $[x_1, x_2] \subseteq [0, 1]$ and $[y_1, y_2] \subseteq [0, 1]$,

$[x_1, x_2] \cdot [y_1, y_2] = [m(x) \cdot m(y) - k, m(x) \cdot m(y) + k] \subseteq [0, 1]$, where $k = \min \{m(x) \cdot m(y) - \alpha, \beta - m(x) \cdot m(y)\}$, $\alpha = \min \{x_1y_1, x_1y_2, x_2y_1, x_2y_2\}$ and $\beta = \max \{x_1y_1, x_1y_2, x_2y_1, x_2y_2\}$.

The prioritization of the scenario is done by assigning weight to each scenario according to the table 2.

Step 3: *Prioritization of concerns* is done by taking the average of the weights of the associated scenarios calculated as in above step 2.

Step 4: *Concern / Stakeholder matrix* The cells containing the tick sign indicates that the stakeholder for this cell is in someway related the aspect of this cell. The cell which contains nothing indicates that the stakeholder for these cell don't care about the aspects of these cells.

Step 5: *Weighted Concern / Stakeholder Matrix* is obtained by assigning weights to all candidate aspects (i.e crosscutting concerns) with respect to their importance to each stakeholder involved. These weights are assigned by using the scale based on fuzzy values as mentioned above in step 2. This weight assignment to every crosscutting concern by all concerned stakeholders results in the form of a matrix. The only difference between Concern / Stakeholder Matrix and Weighted Concern / Stakeholder Matrix is that in later matrix, we use weights instead of tick sign for all concerns, who contribute negative (-) in contribution matrix depicted as table 1.

Step 6: *Modified Concern / Stakeholder Matrix* is an extension to already constructed matrix in step 5. The only difference between these matrices is that the weight assigned by the stakeholder for each concern in previous matrix is multiplied by the weight of the concern, which is calculated in prioritization of concern i.e step 3.

Step 7: *Conflicts resolution using all above steps.*

Table 1: Contribution matrix (Concern * Concern)

	Aspect ₁	Aspect ₂	Aspect _n
Aspect ₁		+		
Aspect ₂		-		-
.....				
Aspect _n				

Table 3: Concern / Stakeholder matrix

	S ₁	S ₂	S ₃	S _m
C ₁	√	√			√
C ₂		√	√		
.....					√
C _n		√			

Table 2: Weight assignment matrix

Priority Importance	VIM [0.8, 1.0]	IMP [0.5, 0.8]	AVG [0.3, 0.5]	NSI [0.1, 0.3]	DNC [0.0, 0.1]
VIM [0.8, 1.0]	[0.64, 0.98]	[0.40, 0.77]	[0.24, 0.48]	[0.08, 0.28]	[0.0, 0.09]
IMP [0.5, 0.8]	[0.40, 0.77]	[0.25, 0.60]	[0.15, 0.37]	[0.05, 0.21]	[0.0, 0.07]
AVG [0.3, 0.5]	[0.24, 0.48]	[0.15, 0.37]	[0.09, 0.23]	[0.03, 0.13]	[0.0, 0.04]
NSI [0.1, 0.3]	[0.08, 0.28]	[0.05, 0.21]	[0.03, 0.13]	[0.01, 0.07]	[0.0, 0.02]
DNC [0.0, 0.1]	[0.0, 0.09]	[0.0, 0.07]	[0.0, 0.04]	[0.0, 0.02]	[0.0, 0.01]

4. CASE STUDY

For illustrating the approach, we apply it to a case study presented in [22] originally adopted from [23]. The case study is about the First Responder Interactive Emergency Navigational Database (FRIEND), an accident management system. The system is being developed to help and manage the enormous amounts of information involved with accident management [23]. It supports several classes of users including first responders (workers in the field), field supervisors, Dispatchers, and resource allocators. These users collaborate with the help of this system to manage the information associated with an accident(s), including resource information, activities and actions taken in response to an accident, geographical information, Emergency Operations Plan (EOP) information, and hazardous materials information. The requirements are stated as follows: "In FRIEND system, a field officer, such as a police officer or fire fighter, has access to a wireless computer that enables them to interact with a Dispatcher. The Dispatcher in turns can visualize the current status of all its resources, such as a police van or a fire unit or a paramedic unit, on a computer screen and dispatch a resource by issuing commands from a workstation. The system administrator is responsible for managing all users and terminals and also for assigning permissions to different users. The system administrator should be able to store the different users and their permissions, restricting their access [24]"

Since our work is only based on conflict resolution between concerns so the identification, representation and separation of concern will be assumed to be the same as mentioned in [22]. The concerns as identified in [22] for this case study are RE: Report Emergency, OI: Open Incident, AR: Allocate Resources, VM: View Map, CN: Concurrency, RT: Response Time, AC: Accuracy, CP: Compatibility, AV: Availability, and SC: Security.

According to the case study in [22] the "Field Officer" stakeholder shows a conflicting behaviour between security, response time and concurrency because these aspects contribute negatively to each other. Similarly "Other Databases" stakeholder also exhibits conflicting behaviour between response time and concurrency.

*Step 1: Building a contribution matrix (Concern * Concern):* The contribution matrix defines the relationship between different concerns. The relationship can be positive or negative depending upon the application being considered. In our case study the relationship between concerns is given as, Response time, security, concurrency and availability are the concerns which contain negative signs in their respective cells in the matrix, while the other concerns contain positive signs which mean that those concerns affect each other in a positive manner thus not leading to a conflict. Here, we will only deal with three concerns i.e. response time, security, and concurrency.

Table 4: Contribution matrix

(RE: Report Emergency, OI: Open Incident, AR: Allocate Resources, VM: View Map, CN: Concurrency, RT: Response Time, AC: Accuracy, CP: Compatibility, AV: Availability, SC: Security)

	RE	OI	AR	VM	RT	AV	SC	CM	AC	CN
RE					+	+	+		+	+
OI				+	+	+	+		+	+
AR				+	+	+	+		+	+
VM					+	+			+	+
RT						+	-		-	-
AV										-
SC									+	
CM										
AC										
CN										

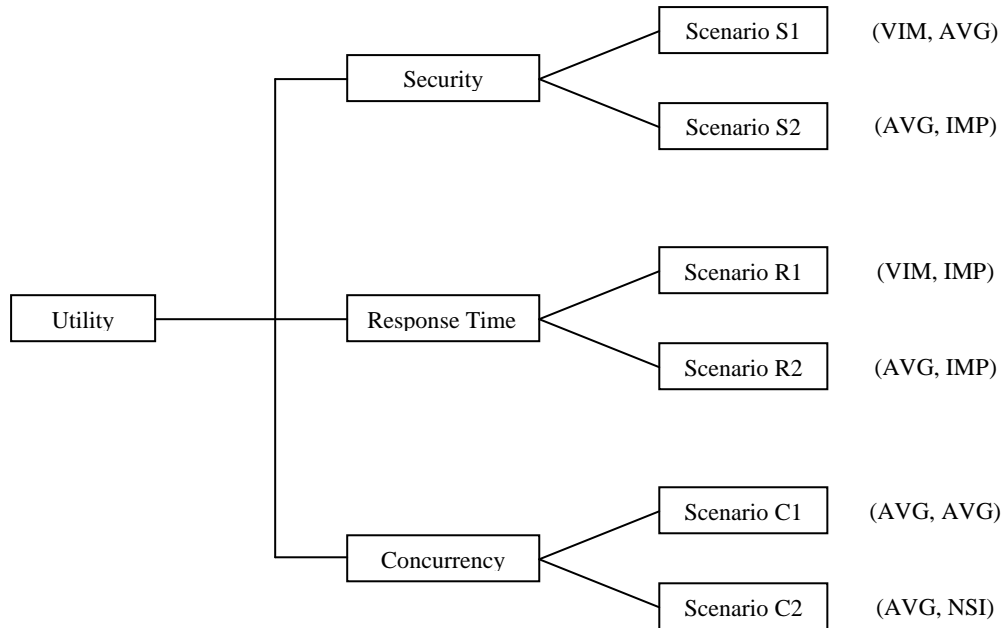


Figure 1: Utility for the case study

Step 2: *Prioritization of scenarios* is achieved by scenarios modelling where two scenarios are generated for each conflicting concern and further a utility tree is derived for these scenarios.

We first generate the scenario for the concern security

#1: The system administrator is responsible for managing all users and terminals and also for assigning permissions to different users.

#2: The system administrator should be able to store the different users and their permissions, restricting their access.

Next, we generate the scenario for the concern response time,

#1: Responds within time (<t) by acknowledging the successful submission of form to the FieldOfficer.

#2: Responds within time (<t) by acknowledging FieldOfficer the selected response submitted by Dispatcher.

Next, we generate the scenario for the concern concurrency,

#1: The primary purpose of FRIEND is to provide users concurrent access to a set of shared information. They may access the data simultaneously or serially.

#2: Concurrent users must see changes to the data as quickly as possible.

The prioritization of the scenario is done by assigning weight to each scenario according to values depicted as table 2. For example, a scenario S1 assigned a weight of (VIM, AVG) means that the importance level of stakeholder for the scenario is VIM (i.e. very important) while the priority assigned by stakeholder to the scenario is AVG (i.e. average). Hence, the scenario S1 has a weight [0.24, 0.48]. Similarly, other scenarios are assigned weights using table 2.

For assigning weights to all scenarios depicted in Figure 1, we are considering weight assignment matrix (table 2).

- Weight for security scenario S1 is equal to “[0.24, 0.48]” and for S2 is “[0.15, 0.37]”.

- Weight for response time scenario R1 is equal to “[0.40, 0.77]” and for R2 is “[0.15, 0.37]”.
- Weight for concurrency scenario C1 is equal to “[0.40, 0.77]” and for C2 is “[0.24, 0.48]”.

Step 3: *Prioritization of concerns* is done by taking the average of the weights of the associated scenarios calculated as in above step 2.

The weight for security concern is $(S1 + S2) / 2 = ([0.24, 0.48] + [0.15, 0.37]) / 2 = [0.19, 0.42]$

The weight for response time concern is $(R1 + R2) / 2 = ([0.40, 0.77] + [0.15, 0.37]) / 2 = [0.27, 0.57]$

The weight for concurrency concern is $(C1 + C2) / 2 = ([0.40, 0.77] + [0.24, 0.48]) / 2 = [0.32, 0.62]$

Step 4: *Concern / Stakeholder matrix for FRIEND as depicted in table 5.*

Step 5: *Weighted Concern / Stakeholder Matrix* is obtained by assigning weights to all candidate aspects (i.e crosscutting concerns) with respect to their importance to each stakeholder involved. See table 6.

Step 6: *Modified Concern / Stakeholder Matrix for FRIEND as depicted in table 7.*

Step 7: *Conflicts resolution using all above steps*

Here, in our case study, we assume that the weights assigned by the stakeholder ‘field officer’ for concerns response time and concurrency are same, which is [0.8, 1.0]. We have also calculated the weights for concerns response time and concurrency, which are [0.27, 0.57] and [0.32, 0.62] respectively. To get the final value Vf for resolving the conflict between the concerns, we multiplied the weights of concerns assigned by the stakeholder to their respective concerns’ calculated weight. For stakeholder “Field officer” the importance of response time concern is more than concurrency so the concern response time will be composed with the requirements associated with the “Field officer”. Similarly, other conflicts are resolved.

Table 5: Concern / Stakeholder matrix for FRIEND

(FO: Field officer, DP: Dispatcher, SA: System analyst, OD: Other databases, RI: Resource information, GI: Geographical information, EOP: Emergency operation plans information, HMI: Hazardous material information)

Stakeholder	FO	DP	SA	LB	OD	RI	GI	EOP	HMI
ReportEmergency	√							√	√
OpenIncident		√					√	√	√
AllocateResource		√				√	√	√	√
ViewMap		√				√	√	√	
Security	√	√	√	√					
Availability	√	√	√		√	√	√	√	√
Response Time	√	√			√	√	√	√	
Accuracy	√	√		√	√	√	√	√	√
Concurrency	√	√	√		√	√	√	√	√
Compatibility	√			√	√				

Table 6: Concern / Stakeholder Matrix with weights to conflicting aspects

(FO: Field officer, DP: Dispatcher, SA: System analyst, OD: Other databases, RI: Resource information, GI: Geographical information, EOP: Emergency operation plans information, HMI: Hazardous material information, RE: Report Emergency, OI: Open Incident, AR: Allocate Resources, VM: View Map, CN: Concurrency, RT: Response Time, AC: Accuracy, CP: Compatibility, AV: Availability, SC: Security)

	FO	DP	SA	LB	OD	RI	GI	EOP	HMI
RE	√							√	√
OI		√					√	√	√
AR		√				√	√	√	√
VM		√				√	√	√	
SC	[0.3,0.5]	[0.3,0.5]	[0.8,1.0]	[0.5,0.8]					
AV	√	√	√		√	√	√	√	√
RT	[0.8,1.0]	[0.8,1.0]			[0.5,0.8]	√	√	√	
AC	√	√		√	√	√	√		√
CN	[0.8,1.0]	[0.8,1.0]	[0.5,0.8]		[0.8,1.0]	√	√	√	√
CM	√			√	√				

Table 7: Modified Concern / Stakeholder Matrix

(FO: Field officer, DP: Dispatcher, SA: System analyst, OD: Other databases, RI: Resource information, GI: Geographical information, EOP: Emergency operation plans information, HMI: Hazardous material information, RE: Report Emergency, OI: Open Incident, AR: Allocate Resources, VM: View Map, CN: Concurrency, RT: Response Time, AC: Accuracy, CP: Compatibility, AV: Availability, SC: Security)

	FO	DP	SA	LB	OD	RI	GI	EOP	HMI
RE									
OI									
AR									
VM									
SC									
AV									
RT	[0.21,0.54]	[0.21,0.54]							
AC									
CN	[0.25,0.59]	[0.25,0.59]							
CM									

5. CONCLUSION

In this paper, we have presented an approach to resolve conflicts in AORE, which is an extension to already existing approach in [1] and [18]. Here, we have applied the concepts of fuzzy logic which is most suitable for representing vague and imprecise data such as human linguistic descriptions. The use of fuzzy logic to conflict resolution is new in its kind that incorporates one domain in other. But, still we need more efforts on this approach to realize it as a valid approach. These efforts include exploring the activity of prioritizing scenarios and validating the approach with more case studies. Our future

work will focus on improving the proposed approach by incorporating all the aspects which are left here.

6. REFERENCE

- [1] Rashid, A., Moreira, A., Araújo, J., "Modularization and Composition of Aspectual Requirements", In 2nd Aspect-Oriented Software Development Conference (AOSD'03), Boston, USA, ACM Press. 11-20, 2003.
- [2] Baniassad, E., Clements, P., Araújo, J., Moreira, A., Rashid, A., Tekinerdogan, B., "Discovering Early Aspects", IEEE Software Special Issue on Aspect-Oriented Programming. 23(1): 61-70, 2006.

- [3] Nuseibeh, B., Easterbrook, S. (2000). Requirements Engineering: A Roadmap. In 22nd International Conference on Software Engineering (ICSE'00), Limerick, Ireland, ACM Press. 35-46.
- [4] Khan, S.S.; Jaffar-ur-Rehman, M., "A survey on early separation of concerns", *Software Engineering Conference*, 2005. APSEC '05. 12th Asia-Pacific, vol., no., pp. 7 pp.-, 15-17 Dec. 2005.
- [5] M. Amroune et al., "A Conflict Resolution Process in AspeCiS Approach", *International Journal of Computer Applications (0975 – 8887)*, Volume 44– No10, April 2012, pg. 14-21.
- [6] Sampaio, A., Greenwood, P., Garcia, A.F.; Rashid, A. 2007. A Comparative Study of Aspect-Oriented Requirements Engineering Approaches, *Empirical Software Engineering and Measurement. ESEM 2007*, 166-175.
- [7] Zadeh, L.A., *Fuzzy Logic* IEEE Computer, pp. 83-93 (1988).
- [8] Yen, J., Tiao, W. (1997). A Systematic Tradeoff Analysis for Conflicting Imprecise Requirements. In 3rd Requirements Engineering Conference (RE'97), Annapolis, USA, IEEE Computer Society. 87-96.
- [9] VanLamsweerde, A., Darimont, R., Letier, E. (1998), "Managing Conflicts in Goal-Driven Requirements Engineering." *IEEE Transactions on Software Engineering*. IEEE Computer Society, 24(11): 908-926.
- [10] Robinson, W., Volkov, S. (1999), Conflict-Oriented Requirements Restructuring. Georgia State University. TR: GSU CIS Working Paper 99-5. <http://cis.gsu.edu/~wrobinso/papers/GSUwp99-5.pdf>.
- [11] Chung, L., Nixon, B., Yu, E., Mylopoulos, J. (2000), *Non-Functional Requirements in Software Engineering*, 0-7923-8666-3, Kluwer Academic Publishers.
- [12] VanLamsweerde, A. (2001), Goal-Oriented Requirements Engineering: A Guided Tour. In 5th Requirements Engineering Conference (RE'01), Toronto, Canada, IEEE Computer Society. 249 - 262.
- [13] Alves, C., Finkelstein, A. (2003), "Investigating Conflicts in COTS Decision-Making" , *International Journal of Software Engineering & Knowledge Engineering*. **13**(5): 1-21.
- [14] M. Ramzan et al., "A genetic algorithms based approach for conflicts resolution in requirement", *International Journal of the Physical Sciences* Vol. 6(4), pp. 828-836, 18 February, 2011.
- [15] Brito, I., Moreira A. (2004). Integrating the NFR approach in a RE model. In Early Aspects Workshop: Aspect-Oriented Requirements Engineering and Architecture Design at 3rd Aspect-Oriented Software Development Conference (AOSD'04), Lancaster, UK. <http://trese.cs.utwente.nl/workshops/early-aspects-2004/Papers/BritoMoreira.pdf>.
- [16] Moreira, A., Rashid, A., Araújo, J. (2005a). Multi-dimensional Separation of Concerns in Requirements Engineering. In 13th Requirements Engineering Conference (RE'05), Paris, France, IEEE Computer Science. 285-296.
- [17] Chitchyan, R., Rashid, A., Rayson, P., Waters, R. (2007a). Semantics-Based Composition for Aspect-Oriented Requirements Engineering. In 6th Aspect-Oriented Software Development Conference (AOSD'07), Vancouver, Canada, ACM Press. 36-48.
- [18] R. W. Waters, "MRAT: A Multidimensional Requirements Analysis Tool", MSc. Dissertation, Lancaster Univ., UK, Oct. 2006.
- [19] F. Hameed and M. Ejaz, "Model for conflict resolution in aspects within Aspect Oriented Requirement engineering", Master Thesis on Software Engineering, School of Engineering, Blekinge Institute of Technology, Sweden, 2008.
- [20] Sardinha, A., Araújo, J., Moreira, A., and Rashid, A. 2010. Conflict Management in Aspect-Oriented Requirements Engineering. *Information Sciences and Technologies Bulletin of the ACM Slovakia, Special Section on Early Aspects*, R.Chitchyan, S. Zsachaler (Eds.), Vol. 2, No. 1 (2010) 56-59.
- [21] L. Bass, P. Clements and R. Kazman, "Software Architecture in Practice", 2nd ed., Addison Wesley. 2003.
- [22] T. Nirmala, D.Datta, H.S.Kushwaha and K.Ganesan, "Inverse Interval Matrix: A New Approach", *Applied Mathematical Sciences*, Vol. 5, 2011, no. 13, 607 – 624.
- [23] N. Singh and N.S. Gill, "The Early Identification of Functional and Non-Functional Crosscutting Concerns", *International Journal of Computer Applications (0975 – 8887)*, Volume 46– No.13, May 2012, pg. 25-32.
- [24] Bernd Bruegge et al., 1994, Design Considerations for an Accident Management System, In Proceedings of the Second International Conference on Cooperative Information Systems, Toronto Press, May 1994.
- [25] B. Bruegge and A.Dutoit, 2007, Object-Oriented Software Engineering Using UML, Patterns, and Java™, 2nd Edition, Pearson Education.