Generalized Conflict and Resolution Model with Approximation Spaces

Sheela Ramanna[1], James F. Peters[2], and Andrzej Skowron[3]

[1] Department of Applied Computer Science,
University of Winnipeg,
Winnipeg, Manitoba R3B 2E9 Canada
s.ramanna@uwinnipeg.ca
[2] Department of Electrical and Computer Engineering,
University of Manitoba
Winnipeg, Manitoba R3T 5V6 Canada
jfpeters@ee.umanitoba.ca
[3] Institute of Mathematics,
Warsaw University
Banacha 2, 02-097 Warsaw, Poland
skowron@mimuw.edu.pl

Abstract. This paper considers the problem of a generalized model for conflict analysis and resolution. Such a model would be helpful in analyzing and resolving conflict in disputes in both government and industry, where disputes and negotiations about various issues are the norm. The issue here is how to model a combination of situations among agents i) where there are disagreements leading to a conflict situation ii) need for an acceptable set of agreements. The solution to this problem stems from pioneering work on this subject by Zdzisław Pawlak, which provides a basis for a generalized model encapsulating a decision system with complex decisions and an approximation space-based conflict resolution using rough coverage. An example of a requirements scope negotiation for an automated lighting system is presented. The contribution of this paper is a rough set based requirements scope determination model using a generalized conflict model with approximation spaces.

Keywords: Approximation space, conflict analysis, conflict resolution, rough sets, requirements engineering, scope negotiation.

1 Introduction

Conflict analysis and resolution play an important role in government and industry where disputes and negotiations about various issues are the norm. To this end, many mathematical formal models of conflict situations have been proposed and studied, e.g., [4,5,6,11,12,15]. The approach used in this paper, is based on a different kind of relationship in the data. This relationship is not a dependency, but a conflict [16]. Formally, a conflict relation can be viewed as a special kind of discernibility, i.e., negation (not necessarily, classical) of indiscernibility relation

S. Greco et al. (Eds.): RSCTC 2006, LNAI 4259, pp. 274–283, 2006.

which is the basis of rough set theory [14]. Thus indiscernibility and conflict are closely related from logical point of view. It is also interesting to note that almost all mathematical models of conflict situations are strongly domain dependent.

Cost effective engineering of complex software systems involves a collaborative process of requirements identification through negotiation. This is one of the key ideas of the Win-Win [1] approach [3] used in requirements engineering. This approach also includes a decision model where a minimal set of conceptual elements, such as win conditions, issues, options and agreements, serves as an agreed upon ontology for collaboration and negotiation defined by the Win-Win process. System requirements (goals) are viewed as conditions. Conflicts arising during system requirements gathering is especially acute due to the nature of the intense collaboration between project stakeholders involved in the process. In particular, determining the scope or the extent of functionality to be developed is crucial.

Recent work in the application of rough sets to handling uncertainty in software engineering can be found in [10,18,19]. However, the basic assumption in all of these papers, is that requirements have already been *decided* and the analysis of gathered requirements data is then performed. This paper extends the earlier work involving the high-level requirements negotiation based on winning conditions [21]. In this paper, the focus is on achieving consensus on detailed set of requirements for each high level requirement that was agreed by all stakeholders. This process is also known as scope negotiation.

The contribution of this paper is a rough set based requirements scope determination model using a generalized conflict model with approximation spaces. Conflict graphs are used to analyze conflict situations, reason about the degree of conflict and explore coalitions. A rough coverage function is used to measure the degree of conformity of sets of similar requirements to negotiation standards. We illustrate our approach in determining scope of a complex engineering system requirements through negotiation.

This paper is organized as follows. An introduction to basic concepts of conflict theory is given Sect. 2. Conflicts, information systems and rough sets are discussed in Sect. 3. A complex conflict model is given in Sect. 4 followed by an illustration of requirements scope negotiation for a home lighting automation system (HLAS) in Sections 5 and 5.1. A generalized conflict model for conflict analysis and conflict resolution considered in the context of approximation spaces is given Sect. 5.2.

2 Basic Concepts of Conflict Theory

The basic concepts of conflict theory that we use in this paper are due to [16]. Let us assume that we are given a finite, non-empty set Ag called the *universe*. Elements of Ag will be referred to as *agents*. Let a *voting function* $v : Ag \rightarrow \{-1, 0, 1\}$, or in short $\{-, 0, +\}$, be a number representing his/her voting result about some issue under negotiation, to be interpreted as *against,neutral*

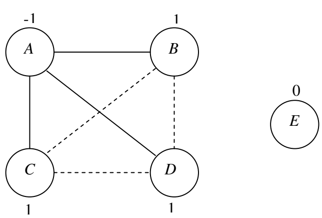
[1] See `http://sunset.usc.edu/research/WINWIN`

and *favorable*, respectively. The pair $CS = (Ag, V)$, where V is a set of voting functions, will be called a *conflict situation*.

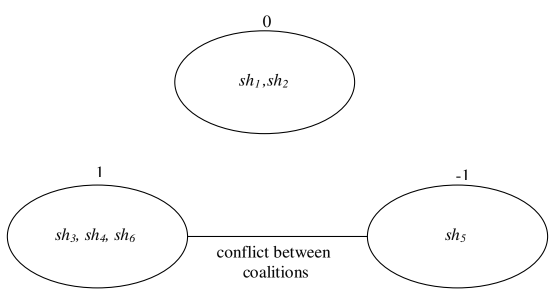
In order to express relations between agents, we define three basic binary relations on the universe: *agreement, neutrality*, and *disagreement*. To this end, for a given voting function v, we first define the following auxiliary function:

$$\phi_v(ag, ag') = \begin{cases} 1, & \text{if } v(ag)v(ag') = 1 \text{ or } v(ag) = v(ag') = 0 \\ 0, & \text{if } v(ag)v(ag') = 0 \text{ and } non(v(ag) = v(ag') = 0) \\ -1, & \text{if } v(ag)v(ag') = -1. \end{cases} \quad (1)$$

This means that, if $\phi_v(ag, ag') = 1$, agents ag and ag' have the same opinion about an issue v (*agree* on issue v); if $\phi_v(ag, ag') = 0$ means that at least one agent ag or ag' has no opinion about an issue v (is *neutral* on v), and if $\phi_v(ag, ag') = -1$, means that both agents have different opinions about an issue v (are in *conflict* on issue v). In what follows, we will define three basic relations R_v^+,R_v^0 and R_v^- on Ag^2 called *agreement, neutrality* and *disagreement* relations respectively, and defined by (i) $R_v^+(ag, ag')$ iff $\phi_v(ag, ag') = 1$; (ii) $R_v^0(ag, ag')$ iff $\phi_v(ag, ag') = 0$; (iii) $R_v^-(ag, ag')$ iff $\phi_v(ag, ag') = -1$. It is easily seen that the *agreement* relation is an *equivalence* relation. Each equivalence class of the agreement relation will be called a *coalition* with respect to v. For the conflict or disagreement relation we have: (i) not $R_v^-(ag, ag)$; (ii) if $R_v^-(ag, ag')$ then $R_v^-(ag', ag)$; (iii) if $R_v^-(ag, ag')$ and $R_v^+(ag', ag'')$ then $R_v^-(ag, ag'')$. For the neutrality relation we have: (i) not $R_v^0(ag, ag)$; (ii) $R_v^0(ag, ag') = R_v^0(ag', ag)$. In the conflict and neutrality relations there are no coalitions. In addition, $R_v^+ \cup R_v^0 \cup R_v^- = Ag^2$. All the three relations R_v^+, R_v^0 , R_v^- are pairwise disjoint. With every conflict situation $cs = (Ag, v)$ relative to a voting function v, we will associate a *conflict graph* CG_v. Examples of conflict graphs are shown in Figure 1. In Figure 1(a), solid lines denote conflicts, dotted line denote agreements, and for simplicity, neutrality is not shown explicitly in the graph.



(a) Exemplary Conflict



(b) Requirement Conflict (see $r1.1$ in Table 2)

Fig. 1. Sample Conflict Graphs

As one can see B, C, and D form a coalition. A conflict degree $Con(cs)$ (or $Con_v(cs)$) of the conflict situation $cs = (Ag, v)$ (we write also CS_v instead of cs) is defined by

$$Con(cs) = \frac{\sum_{\{(ag,ag'):\ \phi_v(ag,ag')=-1\}} |\phi_v(ag,ag')|}{2\lceil\frac{n}{2}\rceil \times (n - \lceil\frac{n}{2}\rceil)} \tag{2}$$

where $n = Card(ag)$. Observe that $Con(cs)$ is a measure of discernibility between agents from Ag relative to the voting function v. For a more general conflict situation $CS = (Ag, V)$ where $V = \{v_1, \ldots, v_k\}$ is a finite set of voting functions each for a different issues the *conflict degree* in CS (*tension generated by* V) can be defined by

$$Con(CS) = \sum_{i=1}^{k} Con(cs_i)/k \tag{3}$$

where $cs_i = (Ag, v_i)$ for $i = 1, \ldots, k$.

3 Conflicts, Information Systems, and Rough Sets

There are strong relationships between the approach to conflicts and information systems as well as rough sets. In this section, we discuss examples of such relationships. The presented approach in this section seems to be promising for solving problems related to conflict resolution and negotiations (see, e.g.,[24]).

An information system is a table of rows which are labeled by *objects* (*agents*), columns by *attributes* (*issues*) and entries of the table are *values of attributes* (*votes*), which are uniquely assigned to each agent and attribute, i.e. each entry corresponding to row x and column a represents opinion of an agent x about issue a. Formally, an *information system* can be defined as a pair $S = (U, A)$, where U is a nonempty, finite set called the *universe*; elements of U will be called *objects* and A is a nonempty, finite set of *attributes* [14]. Every attribute $a \in A$ is a total function $a : U \to V_a$, where V_a is the set of *values* of a, called the *domain* of a; elements of V_a will be referred to as *opinions*, and $a(x)$ is opinion of agent x about issue a. The above given definition is general, but for conflict analysis we will need its simplified version, where the domain of each attribute is restricted to three values only, i.e. $V_a = \{-1, 0, 1\}$, for every a, meaning *disagreement, neutral* and *agreement* respectively. For the sake of simplicity we will assume $V_a = \{-, 0, +\}$. Every information system with the above mentioned restriction will be referred to as a *situation*.

We now observe that any conflict situation $CS = (Ag, V)$ can be treated as an information system where $Ag = \{ag_1, \ldots, ag_n\}$ and $V = \{v_1, \ldots, v_k\}$ with the set of objects Ag (*agents*) and the set V of attributes (*issues*).

The discernibility degree between agents ag and ag' in CS can be defined by

$$disc_{CS}(ag, ag') = \frac{\sum_{\{i:\ \phi_{v_i}(ag,ag')=-1\}} |\phi_{v_i}(ag,ag')|}{k}, \tag{4}$$

where $ag, ag' \in Ag$. Now, one can consider reducts of CS relative to the discernibility degree defined by $disc_{CS}$. For example, one can consider agents ag, ag' as discernible if

$$disc_{CS}(ag, ag') \geq tr,$$

where tr a given threshold.[2] Any reduct $R \subseteq V$ of CS is a minimal set of voting functions preserving all discernibility in voting between agents that are at least equal to tr. All voting functions from $V - R$ are dispensable with respect to preserving such discernibility between objects. In an analogous way, one can consider reducts of the information system CS^T with the universe of objects equal to $\{v_1, \ldots, v_k\}$ and attributes defined by agents and voting functions by $ag(v) = v(ag)$ for $ag \in Ag$ and $v \in V$. The discernibility between voting functions can be defined, e.g., by

$$disc_{CS^T}(v, v') = |Con(CS_v) - Con(CS_{v'}|, \tag{5}$$

and makes it possible to measure the difference between voting functions v and v', respectively. Any reduct R of CS^T is a minimal set of agents that preserves the differences between voting functions that are at least equal to a given threshold tr.

4 Complex Conflict Model

In this section, we present an extension of the conflict model and we outline an approach to conflict resolution based on such a model. We assume that agents in the complex conflict model are represented by conflict situations $cs = (Ag, v)$, where Ag is the set of lower level agents and v is a voting function defined on Ag for $v \in V$. Hence, agents in the complex conflict model are related to groups of lower level agents linked by a voting function. The voting functions in the complex conflict models are defined on such conflict situations. The set of the voting functions for the complex conflict model is denoted by A. In this, way we obtain an information system (U, A), where U is the set of situations. Observe that any situation $cs = (Ag, v)$ can be represented by a matrix

$$[v(ag)]_{ag \in Ag}, \tag{6}$$

where $v(ag)$ is the result of voting by the agent $ag \in Ag$. We can extend the information system (U, A) to the decision system (U, A, d) assuming, that $d(cs) = Con_v(cs)$ for any $cs = (Ag, v)$. For the constructed decision system (U, A, d) one can use, e.g., the above function(2) to measure the discernibility between compound decision values which correspond to conflict situations in the constructed decision table. The reducts of this decision table relative to decision have a natural interpretation with respect to conflicts. An illustration of conflict analysis with similarity relation can be found in [22].

5 Systems Requirements Identification and Negotiation

A typical system requirements engineering process leads to conflicts between project stakeholders. A stakeholder is one who has a share or an interest in the

[2] To compute such reducts one can follow a method presented in [23] assuming that any entry of the discernibility matrix corresponding to (ag, ag') with $disc_{CS}(ag, ag') < tr$ is empty and the remaining entries are families of all subsets of V on which the discernibility between (ag, ag') is at least equal to tr [6].

requirements for a systems engineering project. Let Ag be represented by the set SH (stakeholders). Let V denote the set of requirements. Let $CS = (SH, V)$ where $SH = \{sh_1, \ldots, sh_n\}$ and $V = \{v_1, \ldots, v_k\}$. A complete example of the problem of achieving agreement on high-level system requirements for a home lighting automation system described in [9] can be found in [21]. In this paper, the focus is on achieving consensus on detailed set of requirements for each high level requirement that was agreed by all stakeholders. This is a crucial step as it determines the scope of the project. In other words, the team needs to know the *extent* of functionality that needs to be implemented. In fact, no serious requirements analysis can begin until the scope of project has been determined.

5.1 Example: Determining Scope of System Requirements

As a part of scope negotiation, several parameters need to be determined: level of effort, importance of a requirement, stability, risk, testability to name a few. In this paper, we consider the following negotiation parameters: *Level of Effort* which is a rough estimate of development effort (High, Medium, Low), *Importance* which determines whether a requirement is essential to the project (High, Medium, Low), *Stability* of a requirement which indicates its volatility (Yes, Perhaps, No), *Risk* which indicates whether the requirement is technically achievable (High, Medium, Low) *Testability* indicating whether a requirement is testable (Yes, No). Let $R1$, E, I, S, R, T denote requirement 1, Effort, Importance, Stability, Risk, and Testability, respectively. Specifically, the example illustrates the high level functionality(R1) of Custom Lighting Scene[21] to be included in release of V1.0 of HLAS System. The negotiation parameter values (attributes) assessed by the development team for R1 are given in Table 1.

Table 1. Scope Negotiation

Negotiation Parameters						
$R1$	E	I	S	R	T	$Conflict\ Degree$
$r1.1$	M	H	N	L	Y	L
$r1.2$	M	H	N	L	Y	M
$r1.3$	H	M	N	M	Y	L
$r1.4$	L	H	Y	L	Y	L
$r1.5$	M	L	P	H	Y	M
$r1.6$	L	H	Y	H	N	H

Assume that R1 includes the following specifications (objects): $r1.1$ - ability to control up to a maximum of 20 custom lighting scenes throughout the residence, $r1.2$ - each scene provides a preset level of illumination (max. of 3) for each lighting bank, $r1.3$ - maximum range of a scene is 20 meters, $r1.4$ - activated using Control Switch, $r1.5$ - activated using Central Control Unit, and $r1.6$ - Ability to control an additional 2 lighting scenes in the yard. The decision attribute is a compound decision denoting the conflict degree which is a result of a matrix given in Table 2.

The voting results of the members drawn from a stakeholders list SH is given in Table 2. The stakeholder list is comprised of builders, electrical contractors and the marketers. Every stakeholder votes on each of the requirements. An algorithm for determining win agreements can be found in [21]. The conflict graph $CG_{r1.1} = (SH, r1.1)$ can be presented in a simplified form as a graph with nodes represented by coalitions and edges representing conflicts between coalitions as shown in Fig. 1(b). From this graph, one can compute the conflict degree using using Eqn. 2 where $Con_{r1.1}(cs) = 0.3$. The degree of conflict for the remaining requirements are $Con_{r1.2}(cs) = 0.44$, $Con_{r1.3}(cs) = 0.2$, $Con_{r1.4}(cs) = 0$, $Con_{r1.5}(cs) = 0.67$, and $Con_{r1.6}(cs) = 0.89$.

Table 2. Voting Results for R1

	Voting Results					
Stakeholder	$r1.1$	$r1.2$	$r1.3$	$r1.4$	$r1.5$	$r1.6$
sh_1	0	1	-1	0	-1	-1
sh_2	0	1	0	0	-1	-1
sh_3	1	-1	0	1	1	-1
sh_4	1	1	0	1	1	-1
sh_5	-1	0	1	1	1	1
sh_6	1	1	-1	1	0	1

5.2 Generalized Conflict Model with Approximation Spaces

This section introduces a generalized model for conflict analysis and conflict resolution that combines earlier work on modeling conflict with approximation spaces. Conflict degree $Con_v(cs)$ for any $cs = (Ag, v)$ plays the role of a decision in Table 1. $Con_v(cs)$ is a subjective value that is a result of voting with the following levels: L (*conflict degree* ≤ 0.3), M ($0.3 \leq$ *conflict degree* ≤ 0.7) and H ($conflict\ degree > 0.7$). For example, to determine which requirements should be included in product release version V1.0, negotiation must occur at two levels, namely, voting and table. At the voting level (lower level), the basic conflict model is used. At the decision table level, conflicts are evaluated within an approximation space [20], which is the approach used in [17].

Let $DS = (U_{req}, A, d)$, where U_{req}, A, d denote a non-empty set of requirements, a non-empty set of scope negotiation parameters, and an estimated degree of conflict, respectively (see Table 1). Let D_i denote the i^{th} decision, i.e., $D_i = \{u \in U_{req} : d(u) = i\}$, which is set of requirements from U_{reg} with conflict level i. For any boolean combination of descriptors over DS and α, the semantics of α in DS is denoted by $\|\alpha\|_{DS}$, i.e., the set of all objects from U satisfying α [14].

In what follows, $i = L$ and D_L denotes a decision class representing a low degree of conflict between stakeholders. Now, we can define a generalized approximation space $GAS = (U_{req}, N_B, \nu_B)$, where for any objects $r \in U$ the neighborhood $N_B(r)$ is defined by

$$N_B(r) = \| \bigwedge_{a \in B} (a = a(r)) \|_{DS}, \tag{7}$$

and the coverage function ν_B is defined by

$$\nu_B(X, Y) = \begin{cases} \frac{|X \cap Y|}{|Y|}, & \text{if } Y \neq \emptyset, \\ 1, & \text{if } Y = \emptyset, \end{cases} \tag{8}$$

where $X, Y \subseteq U$. This form of specialization of a GAS is called a *lower approximation space* [17]. Assuming that the lower approximation $B_* D_i$ represents an acceptable (standard) level of conflict during negotiation, we are interested in the values

$$\nu_B(N_B(r), B_* D_L), \tag{9}$$

of the coverage function specialized in the context of a decision system DS for the neighborhoods $N_B(r)$ and the standard $B_* D_L$ for conflict negotiation.

Computing the rough coverage value for negotiation parameters extracted from a table such as Table 1 implicitly measures the extent to which standards for release parameters have been followed. What follows is a simple example of how to set up a lower approximation space relative to a scope negotiation table based on the $effort$ (E), *risk* (R) and *testability* (T) paramters:

$B = \{E,\ R,\ T\}$, $D_L = \{r \in U : d(r) = L\} = \{r1.1, r1.3, r1.4\}$,
$B_* D_L = \{r1.3, r1.4\}$, $N_B(r1.1) = \{r1.1, r1.2\}$, $N_B(r1.3) = \{r1.3\}$,
$N_B(r1.4) = \{r1.4\}$, $N_B(r1.5) = \{r1.5\}$, $N_B(r1.6) = \{r1.6\}$,
$\nu_B(N_B(r1.1), B_* D_L) = 0$, $\nu_B(N_B(r1.3), B_* D_L) = 0.5$, $\nu_B(N_B(r1.4), B_* D_L) = 0.5$, $\nu_B(N_B(r1.5), B_* D_L) = 0$, $\nu_B(N_B(r1.6), B_* D_L) = 0$.

Based on the experimental rough coverage values, we can set a threshold for acceptance. In this case, let $\nu_B \geq 0.5$. Consequently, for the first release version V1.0 of the $HLAS$, requirements r1.3 and r1.4 would be included. On the other hand, if negotiation parameters $effort$ (E), *importance* (I) and *stability* (S) are considered, with the same threshold of acceptance, only requirement r1.3 would be included in release version V1.0 as the following calculations show:

$B = \{E,\ I,\ S\}$, $D_L = \{r \in U : d(r) = L\} = \{r1.1, r1.3, r1.4\}$, $B_* D_L = \{r1.3\}$
$N_B(r1.1) = \{r1.1, r1.2\}$, $N_B(r1.3) = \{r1.3\}$, $N_B(r1.4) = \{r1.4, r1.6\}$,
$N_B(r1.5) = \{r1.5\}$,
$\nu_B(N_B(r1.1), B_* D_L) = 0$, $\nu_B(N_B(r1.3), B_* D_L) = 1$, $\nu_B(N_B(r1.4), B_* D_L) = 0$, $\nu_B(N_B(r1.5), B_* D_L) = 0$.

The proposed attempt to assess negotiation offers deeper insight into conflict dynamics whereby one can observe changes to the requirements set based on coverage and level of conflict when different negotiation parameters are selected.

The risk analysis can also be performed using reducts preserving conflicts to a degree and their approximations. It should also be noted that the conflict model discussed in Sect. 4 is more general where any situation will correspond to a set of stakeholders. Analogous to the approach in Sect. 4, we can compute for any such situations conflict degrees and continue with the conflict resolution process. In this situation, we can consider issues for negotiation relative to groups of stakeholders.

6 Conclusion

This paper introduces rough set-based requirements scope determination using a generalized conflict model with approximation spaces. In other words, the generalized conflict model with approximation spaces provides the ability to (i) define a level of conflict that is acceptable, (ii) determine the equivalent set of requirements based on a specified set of negotiation parameters tailored to a specific project, and (iii) select requirements to be included in the system with a measure indicating the extent that standards for release parameters have been followed. The application of rough sets can bring new results in the area related to conflict resolution and negotiations between agents because this make it possible to introduce approximate reasoning about vague concepts into the area.

Acknowledgments. The authors gratefully acknowledge suggestions by Zdzisław Pawlak about conflict graphs. Authors would like to thank Gianpiero Cattaneo for correcting the previous version of the formula (1). The research of Andrzej Skowron, James F. Peters, and Sheela Ramanna is supported by grant 3 T11C 002 26 from the Ministry of Scientific Research and Information Technology of the Republic of Poland, NSERC Canada grants 194376 and 185986, respectively.

References

1. Alpigini, J. J., Peters, J. F., Skowron, A., Zhong, N. (eds.): Third International Conference on Rough Sets and Current Trends in Computing(RSCTC'2002). LNAI, Vol. 2475. Springer-Verlag, Heidelberg Germany (2002)
2. Bazan, J., Osmólski, A., Skowron, A., Ślęzak, D., Szczuka, M., Wróblewski, J.: Rough set approach to the survival analysis. In: Alpigini et al. [1], 522–529.
3. Boehm, B., Grünbacher, P., Kepler, J.: Developing Groupware for Requirements Negotiation: Lessons Learned. IEEE Software, (May/June 2001) 46-55.
4. Casti, J. L. Alternative Realities – Mathematical Models of Nature and Man. John Wiley and Sons (1989)
5. Coombs, C. H., Avrunin, G. S. : The Structure of Conflict. Lawrence Erlbaum Associates (1988)
6. Deja, R., Skowron, A.: On Some Conflict Models and Conflict Resolutions, *Romanian Journal of Information Science and Technology*, 5(1-2) (2002) 69-82.
7. Dunin-Kęplicz, B., Jankowski, A., Skowron, A., Szczuka, M. (eds.): Monitoring, Security, and Rescue Tasks in Multiagent Systems (MSRAS'2004). Advances in Soft Computing. Springer-Verlag, Heidelberg Germany (2005)

8. Easterbrook, S.: Handling Conflict between Domain Descriptions with Computer-Supported Negotiation. *Knowledge Acquisition: An International Journal*, Vol. 3 (1991) 255-289.
9. Leffingwell, D., Widrig, D.: Managing Software Requirements. Addision-Wesley. Boston U.S.A (2003)
10. Li, Z., Ruhe, G.: Uncertainty handling in Tabular-Based Requirements Using Rough Sets. In: Ślęzak, D., Yao, J.T., Peters, J.F., Ziarko, W., Hu, X (eds.): Rough Sets, Fuzzy Sets, Data Mining and Granular Computing. LNAI, Vol. 3642. Springer, Heidelberg Germany (2005) 678-687.
11. Maeda, Y., Senoo, K., Tanaka, H.: Interval density function in conflict analysis. LNAI, Vol. 1711. Springer-Verlag, Heidelberg Germany (1999) 382-389.
12. Nakamura, A.: Conflict logic with degrees. In: Pal, S. K., Skowron. A. (eds.): Rough Fuzzy Hybridization: A New Trend in Decision-Making. Springer-Verlag, Heidelberg Germany (1999) 136-150.
13. Pawlak, Z.: On Conflicts. *Int. J. of Man-Machine Studies*,Vol. 21 (1984) 127-134.
14. Pawlak, Z.: Rough Sets – Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers (1991)
15. Pawlak, Z.: An inquiry into anatomy of conflicts. *Journal of Information Sciences*, Vol. 109 (1998) 65-78.
16. Pawlak, Z., Skowron, A.: Rough Sets and Conflict Analysis. In: E-Service Intelligence – Methodologies, Technologies and Applications Book series on Computational Intelligence. Springer-Verlag, Berlin Germany (2006) [to appear]
17. Peters, J. F., Henry, C.: Reinforcement Learning with Approximation Spaces. *Fundamenta Informaticae* 71 (2006) 323-349.
18. Ramanna, S.: Approximation Space for Software Models. In: Peters, J.F., Skowron, A. (eds.): *Transactions on Rough Sets I.* LNCS, Vol. 3100. Springer-Verlag, Berlin Germany (2004) 338-354.
19. Peters, J. F., Ramanna, S.: Towards a software change classification system: A rough set approach. *Software Quality Journal*, Vol. 11 (2003) 121-147.
20. Skowron, A., Stepaniuk, J.:Generalized Approximation Spaces. In: Lin, T.Y., Wildberger, A.M. (eds.): Soft Computing. Simulation Councils, San Diego (1995)18-21; see also Tolerance Approximation Spaces, *Fundamenta Informaticae*, Vol. 27(2-3) (1996) 245-253.
21. Ramanna, S., Peters, J.F., Skowron, A.: Conflict Analysis and Information Systems: A Rough Set Approach. In: Wang, G., Peters, J.F., Skowron, A.. Yao, Y.Y (eds.): Proceedings of the International Conference on Rough Sets and Knowledge Technology (RSKT 2006), LNAI, Chongqing, China, July 24-26 (2006) [to appear].
22. Ramanna, S., Peters, J.F., Skowron, A.: Analysis of Conflict Dynamics by Risk Patterns. Proceedings of the Workshop on Concurrency, Specification and Programming (CS&P 2006), Germany, September 27-29 (2006) [to appear].
23. Skowron, A. Rauszer, C.: The discernibility matrices and functions in information systems. In: Słowiński, R (ed.): *Intelligent Decision Support - Handbook of Applications and Advances of the Rough Sets Theory, System Theory, Knowledge Engineering and Problem Solving*, Vol. 11 (1992) 331-362.
24. Kowalski, R.: A logic-based approach to conflict resolution (2003) 1-28 (manuscript).