

Repositório ISCTE-IUL

Deposited in *Repositório ISCTE-IUL*: 2019-02-08

Deposited version: Post-print

Peer-review status of attached file:

Peer-reviewed

Citation for published item:

Jardim-Gonçalves, R., Cretan, A., Coutinho, C., Agostinho, C. & Bratu, B. (2018). Coordinating negotiations in data-intensive collaborative working environments using an agent-based model-driven platform. Enterprise Information Systems. 12 (8-9), 1100-1128

Further information on publisher's website:

10.1080/17517575.2018.1521998

Publisher's copyright statement:

This is the peer reviewed version of the following article: Jardim-Gonçalves, R., Cretan, A., Coutinho, C., Agostinho, C. & Bratu, B. (2018). Coordinating negotiations in data-intensive collaborative working environments using an agent-based model-driven platform. Enterprise Information Systems. 12 (8-9), 1100-1128, which has been published in final form at

https://dx.doi.org/10.1080/17517575.2018.1521998. This article may be used for non-commercial purposes in accordance with the Publisher's Terms and Conditions for self-archiving.

Use policy

Creative Commons CC BY 4.0 The full-text may be used and/or reproduced, and given to third parties in any format or medium, without prior permission or charge, for personal research or study, educational, or not-for-profit purposes provided that:

- a full bibliographic reference is made to the original source
- a link is made to the metadata record in the Repository
- the full-text is not changed in any way

The full-text must not be sold in any format or medium without the formal permission of the copyright holders.

Coordinating Negotiations in Data-intensive Collaborative Working Environments using an Agent-based Model-driven Platform

Ricardo Jardim-Goncalves^a, Adina Cretan^b*, Carlos Coutinho^c, Carlos Agostinho^d, and Ben Bratu^e

^aCTS, Departamento de Engenharia Electrotecnica, FCTUNL, UNINOVA, Caparica, Portugal, E-mail: rg@uninova.pt;

^bComputer Science Department, "Nicolae Titulescu" University of Bucharest, Romania, E-mail:bratuadina@gmail.com;

^cCaixa Mágica Software/Instituto Universitário de Lisboa (ISCTE-IUL), ISTAR-IUL, Lisboa, Portugal, Email:carlos.coutinho.phd@gmail.com;

^dCTS, FCTUNL, UNINOVA, Caparica, Portugal, E-mail: ca@uninova.pt;

^eDatadirect Networks, IME R&D, Meudon La Foret, France, E-mail: bbratu@ddn.com

Coordinating Negotiations in Data-intensive Collaborative Working Environments using an Agent-based Model-driven Platform

This paper tackles the interoperability problems of enterprise information systems by presenting a distributive model-driven platform for parallel coordination of multiple negotiations in data-intensive collaborative working environments. The proposed model was validated and verified by an industrial application scenario within the European research project H2020 C2NET (Cloud Collaborative Manufacturing Networks). This real scenario developed dataintensive collaborative and cloud-enabled tools that allow the optimisation of the supply network of manufacturing SMEs, proposing a negotiation solution based on a model-driven interoperable decentralised architecture.

Keywords: enterprise information systems; interoperability; model-driven; coordination rules; negotiation; intelligent agents

1. Introduction

The path to meet the challenges of globalisation mandates companies – and particularly Small and Medium Enterprises (SMEs) – to find new and improved mechanisms of collaboration within a dynamic networked environment. Particularly in the field of manufacturing, there are numerous SMEs which are providers, partners and customers of the manufacturing business, which need to adapt to cope with the needs of the other partners (mainly big companies which are less prone to change), or to create new opportunities for interconnecting with new businesses. The major strength that these SMEs have is exactly their number, which promotes interoperability, cooperation and composition of the provided services to build more value to the businesses. This heterogeneity in the different approaches that are taken by SMEs may lead to a point where interoperability is not possible, thus we can state that an interoperability breaking point has been reached, i.e., to change the systems and solutions to be able to interoperate with other partners there needs to be a major rework of all APIs, processes, and especially the migration of legacy processes and systems. Such effort may not even in some cases compensate the gains related to the change. In this sense, the best way to ensure that the impact of the changes is the least possible is to define a set of models which describe the running assets and processes, and promote a model-driven approach to adapt automatically to model changes. Moreover, the development of cloud computing and the rapid evolution of data-driven services raise autonomy, adaptability and scalability problems for SMEs. To succeed in having a seamless collaboration, it is crucial for SMEs to establish and maintain the interoperability of their systems and applications to better respond to their customers' demands (Agostinho and Jardim-Goncalves 2015).

This paper presents a model-driven approach based on a negotiation solution to tackle data-intensive Enterprise Interoperability (EI) problems, which aims at reducing the impact of changes, re-establishing and maintaining interoperability in collaborative working environments. The proposed negotiation solution consists of a generic model, described by a methodical sets of coordination rules which handle simultaneous bilateral negotiations, for supporting interoperability in business-to-business interactions within a dynamic networked environment.

This solution fits the goal of experimenting industrial applications, deployment methodologies and demonstrations that will help manufacturing industry assessing the application of agent technology.

This paper presents and analyses reference related work on Section 2. This supports the hypotheses raised in Section 3 and the solution proposed in Section 4. Section 5 follows, providing the coordination model described by negotiation methods, based on a methodical set of rules, and different types of dependencies that can be modelled during the negotiation process. Section 6 discloses the coordination patterns which enable the infrastructure to coherently manage the different coordination rules, depending on negotiating tactics that can be coordinated simultaneously. In addition, based on the proposed model of coordination rules, an example of negotiating tactics (e.g., Transport) is also presented in this section. Section 7 presents the validation of the negotiation framework, while Section 8 describes an industrial validation case from the transport area. This section also presents two negotiation examples that illustrates and verifies the utility of the proposed coordination of negotiations solutions for a transport scenario within the C2NET project. Moreover, a roadmap for model implementation in manufacturing project is provided in this section. Finally, Section 9 discuss the conclusions and directions for future work.

2. Related Work

The new concept of collaborative enterprise composed of several interdependent organisational entities is characterised by its ability to flexibly interact with partners within dynamic business networks in order to better meet customer needs and collectively exploit business opportunities. One of the considerable challenges is to establish and maintain the collaboration among partners at business level as well as at information technology (IT) level requiring new solutions to deal with their different IT applications, platforms and infrastructures. In respect to this, van Sinderen and Spies (2009) introduce the main trends in enterprise computing relating to improved enterprise models, model-driven development (MDD) and service-oriented architecture (SOA). The authors emphasize the importance of the model-driven approach and its application in the development of large distributed IT systems to support collaborative enterprises, especially enterprise interoperability, within dynamic business networks. Model-driven engineering (MDE) approach presents the concept of development process by describing models in the modelling space (Kent 2002). In this regard, Zečević et al. (2018) propose a solution to model hybrid databases, based on empirically validated concepts of MDE.

Despite theoretical advantages of the model-driven approach for the development and evolution of enterprise computing systems, there are some issues related to its practical application. Addressing the problem of enterprise models applications, Iacob and Jonkers (2009) advocate that MDD can become more effective if business rules are incorporated in the context of SOA.

In order to improve enterprise models, a complete understanding of the modelled enterprise is needed, requiring an efficient language to support communication and reasoning in a specific domain. Related to this, Terzić et al. (2017) develop the MicroDSL domain-specific language used for microservice software architecture modelling.

By comparison with model-driven aspects that simplify the engineering process using model abstraction and automation, the data-driven approach, based on machine learning, requires the capability to build advanced-analytics models for predicting and optimizing outcomes. For instance, Wang et al. (2018) aim to provide a data-intensive cycle time parallel prediction for production planning in wafer manufacturing using artificial neural network and parallel computing approaches. In the same context, Stojanov et al. (2018) present a solution for extending a model of data-driven software systems with a software change request service.

Ensuring the seamless collaboration among different enterprise information systems is very difficult to achieve due to the heterogeneity of enterprise applications. In respect to this, Benaben et al. (2013) highlight the major position of information systems (IS) within an enterprise, stating that the critical issue is to ensure and maintain a collaborative situation between the IS of the partners involved. Tackling the issue of integration of enterprise information systems, Izza (2009) argues that maintaining the IS interoperability is the main challenge for organizations, stating that a mediation process is needed to deal with technical, syntactical and semantic conflicts.

In this context, this paper proposes negotiation as a key solution in solving the conflicts in order to achieve and preserve the interoperability of systems and applications by reaching a common agreement unanimously accepted by all parties involved. Many research papers highlight the importance of developing an automated negotiation environment. One of the approaches refers to the improvement of negotiating strategy. In this respect, Fujita (2016) proposes an automated negotiation agent that can predict partner's strategies based on previous negotiations. Caillere et al. (2016) address the issue of multiple negotiations by proposing a protocol and rules that guides the agents to coordinate their interactions and to reach an agreement.

Multi-agent systems (MAS) have been the basis for many automated negotiation processes due to their autonomy and cooperative capabilities. Therefore, many research papers provide multi-agent-based solutions to solve flexible and autonomous problems in dynamic virtual enterprises (VEs) environments. Hence, in order to allow agents to dynamically adapt their negotiation strategies, Yu et al. (2013) present a model for bilateral negotiation based on Bayesian learning, while Zhang et al. (2012) describe a multi-agent architecture for semantic discovery of manufacturing services in VE. Li and Wang, (2007) propose a bilateral negotiation model based on fuzzy constraint satisfaction problems in electronic commerce trading.

Other approaches address the issue of data-intensive services provision in a dynamic environment by proposing bio-inspired algorithms facility (Wang et al. 2013, 2014). In the context of MAS applications of automated negotiations, Alrayes, Kafali, and Stathis (2016) propose an experimental simulation platform which supports concurrent negotiation interactions among agents. Faratin, Sierra, and Jennings (1998) were precursors followed by many researchers that have developed negotiation models for various industrial fields such as logistic domain, more exactly in supply chain community (Leão and Morais 2011).

Regarding the applications of MAS in Industrial Automation, several solutions have been proposed for supporting the implementation of the collaborative manufacturing systems. One of the solutions, related to the open standards Ethernet TCP/IP and Web technologies, is represented by Schneider Electric's Transparent Ready[™] product (Gelin and Colombo 2003), a collaborative approach to share data in real-time. Another solution, Factory Broker[™], has been proposed and implemented by Schneider Electric in cooperation with Daimler AG Corporation, and represents an intra-enterprise holonic control system (Lastra, Torres, and Colombo 2006). In the same context, PROSA (Van Brussels et al. 1998) and ADACOR (Leitão, Colombo, and Restivo 2006) architectures illustrate the application of the holonic concept to manufacturing control systems.

Other approaches tackle the issue related to the design of a negotiation environment, where two directions can be considered: one in which software agents replace humans in negotiations; and other in which negotiation agents provide an efficient support for human users, assisting them in the negotiation process. Regarding the first direction, Lin and Kraus (2010) define a conceptual environment where automated agents can proficiently negotiate with humans. The second direction is reflected in many research papers, (e.g., Badica et al. [2011]; Penders, Pavlin, and Kamermans [2010]) that propose a collaborative solution based on a service oriented architecture which helps inter-organizational information processing in distributed workflows. Then, there are other approaches like data-driven knowledge and inference which are relevant, (Andres et al. [2017]) propose for the C2NET project the establishment of standardised tables (STables) for the collection and analysis of information from the manufacturing area.

This paper proposes a model-driven approach to support the negotiation process by handling the dependencies in complex multilateral and multi-issue negotiations. In previous work, the authors have described the generic architecture of the negotiation system involving several coordination services corresponding to real-world multi-agent negotiations. Moreover, it has also been proposed an approach that divides the negotiation coordination aspects into two distinct classes, depending on the complexity of ongoing negotiations (i.e., negotiation in closed and open environment). Based on that model, the main contributions of the current research focus on solving the coordination problems that arise when the proposed negotiation infrastructure has to coherently manage multiple concurrent negotiations. The proposed solution is based on a distributed negotiation system approach that allows to the coordination model to be distributed over several coordination modules. This approach allows to cope with a scalable environment and to easily reduce/increase the coordination complexity by removing/adding independent coordination modules. In the current work, the authors provide the concepts and the tools allowing to define and integrate new coordination modules for a multi-party negotiation case. This article introduces the concepts and constraints definitions as well as the different types of dependencies (e.g., status, attribute and role dependencies) that can be used as basic bricks to construct a coordination behaviour. Further these concepts and dependencies are used to define a set of methods (i.e., Program Formula) that model the evolution of negotiations in terms of proposals and counterproposals and a set of rules used for the implementation of the

coordination mechanism (i.e., Coordination Pattern) according to chosen negotiation tactic.

By comparison with the proposed state-of-the-art, where the coordination aspects are handled at protocol level or by complexifying the utility functions, the presented solution defines and coordinates the negotiation dependencies at middleware level. Hence, the proposed approach is a generic coordination solution where the utility function is transparent for the communication middleware, thus allowing it to be integrated in any deliberative MAS or directly as a support in a human interaction negotiation systems. The authors have chosen deliberative agents because of their ability to possess an internal image of the external environment, being able to plan their actions and to decide what action to perform in order to achieve their intended purpose. (Larson and Sandholm 2005).

3. Research Question and Hypotheses

This research develops a model-driven solution that facilitates and preserves systems interoperability in a collaborative working environment among partners within a network. The proposed solution refers to the semi-automated negotiation of various changes that may occur in this environment (e.g., due to a partner's change of activity) and could cause interoperability breaking points.

With respect to this, the authors formulate the following open research question: Can the reliability of a business collaboration system improve by solving its applications' interoperability aspects in collaborative working environments, ensuring data-intensive collaboration and recovery of interoperability after a breaking point?

This research question leads to the formulation of three hypotheses:

 i) If business-to-business exchanges are achieved based on a sustainable interoperability negotiation strategy, then the seamless collaboration can be maintained across time;

ii) If several negotiation exchanges and their shared information among the parties can be distributed and verified at the middleware level of the collaboration system, then the network is more robust and will be quicker in resolving and reestablishing broken data-intensive collaboration and interoperation services;

iii) If several negotiation partners are involved in multiple concurrent negotiations on different negotiation objects, then the negotiation infrastructure has to coherently manage these parallel negotiations.

Further, these hypotheses lead to the formulation of the main objectives of this research:

- To propose a negotiation platform that allows each participant to define its own negotiation criteria and constraints. If the negotiation ends with an agreement then the final result is unanimously accepted by all parties involved;
- To divide the coordination mechanism in order to be distributed in independent modules, which can make decisions and modify in parallel the course of the negotiation;
- iii) To propose a negotiation infrastructure able to achieve a global coordination
 (i.e., coordination among multiple participants) by ensuring winners from a
 set of participants, and a local coordination (i.e., coordination among
 multiple negotiations) in which each partner wants to optimize his own
 resources by choosing and conducting negotiations accordingly.

4. Negotiation System

A negotiation platform is typically a multi-agent platform in which intelligent agents are deliberative (Wooldridge and Jennings 1995).

The generic architecture of a negotiation system (Figure 1) has been described in the following paper (Cretan et al. 2012). In the proposed approach, the authors consider that every participant (e.g., a SME) has a negotiation infrastructure built as a single deliberative agent (denoted by NegA) and a common negotiation middleware with all the participants. There, a human participant (i.e., *Manager*) can start a negotiation by providing to his NegA the negotiated task (i.e., *Negotiation Object*) and the way that the negotiation will be conducted (i.e., *Negotiation Framework*) to handle the negotiation with another NegA in the system.

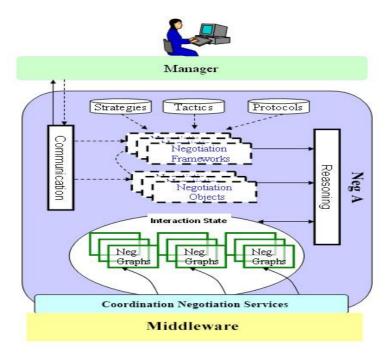


Figure 1. The negotiation system architecture

The Coordination Negotiation Services (CNS) layer ensures two significant roles in the coordination of the negotiation process: on one hand, it facilitates the transition between Neg A and Middleware layers, using a negotiation image described by graph structures;

on the other hand, it enables the implementation of various types of agent behaviours in real life negotiation situations. In this respect, seven negotiation services, namely: *Outsrc; Insrc; Block; Split; Broker; SwapIn/SwapOut; Transport* have been defined to handle manager-selected tactics which set the constraints on the negotiation process. Based on these specific coordination constraints, the negotiation services can build valid offers and counter-offers during the negotiation itself.

The proposed approach splits the coordination issues into two different classes, coordination negotiation services in a closed environment, and in an open environment. The *coordination negotiation services in a closed environment* use information from only one ongoing negotiation, tackling the coordination constraints according to it; while the *coordination in an open environment* uses information from multiple ongoing negotiations, managing the coordination constraints accordingly. Both classes of negotiation services indicate the main characteristics of the proposed coordination approach, namely *distributive* and *parallelism*.

In addition, these distributed coordination services that each partner can use, represent a solution to deal with multiple parallel negotiations. Therefore, a participant is not chaotically participating in various negotiations and accepting various contracts, but on the contrary, he can choose the negotiations in which he wants to participate, considering his resources in an optimal way.

The next sections describe a novel negotiation model-driven defined by methodical sets of rules which can be instantiated and triggered on top of the middleware. They also describe the negotiation-centric set of basic coordination rules validated by an example of negotiation tactics.

5. Coordination Rules

The negotiation-centric set of rules is structured in three main parts: i) status

dependencies – rules setting constraints between the execution time of negotiations and their corresponding states, ii) *attribute dependencies* – rules setting constraints between tasks and their negotiated attributes, and iii) *role dependencies* – rules setting constraints between negotiating partners.

Before detailing the rules, the following subsections indicate the basic concepts and the proposed constraints negotiation model.

5.1.Basic Concepts

The basic concepts refer to the following definitions:

A proposed *Negotiation Model* is represented as a quintuple $M = \langle T, P, N, R, O \rangle$ in which:

- *T* signifies the *time of the negotiation system*;
- *P* signifies the *set of negotiation partners* involved in one or more parallel oneto-one negotiations;
- *N* signifies the *set of negotiations* that take place in the system;
- *R* signifies the *set of negotiation coordination policies*. A coordination policy denotes a set of rules that establish dependencies among ongoing negotiations;
- *O* signifies the *common ontology* composed of a set of definitions of the negotiation object attributes for a particular domain.

Each *negotiation* is defined at a time instance as a set of negotiation sequences. Let $S = {si | i \in \mathbb{N}}$ denote the set of *negotiation sequences*, such that $\forall si, sj \in S, i \neq j$ implies si $\neq sj$. A *negotiation sequence* $si \in S$ such that $si \in N(t)$ is defined as a succession of negotiation graphs that describe the negotiation N from the initial moment and up to the time instance *t*. The *negotiation graph* created at a given time instance *t*, denoted with

G = (A, E) where A is the set of nodes and E is the set of edges, is represented as an oriented graph in which the nodes describe the negotiation offers in terms of attributes and values at that time instance t and the edges indicate the proposal – counterproposal sequences.

Status is defined as a set of possible states for a negotiation. Each state can take one of the values: (Status \in {*initiated*, *undefined*, *success*, *failure*}), such that *initiated* indicates the initial negotiation sequence; *undefined* indicates an ongoing negotiation sequence; and *success* and *failure* indicate the sequence in which an agreement has been reached or, respectively, the negotiation has been ended with a denial.

Issues is defined as a set of attributes with associated values that describe the negotiation offers.

Role is defined as a *set of participant roles* which can have one of the following values: *Role*= {*initiator, guest*}; where *initiator* represents the initial participant for the negotiation N and *guest* represents the invited participant in the negotiation N.

In order to implement the coordination model purposes, functions allowing to retrieve the values of the ongoing negotiation instances have been defined, such as: the function *status()* returns the state of a negotiation instance, the function *issues()* returns the set of pairs attribute-values of a negotiation instance, the functions *role()* and *role_s()* return the role of a participant in a negotiation or a negotiation sequence and the function *view()* returns the global negotiation framework set (i.e., the set of partners, the linked negotiations and the coordination rules) for an ongoing negotiation sequence.

The set of methods that guarantees the consistency of the negotiation model during the successive phases of proposal – counterproposal constitutes the *Program Formula (PF)*. Thus, PF is the representation of the coordination policy defined for a set of negotiations.

The next subsection presents the model that allows defining and coordinating a set of constraints.

5.2. Constraints Definition Model

Each coordination rule is described by the following structure:

<Rule_Definition> ':'[<Paramer_Definition >]* ';' <Graphs_conditions><Condition> <Relation> <Result>

where:

- <Rule_Definition> defines the rule name and its parameters e.g.,
 name_rule(v1,v2, ...,vn), with v_i ∈ T ∪ S.
- **<Paramer_Definition >** sets the variables related to their field of possible values e.g., *synchronize*($T_1, T_2, s1, s2$) : $T_1 \in \mathcal{T}$ $T_2 \in \mathcal{T}$.
- <**Graphs_conditions**> establishes the conditions related to graphs and, in particular, related to the nodes of the negotiation coordination sequences.
- The second part of the coordination rule is described by a left part named *condition* <**Condition**>, a right part named *conclusion* <**Result**>, and a relationship between both named (<**Relation**>).

The proposed constraints definition model sets two types of relationships between condition and conclusion as follows: i) *hard relationship* " \rightarrow •" and ii) *soft relationship* " \rightarrow "

Hard relationship (denoted $\rightarrow \bullet$) proves the following statement: if there is a time instance t₁ where conditions are met, then the conclusion of the relationship will be satisfied to the next time instance (t₁₊₁).

Soft relationship (denoted \rightarrow) proves the following statement: if there is a time instance t₂ where the conclusion is met, then the conditions have been met at the time instance t₁ < t₂ being true until the time instance t₂₋₁.

Therefore, even if at some point the conditions of the *soft relationship* are met, there is no guarantee that the outcome of the relationship will be obtained. A participant tries to meet the conditions of a soft relationship because they are necessary, but not sufficient to achieve the result.

Next, the different types of dependencies are presented which can be modelled among negotiation sequences.

5.3. Status Dependencies

Status dependencies set the constraints between the states of different negotiation sequences. Given the fact that the relations among negotiation sequences can be completely independent of the negotiated tasks, the negotiation process can be perceived as any other interaction that takes place in a time frame.

Considering the function *status*(*t*, *s*, *a*, *s*) with values in the set *Status* \in {*initiated*, *undefined*, *success*, *failure*}; we can say that *status*: $E_{\Phi 4} \rightarrow Status$ where $E_{\Phi 4}$ is the graph of Cartesian product $\mathcal{T} \times S \times S$ that meets the relation Φ_4 such that: \forall (t, si, sj) \in T × S × S, exists Φ 4(t, si, sj) if and only if si, sj \in S, where si denotes the initial negotiation sequence.

For instance: status(t, s0, 2, s2) returns the state of the negotiation proposal in the negotiation node 2 initiated in sequence s0 and visible in s2 at the time instance t.

In the following example, with *si* a sequence in a negotiation Nx and *sj* a sequence in the negotiation Ny, the status dependency rule shows that if the negotiation Nx is finalised at the time instance T1, with either a failure or success, a new negotiation Ny will start at the next moment of time. This rule, based on the Allen's

meets operator concept (Allen and Hayes 1985), allows launching a new negotiation at the service level with no intervention from the Neg A or the user.

end_start_successively (T1, T2, si, sj): $T1 \in T$ $T2 \in T$;

 $\exists a \in si(T1) \quad \forall b \in sj(T2)$

 $status(T1, si, a, si) = failure \lor status(T1, si, a, si) = success \rightarrow \bullet status(T2, sj, b, sj) = initiated$

5.4. Attribute Dependencies

The attribute dependency sets the constraints among the attribute values of the negotiated tasks in two or more sequences. This type of dependence helps in maintaining coherence between the existing proposals at a time instance and future proposals. Dependencies among attribute values of the negotiated tasks may establish several constraints:

- *Relational constraints*: the negotiated attribute values in dependent negotiation sequences must satisfy certain relations of equality or inequality;
- *Arithmetic constraints*: the negotiated attributes in dependent sequences of negotiation have values calculated according to the values of other attributes;
- *Cumulative constraints*: the negotiated attribute values are involved in relational and arithmetic constraints at the same time (e.g., the sum of the negotiated attribute values in two sequences involved in different negotiations must not exceed a certain value).

In the following example, the coordination rule leads participant p1 to reach an agreement in si sequence of the negotiation Nx if: 1) reach an agreement in sj sequence of the negotiation Ny; and 2) if in sequence si there is a proposal equals with the

negotiated task size in negotiation Ny; and 3) if this proposal is accepted by another participant through a sequence sk attached also negotiation Nx:

success_and_issues_complementary $(T_1, T_2, s_i, s_j, s_k)$: $T_1 \in \mathcal{T}$ $T_2 \in \mathcal{T}$;

 $\exists a \in si(T_1) \exists b \in sj(T_1) \exists a' \in si(T_2) a' = a$

 $status(T_1, sj, b, sj) = success \land issues(T_1, si, a, sk).size = issues(T_1, sj, b, sj).size \land status(T_1, si, a, sk) = success \rightarrow status(T_2, si, a', si) = success$

5.5. Role Dependencies

The role dependencies set the constraints among the roles of one or more participants. This type of dependency is used in coordination rules aimed at managing relationships among multiple negotiations. For example, a participant p1 involved, at a given time, in several negotiations for several tasks (with *guest* role), has the local politics to accept only one task: if a status sequence is a *success* then all negotiation proposals of the other sequences will switch to status *failure*:

success_competition(T_1, T_2, S_i, S_j): $T_1 \in \mathcal{T}$ $T_2 \in \mathcal{T}$ $S_i \in sequences(T_1, p1)$ $S_j \in sequences(T_1, p1)$;

 $\exists a \in Si(T_1) \ \forall b \in Sj(T_1) \ \exists b' \in Sj(T_2) \ b' = b$

 $status(T_1,S_i,a,S_i)=success \land role_s(T_1,S_i)=guest \land \neg status(T_1,S_j,b,S_j)=success \land role_s$ $(S_j,p_1)=guest \rightarrow \bullet status(T_2, S_j, b', S_j)=failure$

The next section details how these constraint rules can be bundled in order to define coherent negotiation tactics that can be coordinated in parallel. It also presents an example of tactics allowing to coordinate the proposals among multiple participants involved in multiple negotiations (i.e., negotiation in open environment).

6. Coordination Pattern

The proposed coordination model should allow the infrastructure to manage in a coherent manner multiple coordination rules for a participant involved in several parallel negotiations. In this respect, the proposed model must be distributed over several coordination modules, in line with the characteristics of our distributed negotiation system approach. Hence, the *Coordination Pattern* defines a set of coordination rules used for the implementation of the coordination mechanism, having the following representation:

PattCoor = (TriggerSet, RulesSet, ProgramFormulaSet)

TriggerSet represents a set of trigger conditions that must be met at the same time instance allowing to apply the coordination pattern.

RulesSet represents a set of coordination rules which the coordination pattern is required to synchronize.

According to the proposed model, the coordination rules are set and globally represented (i.e., visible at several sequences level). Furthermore, in order to be managed by a single negotiation sequence, these rules are split into coordination policies locally represented (i.e., visible at a single sequence level).

To model the negotiation process, the authors introduced the *Program Formula* defined by a set of methods that model the coherent execution of a negotiation. Also, a negotiation sequence has been defined as the image of the evolution of a negotiation according to the rules that the sequence must meet. Thus, to link the image of a negotiation through a sequence and the methods that model this image, the *ProgramFormulaSet* will be also introduced.

ProgramFormulaSet represents a set of *Program Formula*, one for each negotiation sequence involved in managing the coordination rules.

Consequently, the coordination pattern represents a set of coordination rules which depend on the chosen negotiation tactic.

The following example presented in the next subsection shows how to use the proposed coordination model based on a negotiating tactic on the transport coordination of the two tasks.

6.1.Coordination in an Open Environment through a Set of Bilateral Negotiations

The coordinating negotiations process in an open environment has two important characteristics: i) The new dependencies among the negotiations coordinated in an open environment are not in contradiction with the dependencies managed by the coordination patterns in a closed environment; ii) The proposed model refers to dependencies between bilateral negotiations of different negotiations or direct, among the negotiations composed of multiple bilateral negotiations taking place at the same time. Therefore, considering a participant involved in several parallel negotiations, the coordination pattern in an open environment has to manage: a) attribute dependencies that model the new proposals, taking into account the submitted/received proposals during several negotiations; b) status dependencies that decide to continue or not some negotiations, taking into account the status of other negotiations; c) role dependencies

To illustrate how the set of bilateral negotiations is coordinated, a scenario is proposed where there are at least two parallel negotiations conducted by the same organization (i.e., participant P1). In this regard, the scenario proposed in section 8 involves three SMEs, two in the metalworking area and the other one in the transport area. The three SMEs are involved in negotiations on the supply of raw materials. In the case of raw material tasks, transportation is also part of the contract and, therefore, a SME may have contracted tasks to be delivered at a certain time and in an identical or very close geographical area. Consequently, the transport SME manager wants to link the two negotiations to minimize transport costs, using a single transport car for the two deliveries. The manager should handle the two negotiations in parallel, therefore, the deliveries must take place in close time intervals, the size of the two tasks should allow to share the same car and the two negotiations must end with contracts. If the manager would be able to conclude agreements in both cases, he would be obliged to coordinate the tasks' transport to provide a single transport of the raw materials.

Moreover, this scenario illustrates how the proposed approach deals with the issue of coordination of two parallel negotiations by performing the following actions:

First, in each of two parallel negotiations, the invited SMEs will propose and accept proposals according to the optimal allocation of their own resources;
Second, the two parallel negotiations should satisfy a common constraint (i.e., the size of both transportation tasks must be complementary enough to fit the same truck) according with the initiator SME's objective.

Next, the authors present the dependencies and the modality in which the two negotiated tasks are managed by the *Transport* tactic and the proposed coordination pattern.

TriggerSet

For this type of coordination pattern there are no constraints on the role the SME plays in the current negotiations. Therefore, the two different negotiations must meet only the constraints related to the status of negotiations and the attributes of negotiated tasks.

The last two expressions set the fact that the two negotiations require raw material providers and a delivery in a nearby geographical area, therefore it is mandatory for the

attribute identifying the raw providers and the delivery destinations to have similar values in both negotiations. The common ontology *O* may specify the "location" concept in terms of GPS coordination or street, city, region names etc.

 $(p1 \in participants(t,N1))$ $(p1 \in participants(t,N2));$ (view(s11)=(p1, N1, R11)) (view(s21)=(p1, N2, R21)); $(\exists a \in s11(t1), \exists b \in s21(t) :$ $issues(t,s11,a,s11).rawLocation = issues(t,s21,b,s21).rawLocation \land$ issues(t,s11,a,s11).deliveryLocation = issues(t,s21,b,s21).deliveryLocation)

RulesSet

The proposed coordination pattern aims at managing two parallel negotiations for the participant P1. The common transportation scenario objectives have direct implication on the negotiated attributed other than the location. Taking in consideration only the delivery location constraint, the coordination should guarantee not only the location but also that the deliveries should be made in close time frames and the size of both tasks to be complementary enough in order to fit the same transport (i.e., the same truck). Assume that the maximal transport size P1 is able to do is of 1000kg; thus, in both N1 and N2 negotiations, the dependency is given by the fact that the sum of the sizes of the two tasks is lower than the maximal delivery size. Therefore, the main rules proposed below - rules a) and b) - describe the dependencies among the proposals made in both negotiations in order to meet the constraints of a possible synchronization of a common delivery for the two tasks. This synchronization requires a coordination among the assertions made for both attributes *size* and *delayTransp* in the two negotiations. Defining *Size1* and *Size2* as being data structures that establish the complementary values of the attributes *Size*, the following rules can be described:

a) transp_coordination($T_1, T_2, s11, s21$): $T_1 \in \mathcal{T} \ T_2 \in \mathcal{T}$;

 $\forall a \in s_{11}(T_1) \ \forall b \in s_{21}(T_1) \ \exists c \in s_{21}(T_2)$ issues(T₁,s11,a,s11).size = Size1 \land -issues(T₁,s21,b,s21).size=Size2 \rightarrow • issues(T₂,s21,c,s21).size = Size2 \land issues(T₂,s21,c,s21).delayTransp = issues(T₁,s11,a,s11).delayTransp;

Consequently, if in the sequence s11 of the participant p1 involved in the negotiation N1, there is a proposal described by a set of attributes with value Size1 for the size of the task, and if in sequence s21 of the same participant p1 in the negotiation N2, there is no proposal described by a set of complementary values Size2, then a new proposal will be created in the sequence s21, which will contain a complementary proposal.

b) transp_coordination(T_1 , T_2 , s21, s11): $T_1 \in T$ $T_2 \in T$; $\forall a \in s_{21}(T_1) \forall b \in s_{11}(T_1) \exists c \in s_{11}(T_2)$ issues(T_1 , s21, a, s21).size=Size1 $\land \neg$ issues(T_1 , s11, b, s11).size=Size2 $\rightarrow \bullet$ issues(T_2 , s11, c, s11).size = Size2 \land issues(T_2 , s11, c, s11).delayTransp = issues(T_1 , s21, a, s21).delayTransp;

It is the same rule as before, but this time the proposal is automatically created in the sequence s11 of the negotiation N1.

The coordination pattern proposed below establishes hard relationships such that, if one negotiation fails, the other is stopped - rules c) and d) proposed below:

c) fail_complementary_hard(T_1, T_2, s_{11}, s_{21}): $T_1 \in T \ T_2 \in T$; $\forall a \in s11 \ (T1) \ \forall b \in s21 \ (T1) \ \exists b' \in s21 \ (T2) \ b' = b$ $status(T1,s11,a,s11)=failure \ \land issues(T1,s11,a,s11).size = Size1 \ \land$ $issues(T1,s21,b,s21).size=Size2 \ \land issues(T1,s21,b,s21).delayTransp =$ $issues(T1,s11,a,s11).delayTransp \rightarrow e \ status(T2,s21,b',s21) = failure;$ $d) fail_complementary_hard(T_1, T_2, s_{21}, s_{11}) : \ T_1 \in T \ T_2 \in T;$ $\forall a \in s11 \ (T1) \ \forall b \in s21 \ (T1) \ \exists b' \in s21 \ (T2) \ b' = b$ $status(T1,s11,a,s11) = failure \land issues(T1,s11,a,s11).size = Size1 \land$ $issues(T1,s21,b,s21).size = Size2 \land issues(T1,s11,a,s11).delayTransp =$ $issues(T1,s21,b,s21).delayTransp \rightarrow status(T2,s21,b',s21) = failure;$

Therefore, if the negotiation N1 ends with a failure in a proposal, then the negotiation N2 is stopped in the proposals containing the complementary *size* and *delayTransp* values.

Considering that in the two negotiations the rules are the same, this coordination pattern will be implemented through the instantiation of two identical *Transport* services, one for each negotiation involved in this coordination. When the service will be instantiated in a negotiation as a new tactic to follow, a new negotiation sequence will start. The coordination policies of the new sequence, called sequence *Transport* involved in this coordination mechanism will be further elaborated (denote *Transp* for simplifying).

Policy for sequence STransp1:

The sequence $s_{Transp1}$ – created by the instantiation of Transport service in negotiation N1 - maintains the coordination between the external representations for the sequence s_{11} from N1 (i.e., the sequence s_{11} allows to continue the negotiation N1 with other tactics and, consequently, with proposals without the transportation constraints), and another sequence of type Transp, $s_{Transp2}$ - currently in negotiations N2.

The transcription of the general Transport tactic rules described above for the sequence s_{Transp1} is as follows:

transp_coordination($T_1, T_2, s_{Transp1}, s_{Transp2}$): $T_1 \in \mathcal{T}$ $T_2 \in \mathcal{T}$; $\forall a \in s_{Transp1}(T_1) \exists a' \in s_{Transp1}(T_2)$: a' = a; $issues(T_1, s_{Transp1}, a, s_{Transp2}).size = Size1 \land \neg issues(T_1, s_{Transp1}, a, s_{Transp1}).size =$ $Size2 \rightarrow \bullet issues(T_2, s_{Transp1}, a', s_{Transp1}).size = Size2 \land issues(T_2, s_{Transp1}, a', s_{Transp1}).delayTransp = issues(T_1, s_{Transp1}, a, s_{Transp2}).delayTransp$

This means that a proposal received in the sequence $s_{Transp2}$ containing a task size *Size1*, generates a complementary proposal in the sequence $s_{Transp1}$.

transp_coordination(T_1 , T_2 , $s_{Transp1}$, s_{11}): $T_1 \in TT_2 \in T$; $\forall a \in s_{Transp1}$ (T_1) $\land \exists a' \in s_{Transp1}$ (T_2) : a' = a; $issues(T_1, s_{Transp1}, a, s_{11}).size=Size1 \land \neg issues(T_1, s_{Transp1}, a, s_{Transp2}).size=Size2 \rightarrow \bullet issues(T_2, s_{Transp1}, a', s_{Transp1}).size=Size1 \land issues(T_2, s_{Transp1}, a', s_{Transp1}).delayTransp = issues(T_1, s_{Transp1}, a, s_{11}).delayTransp \land issues(T_2, s_{Transp1}, a', s_{Transp1}, a', s_{Transp1}).size=Size2$

In other words, a proposal received in the sequence s_{11} containing a task size *Size1*, generates in the sequence $s_{Transp1}$ a complementary proposal for the external representation of the sequence $s_{Transp2}$ creating, thus, a new proposal in the negotiation N2.

fail_complementary_hard($T_1, T_2, s_{Transp1}, s_{Transp2}$): $T_1 \in \mathcal{T} \ T_2 \in \mathcal{T}$; $\forall a \in s_{Transp1} \ (T_1) \land \exists a' \in s_{Transp1} \ (T_2)$: a' = a; $status(T_1, s_{Transp1}, a, s_{Transp2}) = failure \rightarrow \bullet \ status(T_2, s_{Transp1}, a', s_{Transp1}) = failure$

Hence, a failure in the sequence *s*_{*Transp*2} automatically generates a failure in the sequence *s*_{*Transp*1}.

fail_complementary_hard($T_1, T_2, s_{Transp1}, s_{11}$): $T_1 \in \mathcal{T} \ T_2 \in \mathcal{T}$; $\forall a \in s_{Transp1} \ (T_1) \land \exists a' \in s_{Transp1} \ (T_2)$: a' = a; $status(T_1, s_{Transp1}, a, s_{11}) = failure \rightarrow \bullet status(T_2, s_{Transp1}, a', s_{Transp1}) = failure$

Therefore, a failure in the sequence s_{11} automatically generates a failure in the sequence

STransp1.

The coordination policy for the sequence $s_{Transp2}$ has the same structure as the coordination policy for the sequence $s_{Transp1}$, described above; the only difference is the fact that the sequence $s_{Transp2}$ keeps the external representations of the sequence $s_{Transp1}$ and mirrors the sequence s_{21} from the negotiation N2.

According to the proposed approach, a negotiation can conduct multiple tactics handled by the different services. Considering the sequence s_{11} as the sequence corresponding to the services initiating a negotiation (i.e., Outsource and Insource), its policy will be completed with the following rule:

Policy for sequence s11:

fail_complementary_hard(T_1 , T_2 , s_{11} , $s_{Transp1}$): $T_1 \in TT_2 \in T$; $\forall a \in s_{11}(T_1) \land \exists a' \in s_{11}(T_2)$: a' = a; $status(T_1, s_{11}, a, s_{Transp1}) = failure \rightarrow \bullet status(T_2, s_{11}, a', s_{11}) = failure$

For the sequence s_{11} , the coordination mechanism consists of a synchronisation between the sequences $s_{Transp1}$ and s_{11} , in order to stop pursuing the negotiation with the proposals that were created by or for the sequence $s_{Transp1}$.

ProgramFormula

As has been shown in the subsection 5.1, the various *Program Formula* translate the coordination policies described above into methods that model the content of the corresponding negotiation graphs in a set of sequences based on the instantiated coordination services. Furthermore, the behaviour of a sequence is characterized by a set of methods that defines the coordination policy corresponding to a negotiation tactic. This flexible and decentralized structure of the coordination mechanism in several separate services allows taking into account the predefined dependencies, as well as

dynamically-defined dependencies. The coordination rules presented in this work define the functionalities of the coordination services within a negotiation: i) *validation* of the received or submitted proposals in a negotiation. A valid proposal is a proposal that meets the dependency relations managed by the coordination modules involved in the current negotiation; ii) *creation* of new proposals to meet the dependency relations. The proposals are created if all constraints are satisfied in an environment in which all information is known and, therefore, the coordination modules must not make any nondeterministic choice. Therefore, the main advantage of this coordination mechanism is the ability to dynamically maintain a representation of a negotiation in line with the dependencies established in the current negotiations.

7. Validation of the Negotiation Mechanism

The negotiation mechanism has been experimented and validated within an aerospace environment, for the enhancement of the interoperability between space mission design teams at ESA (Coutinho 2012). This scenario gathered in a single room, teams with sometimes more than 50 experts. The results of the validation showed that the interoperation between the design teams improved. Thus, the interoperability recovery time (in minutes) was much higher using the negotiation mechanism; the number of people was small (due to the time taken doing a negotiation), as the number of experts grew and the complexity of the interoperation grew geometrically with the formula:

Channels =
$$\frac{N \cdot (N-1)}{2}$$
 (Figure 2):

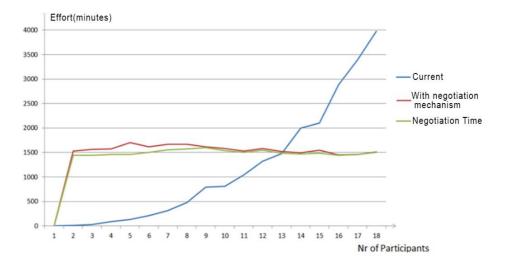


Figure 2. Validation of the negotiation mechanism in an interoperability environment The above graph plots one of the experiments, showing that the application of the proposed negotiation mechanism in the tested environment started to be beneficial after gathering more than 13 people in a room for negotiating. Furthermore, this approach is being applied to the H2020 project C2NET (section 8), and the validation within this scenario is ongoing.

8. Application to the Manufacturing Industry

The results of the proposed research are being applied in the manufacturing industry with the support of the European Project C2NET (H2020 project n° 636909). The overall objective of C2NET is to support supply networks, optimizing the utilization of manufacturing and logistic assets, and providing the services to facilitate collaborative demand, production and delivery planning. With a special focus on SMEs, C2NET is proposing a cloud-based system to manage such optimization and collaboration.

The coordination negotiation work presented in this article is being applied in a specific pilot case where the model-driven negotiation methodology is complementing the C2NET solutions and supporting the creation of a collaborative transportation plan (CTP) involving two SME metalworking companies (A and B) from the north of

Portugal. Located in the industrial region of Ave Valley in the North of Portugal, both companies have their own specificities and needs, but they have some common problems, which could be mitigated though improved collaboration with each other and with other companies in the same industrial park. A scenario of business-to-business is presented, illustrating the interaction among partners sharing networked resources in a collaborative environment. Each enterprise has its own business process, but at certain times, the proposed negotiation system provides support for interoperability among partners which require to share resources, (e.g., sharing transport to/from common locations to decrease costs).

Both companies struggle with the logistics sector, dealing with the high prices related with transportation both on acquiring raw materials and on delivering their finished goods. When the material to be transported has wide relation volume/weight as the case of steel, it will increase the price to pay. Steel is a heavy material, so it will only occupy a small volume of the space when transported, hence the truck will be almost empty but close to its maximum load capacity. Company A usually purchases huge coils with more than 20 tons while Company B usually purchase small coils with less than 5 tons. Also, since both companies do not want to have raw materials in stock, they must accurately plan their production strategies, having to manage new setups of the machinery when delays in the delivery of supplies occur (Cunha et al 2017). This presents a candidate case for collaboration in transport and negotiation with the transporter company, where the negotiation parameters are the *price*, *transportation size* and the *delivery time*. As identified in the table below (Table 1), C2NET supports this directly process in a number of steps (1-6, 8) enabling a company to provide a transportation plan open to the network of partners, its constraints in terms of time, transportation size, price and location, and the work presented in this article acts mostly

on step 7, managing the negotiation with the transportation company.

Activities	Steps
Manage Collaborative Transportation Plan (CTP)	1. Import transportation plan to C2NET
	2. Analyse is compatible CTPs are available
	3. Create/Subscribe a an open CTP
Manage Subscriptions	4. Define Subscription rules
	5. Notify subscriptions
	6. Close CTP
Manage Transportation	7. Transportation Negotiation (focus of the
	work)
	8. Notify negotiation result

Table 1. CTP Main Activities (from C2NET deliverable D7.4)

The scenario demands the establishment of collaborative strategies in step 9, between networked companies that are geographically closely located and the transportation company, enhancing the process displayed in Figure 3. Collaborative transportation will allow companies to contract a full transport, sharing the same truck when the time constraint is met, and dividing costs.

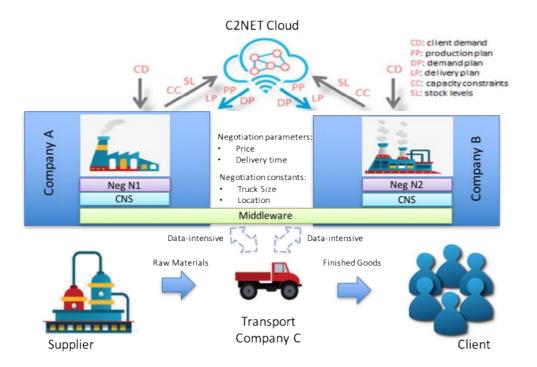


Figure 3. C2NET Collaborative Transport Scenario (white dashed arrows represent

flows of intensive data)

Based on customer demands, production or delivery plans, interoperability needs to be maintained to support the coordination of companies A, B, and C, which need to negotiate in order to tackle their demands and respect their constraints.

Considering the scenario described in Section 6.1., the manager of the participant P1 manages the two negotiations (N1, N2) in parallel. The negotiation constraint refers to the two negotiations that must end with a success in order to conclude the two contracts (i.e., the size of both tasks must be complementary enough in order to fit the same truck). The proposed architecture manages the *Transport relations* using two instances of negotiation services and a data structure, such as: i) The two instances of *Transport* service communicate each other through a shared data structure (Blackboard), denoted *Transp Space*; ii) The instances of *Transp service* can read and write on the *Transp Space* the information about the negotiation attributes (e.g., *price, transportation size, delayTransp, location*).

The datasets needed for the illustrated negotiation are managed in C2NET within the STables model, including information such as: Customer, Supplier, Part, Order, Site etc.

The following subsection presents two examples showing how the proposed infrastructure may conduct two parallel negotiations. The negotiation on transport coordination within the project scenario is illustrated in the second example.

8.1 Negotiation Examples

The first example

The following example provides the evolution of a negotiation in terms of proposals and counter-proposals modelled by a bicolored graph in which white nodes, representing *negotiation contexts* (i.e., the negotiation attributes with associated values), and black

nodes, representing *decision points* with multiple alternatives.

The negotiation graph is built using the verbs of the XPLORE protocol (e.g., Connect, Open, Assert, Request, Ready and Quit). The CONNECT verb allows to dynamically involve a new participant in a negotiation. The OPEN and ASSERT verbs enable a participant to build the negotiation graph, by creating and populating context nodes with information about the negotiation state at these nodes. The REQUEST verb avoids DeadLock situations by allowing participants to express their information needs on some given terms of the negotiation in order to proceed in the negotiation. The READY and the QUIT verbs allow a participant to declare respectively, that he is "ready to sign a contract" in the state of a given negotiation context, or, on the contrary, that he wants to give up the negotiation at that state (but he may pursue the negotiation in other branches).Therefore, it is considered that the transport company C initiates a negotiation on a transport job. The negotiation attributes are: *price, size, delayTransp, location*.

Therefore, the company C sells transport and wants to have the transportation *size* in units (<1000) for each customer and *location* in City1 fixed. The seller wants to have the reservation price 50/unit and *delayTransp* <2.

There are two customers: company A and company B, each seeking for the product *job*. The company B wants to send in City1 x units with reservation price 54/unit and delayTransp = 1.

The company A wants to send in City1 y units with reservation price 56/unit and delayTransp = 1.

Assuming that negotiating agents B and A have a simple proposal generation mechanism starting from a minimal cost estimate 40/unit, then it increments with a fixed step (e.g., the bid increment of 6 for B, respective the bid increment of 8 for A) until the proposal is accepted or reached a maximum acceptable value (54 for B and 56 for A).

Table 2 shows how the two negotiations are managed in parallel by the initiator C:

Table 2. Negotiation steps for two negotiations managed in parallel by an initiator C

Negotiations between the initiator C and two guests B and A 1. C : Open node 0 (white) ; assert(size<1000, location=City1, delayTransp <2); request(price); request (delayTransp) 2. C: Open node 1 (black)	
3. C: Open node 2 (white); connect (2, companyB)	(3') C: Open node 3 (white); connect (3, companyA)
4. B: Open node 4 (black)	(4') A: Open node 8 (black)
5. B: Open node 5 (white) from 4; assert	(5') A: Open node 9 (white) from 8; assert
(price=40; size = 800; delayTransp=1)	(price=40; size = 950; delayTransp =1)
6. B: Open node 6 (white) from 4; assert	(6') A : Open node 10 (white) from 8; assert
(price=46; size = 800; delayTransp=1)	(price=48; size = 950; delayTransp =1)
7. B: Open node 7 (white) from 4; assert	(7') A : Open node 11 (white) from 8; assert
(price=52; size = 800; delayTransp=1)	(price=56; size = 950; delayTransp =1)
8. C, B: Ready node 7	(8') C, A: Ready node 11
9. C: Quit node 5	(9') C: Quit node 9
10. C: Quit node 6	(10') C: Quit node 10

As it can be seen in Table 2, the companies C and B negotiate starting with node 2 and the companies C with A negotiate starting with node 3.

Figure 4 represents the negotiation graph for the initiating partner, company C, while

figure 5 and figure 6 represent the negotiation graphs for the guests, company B,

respective, company A.

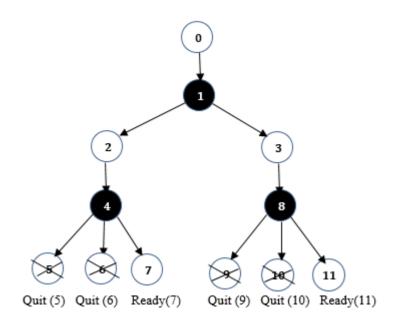


Figure 4. Negotiation graph for the company C

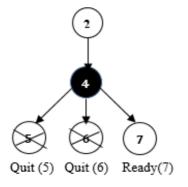


Figure 5. Negotiation graph for the company B

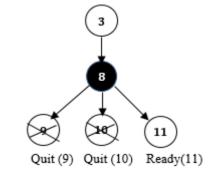


Figure 6. Negotiation graph for the company A

This example shows how two parallel negotiations can be managed and coordinated by the proposed negotiating infrastructure. First, the negotiation N1 between the agents C and A ended with success. For the negotiation N1, the highest accepted bid was 56 and this value is higher than the reservation price 50 of the negotiation agents C. Second, the negotiation N2 between the agents C and B ended with success. For the negotiation N2, the highest accepted bid was 52 and this value is higher than the reservation price 50 of the negotiation N2, the highest accepted bid was 52 and this value is higher than the reservation price 50 of the negotiation N2, the highest accepted bid was 52 and this value is higher than the

Therefore, this is an example of a win-win negotiation in which agents try to get a fair deal to maximize the gain for everyone involved.

The second example of transport negotiation

The negotiation starts when an instance of *Outsrc* service invites an instance of the *Insrc* service and also invites an instance of the *Transp* service in the negotiation process with the purpose to fill as much as possible the transportation truck (i.e., the transportation size should be as close as possible to 2000 units). For the initiator, the instance of the *Transp* service updates the *Transp Space* (blackboard) with the new negotiating context in which it is envisaged a synchronization of transport relation. Further, the *Transp* instance seeks and adds in the blackboard, the negotiation contexts that allow a transport synchronization.

In this example, the initiator C begins two separate negotiations both involving an instance of the *Transp* service.

Thus, the negotiation between the initiator C and the guest B is the following:

- C: Open node 0 (white); assert(size<=2000, location=City1, delayTransp <2);
 request(size); request(price); request (delayTransp)
- (2) C: Open node 1 (black)
- (3) C: Open node 2 (white); connect (2, companyB, serviceTransp)
- (4) B: Open node 3 (black)
- (5) B: Open node 4 (white) from 3; assert (size = 600)
- (6) B: Open node 5 (white) from 3; assert (size = 800)
- (7) B: Open node 6 (black) from 2
- (8) B: Open node 7 (white) from 6; assert (price=46; delayTransp=1)
- (9) B: Open node 8 (white) from 6; assert (price=40; delayTransp=1)

- (10) C: Open node 9 (black) from 5 and 7
- (11) C: Open node 10 (white) from 9; with attribute values (size = 800; price=46; delayTransp=1)

According to the properties of the negotiation graph, the white node 10 inherits the values of the attributes from its parent white nodes.

- (12) C: Open node 11 (black) from 4 and 8
- (13) C: Open node 12 (white) from 11; with attribute values (size = 600; price=40; delayTransp=1)

At this point, we can consider that in the second negotiation with the partner A, the *Transp* service is making a proposal with size=900. Therefore, the *Transp* service from that negotiation is adding the info on the *Transp Space*. Consequently, the *Transp* service from the current negotiation is seeing it and is creating a new complementarity proposal in the node 13 (size = 1100) to reach the maximum size of 2000 units.

- (14) C, serviceTransp: Open node 13 (white) from 3; assert (size = 1100)
- (15) C, serviceTransp: Open node 14 (white) from 6; assert (price=52; delayTransp=1)
- (16) C: Open node 15 (black) from 13 and 14
- (17) C: Open node 16 (white) from 15; with attribute values (size = 1100; price=52; delayTransp=1)
- (18) C, B: Ready node 16
- (19) C: Quit node 10
- (20) C: Quit node 12

Figure 7 shows the graph of the negotiation between C and B:

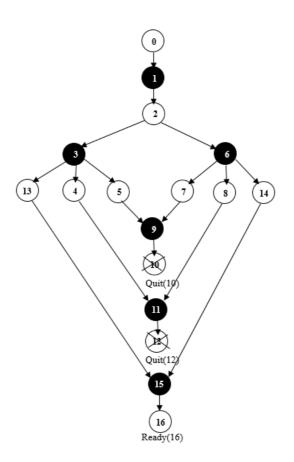


Figure 7. The graph of the negotiation between C and B

The negotiation between the initiator C and the guest A is the following:

- C: Open node 0 (white); assert(size<=2000, location=City1, delayTransp <2);
 request(size); request(price); request (delayTransp)
- (2) C: Open node 1 (black)
- (3) C: Open node 2 (white); connect (2, companyA, serviceTransp)
- (4) A: Open node 3 (black)
- (5) A: Open node 4 (white) from 3; assert (size = 700)
- (6) A: Open node 5 (white) from 3; assert (size = 900)
- (7) A: Open node 6 (black) from 2
- (8) A: Open node 7 (white) from 6; assert (price=53; delayTransp=1)
- (9) A: Open node 8 (white) from 6; assert (price=48; delayTransp=1)
- (10) C: Open node 9 (black) from 5 and 7

(11) C: Open node 10 (white) from 9; with attribute values (size = 900; price=53; delayTransp=1)

The *Transp* service from this negotiation is seeing the proposal with size=900 made by A and is adding the info on the *Transp Space*, therefore, the *Transp* service from the negotiation with B is creating a new proposal with the complementary task size (size = 1100).

- (12) C: Open node 11 (black) from 4 and 8
- (13) C: Open node 12 (white) from 11; with attribute values (size = 700; price=48; delayTransp=1)

Similarly, the *Transp* service is seeing the proposal made in node 12 and is adding this info on the Transp Space.

(14) C, A: Ready node 10

At the same time, when a proposal is accepted, all other proposals in the other white nodes will be denied.

(15) C: Quit node 12

Figure 8 shows the graph of the negotiation between C and A:

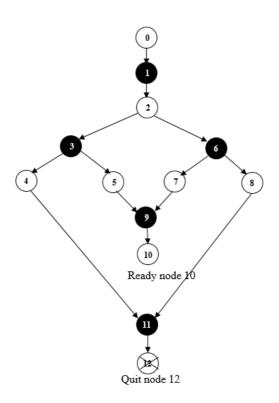


Figure 8. The graph of the negotiation between C and A

The following subsection provides a roadmap that shows how the proposed solution will be implemented within a manufacturing project.

8.2. The roadmap for implementation in Manufacturing project

The proposed solution consists in the development of a centralized negotiation platform for manufacturing projects with the following requirements for implementation based on the proposed negotiation stages:

a) Initialization

A client C already registered wants to be able to negotiate its job, for example: How can I negotiate a transport job (ex. size, price, delay etc.)? How can I create and configure the communication media? How can I identify myself? The initialization of a negotiation in the proposed solution supposes to be able to start and to configure the negotiation framework. The negotiation clients software will require to have access to an API allowing them to identify themselves, to initiate the negotiation services and to define the attributes and values. Behind this API, we should have our main negotiation and coordination services.

As communication media, we will develop a solution allowing an asynchronous communication. In the conceptual proposal, we have pointed out a solution based on blackboard but other solutions will be considered (i.e., publish/subscribe open source solutions).

b) Selection of partners

The client wants to interact with all potential partners:

How can Broker service ask clients to connect to a conversation?

Before engaging in a negotiation, the platform should ensure the selection of the potential participants. A first implementation will be based on broadcast solutions, but depending on the performances and scalability aspects we will provide the implementation of a client database with search and select facilities.

c) Negotiation and finding a solution to close the negotiation

The client starts a negotiation to find out the following information:

What are the proposals for the job (ex the price of transport)?

How my acceptance rules will be verified?

For the negotiation phase, the main implementation will consist in providing the data structures allowing to construct the negotiation graphs. A first implementation will consist in shared in memory data structures among independent workers each with their coordination and ending of the negotiation rules.

9. Final Considerations and Future Work

The main idea of this research is based on negotiation as a key solution for dealing in a short time with any change that may occur in a collaborative working environment leading to the breaking of interoperability. Based on negotiation, the various parties involved can express their preferences about the chosen negotiation strategy, the attributes of the negotiation object, the definition of the contract, the negotiation data and terms. Therefore, by negotiation, the parties can reach a common agreement ensuring the restoration of interoperability through a unanimous decision leading to a seamless collaboration across time.

With respect to this, the proposed research defines a generic coordination model-driven platform described by a methodical set of rules and patterns to handle parallel and concurrent negotiations. In the context of the interoperability solving problem, we have distributed the solution search between several services, each with their predefined set of coordination rules. This distribution allows advancing in the space of possible solutions in different parallel directions until a solution satisfying the different constraints are jointly met. The fact that the coordination model is composed of different services makes it very flexible, allowing to be utilized for simple interoperability negotiations (i.e., bilateral negotiation on simple attributes and with limited number of interactions) up to very complex interoperability problems (i.e., multiple participants, complex attributes and dependent or nested negotiations).

The issues of the parallel negotiations are addressed by the proposed synchronization mechanisms allowing to choose the winners from a set of participants as well as to ensure that each participant is making the choices of negotiation proposals depending on his optimal allocation of resources. The successful interoperability solution is found and committed only through negotiation objects proposal and the coordination constraints, and rules are never disclosed among the participants. In fact, the proposed coordination model preserves the participants' autonomy, allowing the model to be used for solving interoperability aspects inside an enterprise or between enterprises in a competitive environment. Finally, the coordination model is independent of the communication and decision frameworks, allowing an easy integration with different communication platforms or with different decision-making solutions. Experimentation within an aerospace environment has been conducted and a C2NET collaborative transport scenario is outlined to prove the viability of the model. Future work relates to the description and implementation of other negotiation tactics specific to different industrial areas. More advanced studies will be further integrated in the negotiation process objectives not only to find an interoperability solutions but also to estimate the performance value for each solution, along with its stability in time.

Acknowledgment

The authors wish to acknowledge the support of the European Commission through the funding of the H2020 C2NET project for their support, interaction and contribution in the development of the business case that is presented on this paper.

References

- Agostinho, C., and R. Jardim-Goncalves. 2015. "Sustaining Interoperability of Networked Liquid-Sensing Enterprises: A Complex Systems Perspective."
 Annual Reviews in Control 39: 128–143. doi: 10.1016/j.arcontrol.2015.03.012.
- Allen, J.F., and P.J. Hayes. 1985. "A common-sense theory of time." In Proceedings of the Ninth International Joint Conference on Artificial Intelligence, Los Angeles, CA, pp. 528-531.
- Alrayes, B., O. Kafali, and K. Stathis. 2016. "RECON: A Robust Multi-agent Environment for Simulating COncurrent Negotiations." *Recent Advances in*

Agent-based Complex Automated Negotiation, Studies in Computational Intelligence 638: 157-174. doi:10.1007/978-3-319-30307-9_10.

- Andres, B., Sanchis, R., Poler, R., Saari, L. (2017) A proposal of standardised data model for cloud manufacturing collaborative networks. IFIP Advances in Information and Communication Technology. Volume 506, 2017, Pages 77-85
- Badica, C., S. Ilie, M. Kamermans, G. Pavlin, and M. Scafes. 2011. "Using Negotiation for Dynamic Composition of Services in Multi-organizational Environmental Management." *Environmental Software Systems. Frameworks of eEnvironment. IFIP Advances in Information and Communication Technology* 359: 177-188. doi: 10.1007/978-3-642-22285-6_20.
- Benaben, F., Boissel-Dallier, N., Pingaud, H., & Lorre, J.-P. 2013. "Semantic issues in model-driven management of information system interoperability."
 International Journal of Computer Integrated Manufacturing, 26(11), 1042–1053.
- Caillere, R., S. Arib, S. Aknine, and C. Berdier. 2016. "A Multiagent Multilateral Negotiation Protocol for Joint Decision-Making." *Recent Advances in Agentbased Complex Automated Negotiation, Studies in Computational Intelligence* 638: 71 - 88. doi: 10.1007/978-3-319-30307-9_5
- Coutinho, Carlos. 2012. "NEGOSEIO: Framework for the Sustainability of Modeloriented Enterprise Interoperability." PhD thesis, Universidade Nova de Lisboa.
- Cretan, A., C. Coutinho, B. Bratu, and R. Jardim-Goncalves. 2012. "NEGOSEIO: A Framework for Negotiations toward Sustainable Enterprise Interoperability." IFAC Journal Annual Reviews in Control 36 (2): 291-299. doi: 10.1016/j.arcontrol.2012.09.010
- Cunha, A., J.P. Mendonca, P. Silva, M. Marques, C. Agostinho. 2017. "Metalworking SMEs Network Resources Optimization : C2NET Cases." In: 23rd ICE/IEEE International Technology Management Conference (ICE/ITMC). IEEE, Funchal, Portugal.
- Faratin, P., C. Sierra, and N. R. Jennings. 1998. "Negotiation Decision Function for Autonomous Agents." Journal of Robotics and Autonomous Systems 24 (3–4): 159–182.
- Fujita, K. 2016. "Automated Negotiating Agent with Strategy Adaptation for Multitimes Negotiations." *Recent Advances in Agent-based Complex Automated*

Negotiation, Studies in Computational Intelligence 638: 21-37. doi: 10.1007/978-3-319-30307-9_2

- Gelín, P., and A. W. Colombo. 2003. "Open Collaborative Control over Ethernet."
 Presented at the 3rd International Symposium on Open Control Systems,
 Towards Modular Self-Organizing Re-Configurable Automation Systems.
 Helsinki, Finland, September 9-10.
- Iacob, M. E., and H. Jonkers, 2009. "A model-driven perspective on the rule-based specification and analysis of service-based applications," *Enterprise information systems*, 3(3), 279-298. Doi: 10.1080/17517570903042762
- Izza, S. 2009. "Integration of industrial information systems: From syntactic to semantic integration approaches." *Enterprise Information Systems*, 3, 1–57.
- van Sinderen, M., and M. Spies, 2009. "Towards model-driven service-oriented enterprise computing," *Enterprise Information Systems* 3(3):211-217. Doi:10.1080/17517570903116442.
- Van Brussel, H., J. Wyns, P. Valckenaers, L. Bongaerts, and P. Peeters. 1998. "Reference architecture for holonic manufacturing systems: PROSA." *Computers in Industry* 37 (3): 255-274.
- Kent, S., 2002. "Model-driven engineering." Proceedings of the third international conference on integrated formal methods (IFM), 15–17 May 2002, Turku, Finland: LNCS 2335, Springer, 286–298.
- Lastra, J. L. M., E. Lopez Torres, and A. W. Colombo. 2006. "Engineering Framework for Agent-based Manufacturing Control." *IFAC Int. Journal Engineering Applications of Artificial Intelligence (EAAI)*.
- Larson, K., and T.Sandholm, 2005. "Mechanism Design and Deliberative Agents." Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems. The Netherlands, pp. 650-656
- Leão, J., and D. C. Morais. 2011. "A bilateral negotiation model for supply chain." IEEE International Conference on Systems, Man, and Cybernetics (SMC), Anchorage, AK, pp. 3201-3206, October 2011.
- Leitão, P., A.W. Colombo, and F. Restivo. 2006. "A Formal Specification Approach for Holonic Control Systems. The ADACOR Case." *The International Journal of Manufacturing Technology and Management (IJMTM)* 8: 37-57.
- Li, H., and H. Wang, 2007. "A multi-agent-based model for a negotiation support system in electronic commerce". *Enterprise Information Systems*, Volume 1(4).

- Lin, R., and S. Kraus. 2010. "Can Automated Agents Proficiently Negotiate with Humans?" *Communications of the ACM* 53(1): 78-88. doi: 10.1145/1629175.1629199
- Penders, A., G. Pavlin, and M. Kamermans. 2010. "A Collaborative Approach to Construction of Complex Service Oriented Systems." *Intelligent Distributed Computing IV. Studies in Computational Intelligence* 315: 55–66. doi: 978-3-642-15211-5_7
- Stojanov, Z., D. Dobrilovic, and J. Stojanov, 2018. "Extending data-driven model of software with software change request service," *Enterprise Information Systems*. 1-25, doi: 10.1080/17517575.2018.1445296.
- Terzić, B., V. Dimitrieski, S. Kordić (Aleksić), G. Milosavljevic, and I. Luković, 2017.
 "MicroBuilder: A Model-Driven Tool for the Specification of REST Microservice Architectures," *Enterprise Information Systems*, doi: 10.1080/17517575.2018.1460766
- Wang, J., J. Yang, J. Zhang, X. Wang, and W. Zhang, 2018. "Big data driven cycle time parallel prediction for production planning in wafer manufacturing," *Enterprise Information Systems*, 12, 1-19, doi:10.1080/17517575.2018.1450998.
- Wang, L., J. Shen, Q. Zhou and G. Beydoun, 2014 "Ant-Inspired Multi-phase and Multi-party Negotiations in the Data-Intensive Service Provision," 2014 IEEE International Conference on Services Computing (SCC), Anchorage, AK, USA, 2014, pp. 211-218. doi:10.1109/SCC.2014.36
- Wang, L., J. Shen, 2013. "Economical data-intensive service provision supported with a modified genetic algorithm," in *IEEE 2nd International Congress on Big Data*. Washington, DC, USA: IEEE Computer Society, 2013, pp. 358–365.
- Wooldridge, M., and N.R. Jennings. 1995. "Agent Theories, Architectures and Languages: A Survery." In *Intelligent Agents: ECAI-94 Workshop on Agent Theories, Architecture and Languages*, edited by Wooldridge, Michael J., Jennings, Nicholas R., 1-39.
- Yu, C., F. Ren and M. Zhang. 2013. "An adaptive bilateral negotiation model based on Bayesian learning". Studies in Computational Intelligence, 435 75-93.
- Zečević, I., P. Bjeljac, B. Perisic, S. Stankovski, D. Venus and G. Ostojic, 2018. "Model driven development of hybrid databases using lightweight metamodel extensions," *Enterprise Information Systems*, doi: 10.1080/17517575.2018.1445295.

Zhang, W.Y., S. Zhang, Y.G. Chen, X.W. Pan. 2012 "An agent-based peer-to-peer architecture for semantic discovery of manufacturing services across virtual enterprises". *Enterprise Information Systems* 9 (3): 233-256, doi: 10.1080/17517575.2012.747002