Proceedings of the 2007 IEEE Symposium on
Foundations of Computational Intelligence (FOCI 2007)

Requirements Interaction and Conflicts: A Rough Set Approach

Sheela Ramanna
Department of Applied Computer Science
University of Winnipeg
Winnipeg, Manitoba R3B 2E9, Canada
Email: s.ramanna@uwinnipeg.ca

Andrzej Skowron
Institute of Mathematics
Warsaw University
Banacha 2, 02-097 Warsaw, Poland
Email: skowron@mimuw.edu.pl

Abstract— **Rough set theory and approximation spaces introduced by Zdzisław Pawlak provide frameworks for modeling interaction amongst requirements. These interactions can be viewed as conflicts. The problem here is how to model a combination of complex situations where there are social conflicts (due to differing stakeholder views) and technical conflicts (due to inconsistent requirements). These two sources of conflicts are intertwined. That is, inconsistent requirements often reflect the inconsistent need of stakeholders. The solution consists of a unified requirements interaction framework that simplifies the representation of technical (requirements inconsistency) and social conflicts (stakeholder views). The contribution of this paper is a new *socio-technical* model. The model makes it possible to represent conflict degrees and trace dependency information. An illustrative example of such a framework is presented. Reasoning about conflict dynamics is made possible by risk patterns.**

I. INTRODUCTION

Requirements Engineering provides the appropriate mechanism for understanding what the customer wants, analyzing the need, negotiating a reasonable solution, specifying the solution unambiguously, validating the specification, and managing the requirements as they are transformed into an operational system [24]. Consequently, requirements engineering research spans a wide range topics, but a topic of increasing importance is the analysis and management of dependencies (relationships) among requirements also known as Requirements Interaction Management (RIM) [23]. Understanding and representing requirements conflict is one of the objectives of RIM. Rough set theory and approximation spaces introduced by Zdzisław Pawlak provide frameworks for modeling $socio-technical$ interaction amongst requirements. These interactions can be viewed as conflicts. In particular, development of complex software systems involves a collaborative process of requirements identification through negotiation. Conflicts arising during this negotiation process are especially acute due to the nature of the intense collaboration between project stakeholders. The problem here is how to model a combination of complex situations as a result of requirements interaction where there are social conflicts (due to differing stakeholder views) and technical conflicts (due to inconsistent requirements). The solution is a rough-set based requirements interaction framework that facilitates assessment of conflict dynamics as a means of achieving consensus regarding the scope of system functionality that needs to be developed. The basic architecture for requirements interaction framework is shown in Fig. 1.

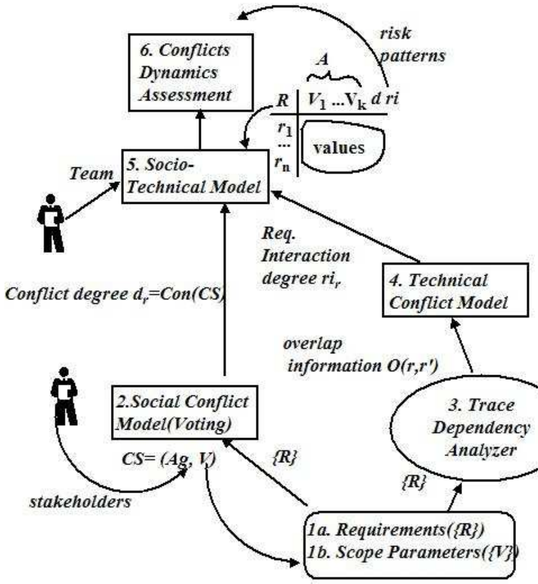

Fig. 1. Requirements Interaction Framework

This paper is organized as follows. In Sect. II, we introduce the architecture for a requirements interaction framework which unifies technical and social conflicts and describe the steps for setting up the model. We present the formal model for social conflicts in Sect. III. This is followed by a discussion of a model for technical conflicts in Sect. IV. We introduce a socio-technical model in the context of an example for a home lighting automation system in Sect. V. Conflict dynamics assessment with discernibility degree and risk patterns are illustrated in Sect. VI. A brief discussion of related works is given in Sect. VII.

1-4244-0703-6/07/$20.00 ©2007 IEEE

II. REQUIREMENTS INTERACTION FRAMEWORK

In Fig. 1, the $first\ step$ is to identify requirements R and also to decide on parameters V. The parameters form a checklist of questions that need to be answered as a part of the review of each requirement [14]. These are: level of effort, importance of a requirement, stability, risk, testability to name a few. The requirements and parameters (also called issues) become input to the social conflict model. The basic concepts of conflict theory upon which the social conflict model is built is due to [9]. This model represents subjective opinions about issues (parameters) of stakeholders of each requirement in the form of votes. As a result, one can compute degree of conflict amongst stakeholders, explore coalitions and measure discernibility. In other words, the social model encapsulates conflict situations and facilitates win agreements about requirements. Thus, the $second\ step$ involves computation of a conflict degree d for each requirement. The $third\ step$ involves the determination of degree of overlap between requirements based on trace dependency information. In this paper, we assume that this information is obtained by an requirements management tools such as those listed in[1]. In theory, steps 2 and 3 can be performed in parallel. However, in practice, it is better to get agreements between stakeholders about the specific requirements before exploring inconsistencies between them. The technical model in ($step\ four$) captures the degree of overlap, type of requirements and the extent to which requirements conflict or cooperate. At this time, this model is used as a look-up table. The output of $step\ four$ is the requirements interaction degree ri. The socio-technical model is then constructed which is a decision table [11] and elaborated in [10], [12], [13]. The construction of the decision table in $step\ five$ is rather involved. In addition to d and ri, project teams need to estimate the values for A shown in Fig. 1. The socio-technical model is comprehensive in that it takes into account objective and subjective estimates. In $step\ six$, conflict dynamics is assessed. In this paper, the focus is on an analysis of the deviations to conflict degrees using risk patterns. A distance function used to derive risk patterns represents an acceptable the level of deviation (risk) for a project.

III. SOCIAL CONFLICT MODEL

In a rough set approach to conflict analysis, an information system is represented by a table containing rows labeled by *objects* (*agents*), columns by *attributes* (*issues*). The entries of the table are *values of attributes* (*votes*), which are uniquely assigned to each agent and attribute, i.e. each entry corresponds to a row x and column a representing 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 are called *objects*) and A is a nonempty, finite set of *attributes* [11]. Every attribute $a \in A$ is a partial function $a : U \rightarrow V_a$, 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 the opinion of an agent x about issue a. Although the definition given above is general, 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 restriction mentioned above 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*). A conflict degree $Con(CS)$ of the conflict situation $CS = (Ag, v)$ is defined by

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

where $n = Card(Ag)$. 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 (2).

$$Con(CS) = \frac{\sum_{i=1}^{k} Con(CS_i)}{k}, \qquad (2)$$

In the social conflicts model, $d_r = Con(CS)$ for any $CS = (Ag, V)$ for each $r \in R$. That is, there is a decision for each requirement. As a result, the universe Ag will now consist of SH(set of stakeholders) and the *voting function* $v : SH \rightarrow \{-1, 0, 1\}$, which gives a number representing the voting result about some issue about requirements under negotiation. Formally a social conflict model is $SCM = (SH, V)$ where $SH = \{sh_1, \ldots, sh_n\}$ and $V = \{v_1, \ldots, v_k\}$ where V denotes a finite set of voting functions for scope negotiation parameters. Note the notation V has been used interchangeably (for scope negotiation parameters) in Fig. 1 as well as for voting functions. The conflict situation can now be interpreted as opinions held by two or more stakeholders about requirements that cause an inconsistency. The model for social conflicts has been used to achieve consensus on the high-level requirements (for details, see [15]).

IV. TECHNICAL CONFLICT MODEL

Technical conflicts arise due to contradictory specifications of requirements. Requirements *conflict* with each other when they make contradictory statements about common software attributes, and they *cooperate* when they mutually enforce such attributes [4]. These software attributes are generally referred to, as persistent or non-functional attributes and include quality attributes such as efficiency, reliability, scalability, usability, security to name a few. Formally, a technical conflict model is a *decision system* defined as $TCM = (U, B, ri)$, where U is a nonempty, finite set called the *universe* (elements of U are called *requirements*) and B is a nonempty, finite set of *conflict attributes* and ri represents requirement interaction degree with the restriction $B =$

---

[1] See `http:/www.incose.org`

$\{Type, Degree\_of\_Overlap, Artifact\}$. Every attribute $a \in B$ is a total function $a : U \rightarrow V_a$, where V_a is the set of *values* of a, called the *domain* of a. Although the above given definition is general, for technical conflicts, we also need to restrict the domain of each attribute as follows:

- $V_{Type} = \{FR, ER, UR, RR, SR, RCR, AR, MR\}$, meaning functionality, efficiency, usability, reliability, security, recoverability, accuracy and maintainability respectively indicating the type of requirement
- $V_{DegreeofOverlap} = [0, 1]$,
- $V_{Artifact} = \{R_1, \ldots, R_k\}$,
- $V_{ri} = \{SC, WC, VWC, NC\}$, represents the degree of conflict: strong conflict, weak conflict, very weak conflict and no conflict respectively.

TCM is designed to capture information from requirements traceability[2]. Requirements traceability involves defining and maintaining relationships with artifacts created as a part of systems development such as architectural designs, requirements, source code to name a few. In this paper, we restrict the $artifact\ information$ to other requirements as an aid to identifying conflicting (or cooperating) requirements [3]. The exemplary domain for degree of overlap and conflict degrees are due to [4]. In this paper, we assume that an automated requirements traceability tool makes it possible to automatically extract i) conflicts and cooperation information amongst requirements and ii) trace dependencies. The degree of overlap between requirements and the conflict degrees are to a large extent manually assessed. With TCM one can assess requirements interaction ri_r for each $r \in R$.

V. ILLUSTRATION:SOCIO-TECHNICAL MODEL

In this section, we will illustrate the construction of a socio-technical model with a set of detailed requirements for a single high-level requirement (R1- Custom Lighting Scenes) for a home lighting automation system (HLAS) described in [6]. A complete example of the problem of achieving agreement on high-level system requirements for a home lighting automation system (HLAS) described in [6] can be found in [15]. Assume that R1 includes the following specifications (objects):

- $r1.1$ - ability to control up to a maximum of 20 custom lighting scenes,
- $r1.2$ - each scene provides a preset level of illumination (within 5 seconds) for each lighting bank,
- $r1.3$ - maximum range of a scene is 20 meters,
- $r1.4$ - activated using Control Switch only,
- $r1.5$ - activated within 3 seconds using Central Control Unit,
- $r1.6$ - Ability to control an additional 2 lighting scenes in the yard.

We consider the following negotiation parameters V with the domain for each parameter as follows:

- $Effort$ 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).

Table I represents the Socio-Technical Model with two decision attributes d and ri representing the sources for the two conflicts.

TABLE I
SOCIO-TECHNICAL MODEL

$R1$	E	I	S	R	T	d	ri
$r1.1$	M	H	N	L	Y	0.22	WC
$r1.2$	M	H	N	L	Y	0.44	VWC
$r1.3$	H	M	N	M	Y	0.2	NC
$r1.4$	L	H	Y	L	Y	0	SC
$r1.5$	M	H	P	H	Y	0.67	WC
$r1.6$	M	L	P	H	N	0.89	VWC

Table II represents a partial SCM for requirement R1. The table is meant to give the reader an illustration of a conflict situation for just for individual requirement r1.1 and for a subset of negotiation parameters V. We can now compute the conflict degree for $CS_{r1.1} = (SH, V)$ using Eqn. 2 where $V = \{Priority, Effort, Risk\}$, $Con(CS_{r1.1})$ = 0.22 with $Con((SH, Priority))$ = 0, $Con((SH, Effort))$ = 0.33 and $Con((SH, Risk))$ = 0.33. Note that $+$ indicates the highest level of support, $-$ indicates the lowest level of support and 0 indicates the intermediate level of support. For instance, for the issue *Priority*, $+$ means Critical, $-$ means Useful and 0 means Important. Voting for the remaining requirements $r1.2 \ldots r1.6$ is performed in a similar manner.

TABLE II
REQUIREMENT $r1.1$ - SOCIAL CONFLICT MODEL

$Stakeholder$	$Priority$	$Effort$	$Risk$
sh_1	+	+	0
sh_2	0	-	0
sh_3	+	0	-
sh_4	+	+	+
sh_5	+	0	-

Table III represents a TCM for requirement R1. The assessment of interaction degree follows the approach specified in [4]. Briefly, the approach is based on a generic model of potential conflict and cooperation which highlights the nature of added requirements on other attributes of the system. For example, if a requirement adds $new\ functionality$ to the system, it may have i) no effect (0) on the overall $functionality$ ii) negative effect (-) on $efficiency$ iii) positive effect (+) on $usability$ iv) negative effect (-) on $reliability$ v) negative effect (-) on $security$ vi) no effect(0)on $recoverability$ vii) no effect(0) on $accuracy$ and viii) no effect(0) on $maintainability$. This model is very general

---

[2] IEEE Std. 830-1998

and is a worst best-case scenario. In practice, one must take into account the degree of overlap between requirements and the type of requirement since it has a direct bearing on the degree of conflict or cooperation. Trace dependencies based on scenarios and observations are used to arrive at the degree of overlap [3]. Table III in its current form is used as a look-up table since degree of overlap and the determination of degree of conflict are assessed in a semi-automated manner.

TABLE III

TECHNICAL CONFLICT MODEL

$R1$	$Type$	$Degree\_of\_Overlap$	$Artifact$	ri
$r1.1$	FR	0.7	r1.6	WC
$r1.2$	ER	0.6	r1.5	VWC
$r1.3$	FR	0	-	NC
$r1.4$	FR	0.8	r1.5	SC
$r1.5$	ER	1	r1.4	WC
$r1.6$	FR	0.5	r1.1	VWC

VI. CONFLICT DYNAMICS ASSESSMENT

A. Discernibility Degree

The discernibility degree between agents ag and ag' in CS can be defined by (3).

$$disc_{CS}(ag, ag') = \frac{\sum_{\{i:\ \phi_{v_i}(ag,ag')=-1\}} |\phi_{v_i}(ag, ag')|}{k}, \quad (3)$$

where $ag, ag' \in Ag$. Now, one can consider reducts of the SCM 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 is a given threshold.[3] 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 (4).

$$disc_{CS^T}(v, v') = |Con(CS_v) - Con(CS_{v'})|, \quad (4)$$

which 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. For example, one can observe from Table II that parameters priority and effort are 0.35-indiscernible since $disc_{CS^T}(v, v') = 0.33$. In practice, measuring $disc_{CS}(ag, ag')$ and/or $disc_{CS^T}(v, v')$ could be very useful. In particular, if the stakeholder group is rather large, this measure can have a beneficial effect of shrinking the group size.

B. Risk patterns

Risk patterns are defined by specific reducts (and their approximations) of the socio-technical model. In this section, we assume that the distance between conflict degrees is defined by equation (4).

Now, one can consider reducts of this decision table relative to a fixed distance δ between decision values. Such reducts are called the distance reducts [22]. Let $DT = (U, A, d)$ be a (consistent) decision table [11] and let δ be a distance function between decisions from V_d. Any minimal set $B \subseteq A$ satisfying condition (5).

$$\delta(d(x), d(y)) \geq tr \wedge non(xIND(A)y) \longrightarrow non(xIND(B)y) \quad (5)$$

where $IND(A), IND(B)$ are the indiscernibility relations relative to A, B respectively [11] and tr is a given threshold is called (d, tr)-reduct of DT.

One can use an approach presented in [16] and define modified discernibility reducts making it possible to compute such reducts using Boolean reasoning method. Any such reduct B defines a set of risk patterns. They are obtained by taking the values of attributes from B on any object x from DT, i.e.,

$$\bigwedge_{a \in B} (a = a(x)). \quad (6)$$

From the distance reduct definition, the deviation of the decision on the set of objects satisfying formula (6) in DT, i.e., on the set (7).

$$\| \bigwedge_{a \in B} (a = a(x)) \|_{DT} = \{y \in U : a(y) = a(x) \text{ for } a \in B\}, \quad (7)$$

is at most tr. Assume that requirement interaction $ri \in [0, 1]$ instead of $ri \in \{SC, WC, VWC, NC\}$, in other words, the team decides to use numbers (finer granularity) to represent technical conflict degree rather than levels. Table I can be rewritten as Table IV.

TABLE IV

SOCIO-TECHNICAL MODEL

$R1$	E	I	S	R	T	d	ri
$r1.1$	M	H	N	L	Y	0.22	0.4
$r1.2$	M	H	N	L	Y	0.44	0.25
$r1.3$	H	M	N	M	Y	0.2	0
$r1.4$	L	H	Y	L	Y	0	0.9
$r1.5$	M	H	P	H	Y	0.67	0.35
$r1.6$	M	L	P	H	N	0.89	0.2

---

[3] To compute such reducts one can follow a method presented in [16] 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 [2].

For example, if we assume $tr = 0.3$, we obtain the following discernibility function for the distance reducts for the socio-technical decision table represented in Table IV (for simplicity, we consider only the decision d):

$$(S \vee R) \wedge (E \vee S). \qquad (8)$$

Hence, we obtain two reducts $\{S\}$ and $\{E, R\}$. One can see that the deviation of the social conflict degree d on the indiscernibility classes defined by $\{S\}$ or $\{E, R\}$ is at most 0.24 and 0.22, respectively. However, if we exclude certain conditions (team negotiation parameters) the deviation increases, e.g., for the indiscernibility class defined by E and $r1.1$, the deviation is 0.67 and for the indiscernibility class defined by R and $r1.1$ is 0.44. However, the distance in technical conflict degree ri on indiscernibility classes defined by $\{S\}$ or $\{E, R\}$ is at most 0.15. Again for the indiscernibility class defined by E and $r1.1$, the deviation is 0.2 and for the indiscernibility class defined by R and $r1.1$ is 0.65. So there is considerable variation in deviations between social and technical conflict degrees. The distance reducts can be generalized by adding a requirement that the reduced set of attributes should not only preserve the discernibility between objects but to preserve it to a degree (at least equal to a given threshold). In this case one can use reducts proposed in [17].

One can analyze the dynamics of conflict degree changes by dropping conditions from defined risk patterns. Excluding certain conditions may cause small changes to the deviation of decisions, while excluding other conditions can lead to a substantial increase in the decision deviation. In other words, different sets of issues can lead to differing deviations in social and technical conflict degrees.

VII. RELATED WORKS

Basic ideas of conflict theory in the context of rough sets are due to [9]. Investigations about conflicts and groups of agents were presented in [18]. Recent research with approximate reasoning about vague concepts in conflict resolution and negotiations between agents (information sources) [7], [5], requirements negotiation decision model [1], trace-dependency for identifying conflicts and cooperation among requirements [4], requirements interaction management [23] provide a basis for comparison of the proposed approach and also points to the usefulness of a unified framework for software requirement conflict analysis and negotiation. Inconsistent requirements (technical conflicts) using classifiers based on rough sets can be found in [8]. Central to this research is the notion that a conflict relation can be viewed as a special kind of discernibility relation. The relationships between the approach to conflicts and information systems as well as rough sets are illustrated in [15], [19], [20], [21].

VIII. CONCLUSION

This paper introduces a unified framework and an architecture for a $socio-technical$ conflict model based on rough sets for $requirements\ interaction$ which encapsulates both social conflicts (stakeholder viewpoints) and technical conflicts (requirements interaction). The unified framework makes it possible to represent requirements (both functional and non-functional) and their attributes, identify technical conflicts and coalitions (cooperation) and incorporate trace dependency information. The proposed research points to a means of simplifying the representation of technical and social conflicts. Conflict dynamics using risk patterns where one can measure deviations of the conflict degree amongst social as well as technical viewpoints has been introduced. This is achieved with distance reducts extracted from conflict data using Boolean reasoning.

Acknowledgments. The research of Andrzej Skowron 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 and NSERC Canada grant 194376, respectively.

REFERENCES

[1] Boehm, B., Grünbacher, P., Kepler, J.: Developing Groupware for Requirements Negotiation: Lessons Learned, IEEE Software, May/June (2001) 46-55.

[2] Deja, R., Skowron, A.: On Some Conflict Models and Conflict Resolutions, *Romanian Journal of Information Science and Technology* 5(1-2) (2002) 69-82.

[3] Egyed, A., Grünbacher, P.: Automating Requirements Tracebility: Beyond the Record and Replay Paradigm, Proceedings of the 17th International Conference on Automated Software Engineering, IEEE CS Press, (2002) 163-171.

[4] Egyed, A., Grünbacher, P.: Identifying Requirements Conflicts and Cooperation: How Quality Attributes and Automated Traceability Can Help, *IEEE Software*, November/December (2004) 50-58.

[5] Kraus, S.: *Strategic Negotiations in Multiagent Environments*, The MIT Press, (2001).

[6] Leffingwell, D., Widrig D.: *Managing Software Requirements*, Addision-Wesley, MA (2003).

[7] Lai, G., Li, C.. Sycara, K., Giampapa, J.: Literature Review on Multi-attribute Negotiations, Technical Report CMU-RI-TR-04-66 (2004) 1-35.

[8] 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 3642, Springer, Heidelberg (2005) 678-687.

[9] Pawlak, Z.: On Conflicts, *Int. J. of Man-Machine Studies* 21 (1984) 127-134.

[10] Pawlak, Z.: Anatomy of Conflict, *Bulletin of the European Association for Theoretical Computer Science* 50 (1993) 234-247.

[11] Pawlak, Z.: *Rough Sets – Theoretical Aspects of Reasoning about Data*, Kluwer Academic Publishers (1991).

[12] Pawlak, Z.: An Inquiry into Anatomy of Conflicts, *Information Sciences* 109 (1998) 65-78.

[13] Pawlak, Z., Skowron. A.: Rough Sets and Boolean Reasoning, *Information Sciences* 177(1) (2007) 41-73.

[14] Pressman, R.S.: *Software Engineering: A Practioner's Approach*, Mc-Graw Hill, New York, (2001).

[15] Skowron, A., Ramanna, S., Peters. J.F. : 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, LNCS 4062, Springer, Heidelberg (2006) 233-240.

[16] 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 11, Kluwer, Dordrecht (1992) 331-362.

[17] Skowron, A.: Extracting Laws from Decision Tables, *Computational Intelligence: An International Journal* 11(2) (1995) 371-388.

[18] Stepaniuk, J.: Conflict Analysis and Groups of Agents, In: Proceedings of the Poster Session, Tenth International Symposium on Methodologies for Intelligent Systems, (ISMIS'97), Charlotte, NC, October 15-18 (1997), 174-185.

[19] Ramanna, S., Peters, J.F., Skowron, A.: Generalized Conflict and Resolution Model with Approximation Spaces, In: S. Greco, Y. Hata, S. Hirano, S. Miyamoto, H. S. Nguyen, R. Słowiński (eds.), Proceedings of the Fifth International Conference on Rough Sets and Current Trends in Computing (RSCTC 2006), Kobe, Japan, November 6-8, 2006, LNAI 4259, Springer, Heidelberg (2006) 274-283.

[20] Ramanna, S., Peters, J.F., Skowron, A.: Analysis of Conflict Dynamics by Risk Patterns. In: G. Lindemann, H. Schlingloff, H.-D. Burkhard, Czaja, L. Penczek, W. Salwicki, A. Skowron, A. Suraj, Z. (eds.), Proceedings of the 15th International Workshop on Concurrency, Specification, and Programming (CS&P 2006), Wendlitz, Germany, 27-29 September, 2006, Informatik-Berichte 206, Humboldt Universität, Berlin (2006) 469-479.

[21] Ramanna, S., Peters, J.F., Skowron, A.: Approaches to Conflict Dynamics based on Rough Sets. *Fundamenta Informaticae* 75(1-4) (2006) 453-468.

[22] Ślęzak, D.: Approximate Entropy Reducts. *Fundamenta Informaticae* 53 (2002) 365–387.

[23] Robinson, W, N., Pawlowski, D, S., Volkov, V.: Requirements Interaction Management, *ACM Computing Surveys* 35(2) (2003) 132-190.

[24] Thayer, R.H.,Dorfman, M.: *Software Requirements Engineering.* IEEE Computer Society Press (1997).