Issues on Conflict Resolution in Collaborative Networks

Davide Carneiro, Paulo Novais, Flávio Lemos, Francisco Andrade, José Neves

HAL Id: hal-01570006
https://inria.hal.science/hal-01570006

Submitted on 28 Jul 2017

Issues on Conflict Resolution in Collaborative Networks

Davide Carneiro[1], Paulo Novais[1], Flávio Lemos[1], Francisco Andrade[2], José Neves[1]

[1]Department of Informatics, University of Minho, Braga, Portugal
{dcarneiro, pjon}@di.uminho.pt, pg15990@uminho.pt, jneves@di.uminho.pt
[2] Law School, University of Minho, Braga, Portugal
fandrade@direito.uminho.pt

Abstract. Conflicts are frequent in virtually every scenario involving complex interactions. Collaborative Networks, in which there is a compromise between skills, competencies and resources, are not an exception. Moreover, these conflicts can be different in nature, ranging from cultural or relational conflicts to conflicts of interests. Although conflicts are common in these virtual settings, very few tools exist to settle them. Therefore, parties involved in conflict resolution generally have to resort to traditional approaches, which delay the process and waste the advantages of these Virtual Organizations. In this paper we present a structured model of a contract for a Collaborative Network. Based on this model, we developed an information system which is able to build conflict scenarios and determine the possible and probable outcomes. We combine this development with the UMCourt platform, which comprises a rich negotiation and mediation tool, to build a conflict resolution framework in the context of Collaborative Networks.

Keywords: Virtual Organization, Conflict Resolution, Decision Support System.

1 Introduction

Conflicts are natural and emerge as a consequence of our complex society, in which each individual focuses on the maximization of the own gain. A conflict can be seen as an opposition of interests or values which, in a certain way, disturbs or blocks an action or a decision making process. In order for the action to be carried out, the conflict has to be solved [1]. The concept of conflict and its resolution is usually addressed by two main scientific fields: Social and Information Science. The intersection of these two fields is of great interest, particularly from the point of view of a Virtual Organization (VO).

In fact, conflicts in VOs are frequent given the unstable balance between skills, competencies and resources. These conflicts are generated in virtual settings, most of the times supported by an electronic contract. However, very few tools exist to settle conflicts inside their context. As a consequence, conflicting members have to resort to traditional conflict resolution methods, throwing away significant advantages of VOs, rendering them inefficient. The use of technology to develop tools that can support the conflict resolution process, together with the creation of virtual environments for

that purpose is thus of interest. Moreover, Pitt et al. address the issue of the costs of conflicts in the context of VO and the need for alternatives to litigation [2]. In particular, the authors argue that litigation is a slow and costly process which may have a significant impact on the business of companies and governments. The potential for appeals also adds to the amount of delay and cost.

In this paper we present a model for an electronic contract that defines the rules that govern the lifecycle of the VO, focusing on solving Operational Conflicts. Given its structured nature, this model can be handled by a software agent. Specifically, we are interested in developing software agents able to point out the consequences of violating specific rules. The integration with the UMCourt conflict resolution platform [9], specialized in negotiation and mediation algorithms, allows the development of a framework for conflict resolution in line with the recent trend of Online Dispute Resolution (ODR) [3]. In this work we take into consideration the work of prof. Lewiki et al. [6] and Goldberg et al. [8] on the dynamics and processes of conflict resolution, and the work of Raiffa [7] on decision theory and negotiation analysis. During its development, an agent-based approach was followed [10, 11].

2   Conflicts in Virtual Organizations

Conflicts are part of the lifecycle of the VO and should be regarded as natural. Moreover, depending on how well they are managed, conflicts can even be seen as a catalyst for the success of the VO. When it comes to conflicts, solving them in VOs may, at first sight, appear to have an added challenge, when compared to traditional conflicts: the environment of the conflict is a virtual one (e.g. no physical interaction, no social contact, geographical dispersion of members). As a way to overcome these challenges, Online Dispute Resolution tools can be used that support distributed and asynchronous conflict resolution. Such tools aim conflict resolution methods processes that are agile and flexible, in line with the requirements of the VO.

The nature of the conflict in a VO can be quite diversified. Taking ARCON [11] as a reference model, several types of conflicts can be identified. Considering the intrinsic nature of the VO, four dimensions can be pointed out in which conflicts are likely to emerge:

- *Structural dimension*: this dimension includes features related to the structure and composition of the VO, including the members, their relation and roles;
- *Component dimension*: includes hardware/software/human resources as well as knowledge or information;
- *Functional dimension*: includes all the tasks and functionalities supported by the VO as well as methodologies and process models;
- *Behavioral dimension*: includes guidelines for cooperation or conflict resolution, contracts, policies and governance rules applicable to all the members.

On the other hand, considering the environment around the VO, another four dimensions in which conflicts may raise can be identified:

- *Market dimension*: includes the issues related to the interaction of the VO with external entities *such* as clients and competitors;

- *Support dimension*: this *dimension* refers to external support services, including certification, financial or insurance;
- *Social dimension*: *includes* the impact of the VO on the society, including in issues like employability, legal, or educational.
- *Constituent dimension*: models the impact and consequences of the integration of potential *new* members, including issues like sustainability or membership rules.

After performing an analysis of these eight dimensions, it is possible to identify the main types of conflict that can arise in the context of a VO:

- *Operational Conflicts*: operational conflicts emerge from situations in which the norms established in the creation (or modified during the operation) of the VO are being *violated*. Conflicts may emerge from a member not providing a given service as contracted, lack of communication, refusal to share some resource or general infringement of the governance rules;
- *Relational conflicts*: although the members of the VO are independent, they are inter-dependent in terms of *resources*. Trust is thus an important factor. Incompatibilities in the interactions between the members can be seen as relational conflicts. These conflicts can be originated by past negative experiences that influence the level of trust;
- *Conflicts of interests*: the correct availability and use of resources is fundamental for the correct operation *of* a VO. This kind of conflicts is generally related with the misuse of resources, namely the scenarios in which members use resources of the VO outside of its scope;
- *Business strategy conflicts*: members of VO are generally attracted by the eventual access to new technologies, *markets* or information. However, this can be seen as a threat by the members sharing these resources. These conflicts are more evident in the initial phase of the VO, when the members are defining the availability and access to resources and tend to diminish with the time;

3 Monitoring Conflicts in Virtual Organizations

The contracting process in a VO is an important aspect in which the activities of the members of the organization are depicted. The result of this process is a contract in which the VO is described in all its dimensions. Contracts define the role of each member as well as the rules and operating principles of the organization. In that sense, the monitoring of a contract throughout the lifecycle of the VO, mainly during the operational phase, is of utmost importance. In this section we describe the structure of the contract and the monitoring mechanism, developed to deal with Operational Conflicts.

3.1 The Contract

Given that contracts are essential for the detection of Operational Conflicts, there is the need for a contract model that allows representing all the endogenous elements of

the VO, with a special focus on the behavioral and structural dimensions. Based on several works on the field [12 - 14], it is possible to define a contract model from a logical point of view, with four central components:

- **Actions** – are seen as atoms and describe what each member can do. An action is defined as a 4-tuple: $a = (name, sender, receiver, t)$, where *name* is the unique identifier of the action, to be executed by the *sender*, whose recipient is *receiver*, in time *t*. The set of all possible actions within the scope of a contract is defined as: $A = \cup\, a$;

- **Events** – identify the events that take place within the environment of the organization. Events are defined as: $e = (a(n, s, r, t), event_{type})$, where $a \in A$ and event_type $\in \{fact, fulfill, violated\}$ describes the type of event. In this description, *fact* denotes that an action *a* occurred in a given time, *fulfill* denotes that an action *a* was fulfilled in a given time and *violated* denotes that there has been a violation of a norm in a given instant;

- **Obligation** – identify all the obligations that can be subscribed by the members of the organization. An obligation is defined as: $o = (a(n, s, r, t), liveline, deadline)$, where $a \in A$ denotes an action that must be executed within a timeframe defined as [liveline, deadline];

- **Norms** – establish the rules that will guide the behaviour of the members of the *organization*. A norm is defined as: $n = (name, antecedent, consequent)$, where *name* is the unique identifier of the norm and *consequent* describes obligations that must be implemented when a given *antecedent* (an event) is true.

Based on this, a model for a contract has been defined using XML Schema, a rich data typing system that allows defining the structure of XML documents with detailed constraints on its logical structure. Instances of contracts are thus XML files.

3.2   Monitoring Conflicts

In order to monitor the emergence of conflicts during the operation of the VO, a software agent was developed that is able to interpret the contract defined. This model works under the assumption that members of the VO advertise their actions in the form of events, i.e., the model will fail if a member hides his actions from the rest. Thus, during the operational phase, the members of the VO use a framework for advertising actions. The approach followed consisted in developing a software agent that constantly monitors these actions in search for violations of contract. On initialization, it interprets the contract and creates a temporal knowledge base describing the norms, the state of the VO, among other issues. This agent is built by two main modules: *Inference* and *Communication*. The *Communication* module is responsible for establishing a bridge between the framework and the inference mechanism. In that sense, it receives the events and forwards them to the *Inference* module, receiving back messages that it will forward to the framework, in order to inform the members. The *Inference* module is implemented in Prolog, a general purpose logic programming language. In that sense, the rules and facts retrieved from the contract and forwarded by the *Communication* module are represented in the form of Horn clauses.

Whenever a new event arrives in the *Inference* module, it is added to the knowledge base. The inference mechanism will then check if there is a norm $n = (name, event, obligation)$ whose antecedent (event) is true. If that is the case, the inference mechanism will point out the resulting obligation. This obligation will be added to a set M of obligations under monitoring. M is defined as: $M = \cup \{(o, s)\}$, where o represents an obligation and $s \in \{fullfilled, pending, violated, pre\_violated\}$ its state.

Definition 1. Given an event $e = (action_a(n_a, s_a, r_a, t_a), event\_type)$ and an obligation $o = (action_b(n_b, s_b, r_b, t_b), liveline, deadline)$, o is considered to be fulfilled when $action_a = action_b$ and $t \in [liveline, deadline]$. Otherwise, the event e does not satisfy obligation o.

In order to monitor all the obligations in M, the software agent goes through all of its members at regular intervals, with the objective of updating the state of each obligation. As stated above, in a given time instant *t*, the state of an obligation *o* can be one of four:

- *Fulfilled* – when there is an event *e* such that *o* is fulfilled;
- *Pending* – when there is no event *e* such that *o* is fulfilled and $t < liveline$;
- *Violated* – when there is no event *e* such that *o* is fulfilled and $t > deadline$;
- *Pre_Violated* – when there is no event *e* such that *o* is fulfilled and $t \in$ [liveline, deadline].

Whenever there is a pair $(o, e) \in M$ such that $e = violated$, the inference builds and sends a message to the *Interface* module describing the occurrence. The interface will then identify all the members directly and indirectly involved in the action associated with obligation *o*. The members will then decide the course of action. To do it, they may make use of a negotiation platform being developed under the scope of the TIARAC project, presented in the following section.

4   Reaching an Outcome

Members of the VO can use the previously depicted method as a decision support system, to detect violations of contract and possible solutions, in the form of actions. However, the members may not agree on the solution. In that sense, this work was integrated with a negotiation support tool developed in the context of the TIARAC project – *Telematics and Artificial Intelligent in Alternative Conflict Resolution*. The negotiation tool is based on the Case-based Reasoning paradigm. In that sense, it implements an algorithm that looks into past similar conflicts and their respective solutions [15, 16] (Figure 1). It is a dynamic method in the sense that it is able to adapt strategies according to how the process evolves. In that sense, it considers parameters like trust, the attitudes of each member or the historic of actions.

It is now possible to provide a general view on how the whole process develops (Figure 2). At the beginning, the VO is created and a contract is defined according to the model. In the Monitorization module, a logical representation of the norms is created and stored in the knowledge base. When the VO reaches the operational

phase, the process develops around the interactions between the members. Whenever a new event is published by a member, it is forwarded to the Monitorization module, which in turn will check if there is any norm whose antecedent matches the event. If it is true, it will check if the antecedent is violated, in which case a notification is sent to the members of the VO and a conflict resolution process starts.
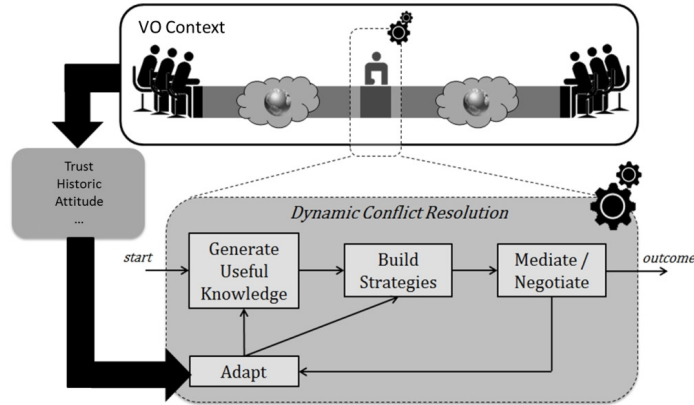

Fig. 1. High level view of the conflict resolution process in the context of a VO.
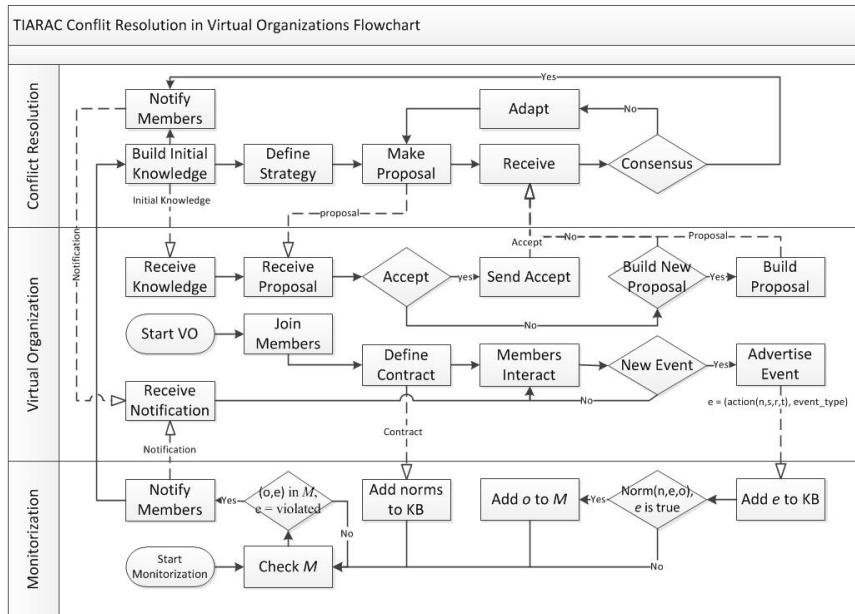

Fig. 2. The flowchart of the Conflict Resolution process in Virtual Organizations.

The process starts by building initial knowledge that is made available for the members. To select this knowledge, the algorithm analyzes past known conflicts and assesses their similarity in terms of the contracts and the nature of the norms addressed and violated. Based on this, it selects a list of relevant cases (i.e. cases

above a given threshold of similarity) and sorts them according to that value of similarity. From this point on, the conflict resolution process develops in repetitive rounds. In each round the Conflict Resolution module proposes a solution (retrieved from the similar cases). Each member of the VO may accept it, reject it or build new proposals for solution (Figure 3). If there is a consensus on the proposals sent by the members, the process ends successfully and the VO may resume its activity. If there is no consensus, the system will adapt its strategy and propose a new solution. This process goes on until all members agree on a solution or at least one member leaves the process.
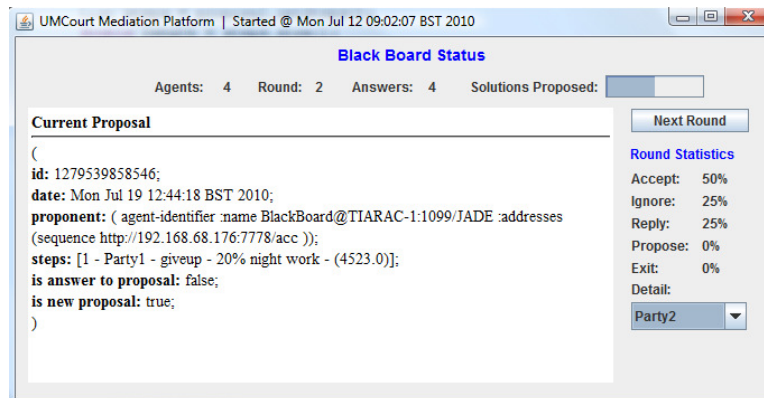

Fig. 3. The interface for the *blackboard* agent, responsible for the control of the lifecycle of the negotiation.

5   Conclusions

In a broad sense, conflicts configure an obstacle in a decision making or action taking process. In that sense, for the action to be carried out, the conflict must be solved first. Inefficient conflict resolution mechanisms may delay these processes. This is particularly disturbing in the context of VOs, especially when no technology-based tools exist to support the process, throwing away significant advantages of this organization paradigm. Particularly negative are the costs and time spent litigating in court, which has a significantly negative impact on the performance of governments and companies. In this paper we presented a tool for conflict resolution, based on the paradigm of Online Dispute Resolution. This tool consists of two main components. The first one constantly monitors the actions of the members of the VO and, with a representation of the contract, determines when a violation occurs, pointing out the obligations of the parties involved. The second component comprises a negotiation tool which allows the members to work out a satisfactory solution. This tool considers past similar problems to more efficiently propose solutions that can rapidly lead to the resolution of the conflict. With this approach we expect to achieve conflict resolution mechanisms fit for the context of Virtual Organizations.

References

1. Lewin, K.: Resolving social conflicts: Field theory in social science. American Psychological Association, ISBN 1557984158 (1997)
2. Pitt, J., Ramirez-Cano, D., Kamara, L., Neville, B.: Alternative Dispute Resolution in Virtual Organizations. In: Artikis, A., O'Hare, G., Stathis, K., Vouros G. (Eds.): Engineering Societies in the Agents World VIII. Lecture Notes in Artificial Intelligence, Vol. 4995. Springer-Verlag, pp. 72-89 (2008)
3. Katsch, E., Rifkin, J.: Online dispute resolution – resolving conflicts in cyberspace. Jossey-Bass Wiley Company, San Francisco (2001)
4. Deutsch, M., Coleman, P., Marcus, E.: The Handbook of Conflict Resolution: Theory and Practice. John Wiley and Sons (2006)
5. Peruginelli, G., Chiti, G.: Artificial Intelligence in alternative dispute resolution. In: Proceedings of the Workshop on the law of electronic agents – LEA (2002)
6. Lewicki, R.J., Barry, B., Saunders, D.M., Minton, J.W.: Negotiation, 4th ed., McGraw-Hill/Irwin (2003)
7. Raiffa, H.: The Art and Science of Negotiation. Harvard University Press (2002)
8. Goldberg, S.B., Sander, F.E., Rogers, N., Cole, S.R.: Dispute Resolution: Negotiation, Mediation and Other Processes, Aspen Publishers, New York (2003)
9. Carneiro, D., Novais, P., Costa, R., Neves, J.: Enhancing the Role of Multi-agent Systems in the Development of Intelligent Environments. In: Advances in Intelligent and Soft Computing, Vol. 71, Springer-Verlag, ISBN 978-3-642-12432-7, pp. 123-130 (2010)
10. Wooldrige, M.: An Introduction to Multiagent Systems. John Wiley & Sons (2002)
11. Camarinha-Matos, L.M., Afsarmanesh, H.: Towards a Reference Model for Collaborative Networked Organizations. In: Information Technology for Balanced Manufacturing Systems, IFIP International Federation for Information Processing, Volume 220/2006, 193-202, DOI: 10.1007/978-0-387-36594-7_21 (2006)
12. Boella, G. and Torre, L. van der , Contracts as Legal Institutions in Organizations of Autonomous Agents, in N. R. Jennings, C. Sierra, L. Sonenberg and M. Tambe, eds., Third International Joint Conference on Autonomous Agents & Multi Agent Systems, ACM Press, New York, NY, United States, 2004, pp. 948-955.
13. Lai Xu: A multi-party contract model. ACM SIGecom Exchanges, Vol. 5, Issue 1, pp. 13-23 (2004)
14. Cardoso, L., Oliveira, E.: A contract model for electronic institutions. In: Sichman, J., Padget, J., Ossowski, S., and Noriega, P. (Eds.): Proceedings of the 2007 international conference on Coordination, organizations, institutions, and norms in agent systems III (COIN'07), Springer-Verlag, pp. 27-40 (2007)
15. Carneiro, D., Novais, P., Andrade, F., Neves, J.: Using Mediation to Solve Disputes with Avoiding Parties. In: Proceedings of the JURISIN 2010 – Fourth International Workshop on Juris-informatics, Tokyo, Japan, ISBN 978-4-915905-42-1, pp. 17-28, (2010)
16. Carneiro, D., Costa, A., Novais, P., Andrade, F., Neves, J.: Providing Relevant Knowledge in Disputes. In: Proceedings of the 6th International Workshop on Online Dispute Resolution, JURIX2010, pp. 63-78, (2010)