The RBVO Formation Protocol

Jiří Vokřínek*, Jiří Bíba and Jiří Hodík

Agent Technology Center
Gerstner Laboratory
Department of Cybernetics
Faculty of Electrical Engineering
Czech Technical University in Prague
Technická 2, 166 27 Prague, Czech Republic
E-mail: vokrinek@agents.felk.cvut.cz
E-mail: biba@agents.felk.cvut.cz
E-mail: hodik@agents.felk.cvut.cz
*Corresponding author

Jaromír Vybíhal

Profinit, s.r.o.
Tychonova 2
160 00 Prague, Czech Republic
E-mail: jaromir.vybihal@profinit.eu

Abstract: The proposed protocol has been designed to support the flexible formation of Request-Based Virtual Organisations (RBVOs) with an emphasis on reflecting the conditions of real competitive environments. It supports automated or semi-automated negotiations mainly in the creation part of a Virtual Organisation (VO) life cycle and it accounts for the use of Service Level Agreements (SLAs). The protocol consists of three phases: (1) potential partner search, (2) negotiation of SLAs and the RBVO establishment, and (3) RBVO (execution and) dissolution. The protocol complies with Foundation for Intelligent Physical Agents (FIPA) standards and it has been utilised in the multiagent prototype in a real industrial case. The prototype has been implemented on top of a web-service-based agent platform.

Keywords: negotiation; protocol; virtual organisation; electronic business; e-business; web services; agent platform; service level agreement; SLA.

Reference to this paper should be made as follows: Vokřínek, J., Bíba, J., Hodík, J. and Vybíhal, J. (2009) 'The RBVO Formation Protocol', *Int. J. Agent-Oriented Software Engineering*, Vol. 3, Nos. 2/3, pp.135–162.

Biographical notes: Jiří Vokřínek is a Researcher in the Agent Technology Center at Gerstner Laboratory, Czech Technical University in Prague, Czech Republic. He has a university degree on Replanning in Multi-agent Systems. His research interests are artificial intelligence, multi-agent systems, planning and replanning in manufacturing, virtual organisations, supply-chain management and logistics. He has participated in several research projects focused on agent-based planning and simulation.

Jiří Bíba works as a Researcher in the Agent Technology Center at Gerstner Laboratory of the Czech Technical University in Prague, Czech Republic. He holds a Master's degree in Distributed Artificial Intelligence. His main research interests are the reconfiguration and evolution of commitments in competitive multi-agent environments, negotiation means for flexible contracting using decommitments for the optimisation of commitments, formal representation of commitments and monitoring mechanisms for inspecting adherence of agents to their commitments.

Jiří Hodík works as a Research Fellow in the Agent Technology Center at the Gerstner Laboratory, Czech Technical University in Prague, Czech Republic. He graduated in Technical Cybernetics in 2001. His research interests include artificial intelligence, multi-agent systems, auctions, multicriteria decision making, virtual markets, trust and reputation building and management, and virtual organisations. During a visiting scholarship to the State University of New York at Binghamton, he worked on immune system methods for information security.

Jaromír Vybíhal has a Master's degree in Biomedical Engineering at the Faculty of Electrical Engineering of the Czech Technical University in Prague, Czech Republic. He gained great experience from the Agent Technology Center of the Gerstner Laboratory in the field of competitive negotiation, reconfiguration and contracting in multi-agent systems. He is currently working as IT Consultant in Profinit s.r.o.

1 Introduction

The formation of a Virtual Organisation (VO) is based on a negotiation between independent partners that are willing to cooperate. Individual partners (mostly Small and Medium Enterprises (SMEs)) are motivated to join the VO to increase their business opportunities and to participate in larger-scale jobs.

The VOs naturally operate in a competitive environment. Every partner follows its own goals and maximises its utility. Each of the individual utility functions may use different metrics and they are usually hidden from the others. Standardised protocols for contracting are often insufficient for bargaining over contracts in such an environment, as the related negotiation mechanisms do not account for it (Vokřínek *et al.*, 2007). This paper presents the Request-Based Virtual Organisation (RBVO) Formation Protocol, which aims at the creation of a VO in the competitive domain by two levels of iterative negotiation – the potential partners search and Service Level Agreement (SLA) negotiation. The execution of the VO is not directly covered by the protocol – it controls only the dissolution of the VO.

2 Theoretical background

The modern cooperation concepts go from subcontracting, through the supply chains and then to VOs. Wiendahl and Lutz (2002) has identified three basic types of subcontracting between partners in a network. Although other reasons for subcontracting exist as well (*e.g.*, strategic reasons), these three are also basic for VOs:

1. *Classic subcontracting*: The production of one partner (producer) is an input for the other one (consumer).
2. *Technology-driven subcontracting*: One partner processes a task, but lacks a competency for some of its parts. Therefore, for those parts of the task a suitable supplier is subcontracted.
3. *Capacity-driven subcontracting*: This is similar to the previous one, but the partner responsible for the task lacks a capacity. The missing capacity is outsourced.

This article proposes a protocol for cooperation establishment in the competitive environment. The protocol focuses on virtual consortia formation, which is in principle a technology- and capacity-driven subcontracting.

2.1 The virtual organisation concept

The VO is understood (*e.g.*, Faisst, 1997; Van Wijk *et al.*, 1998; Gruber and Nöster, 2005) to be "… a specific form of network organizations". Gruber and Nöster (as well as Van Wijk *et al.*, 1998, for example) also specifies the key features defined by most of the definitions: the extensive use of information technology to coordinate the partners, sharing risk and knowledge with partners and focus on core competencies. Fischer describes a core competency of an enterprise as a set of skills, technologies and know-how crucial for the added value provided by the enterprise (Fischer *et al.*, 1996). The core competency of a VO consists of the members' core competencies that are crucial for the added value of the VO. The other commonly mentioned features are (presented, for example, by Faisst, 1997 or Capó *et al.*, 2004): autonomy and independence of members, operating towards the customer as a single company and temporality of an existence, which is mission-oriented. Very often mentioned features are also a distribution of members and slight bureaucratic overhead, and one face to the customer.

In the same manner Aubrey (1991) describes the features of VO members. They are *autonomous* (each entity is an independent company or freelancer with its own interests, commitments and goals), *distributed* (entities are naturally distributed in the real world) and *heterogeneous* (each entity may use different technologies and procedures). All these aspects are directly addressed by distributed artificial intelligence and its component of multiagent technologies (Molina *et al.*, 1998) that have already been utilised in the domain of VOs. Fischer (1996) defines the VO and the agents employed in it. Petersen *et al.* (2001) describes the use of agents for the modelling of Virtual Enterprises (VEs, a special case of a VO concept).

Most of the presented features of a VO are commonly accepted, although along with them, there are implementations of a VO methodology that do not fully comply with all the enumerated features. For example, Faisst (1997) draws attention to VOs that may exist without being supported by information technology. Also, the autonomy of members is restricted in some works, where altruistic behaviour is expected from them. In the VO, it must hold that "loyalty is shared among the partners and the cooperation is based on trust and information technology" (Van Wijk *et al.*, 1998). It is necessary for fruitful collaboration. On the other hand, the VO members would participate in such collaboration only if it is also fruitful for them – purely altruistic behaviour is not typical in a real business environment. Nevertheless, VOs that optimise a common profit only also exist. In such cases, there are no private profits of individual members that could be decreased due to increases in the profits of others. Actually, the VO members are optimising their private profits independently based on the fact that profits are shared.

2.2 *Request-based virtual organisations*

The concept of RBVOs defined by Roberts *et al.* (2005) comprises a cluster of partnering organisations that can get along without a hierarchical ordering into a monolithic organisation. The RBVOs are short-lived entities that are formed to respond to business opportunities offered in electronic commerce. RBVO operations are based on predefined SLAs.

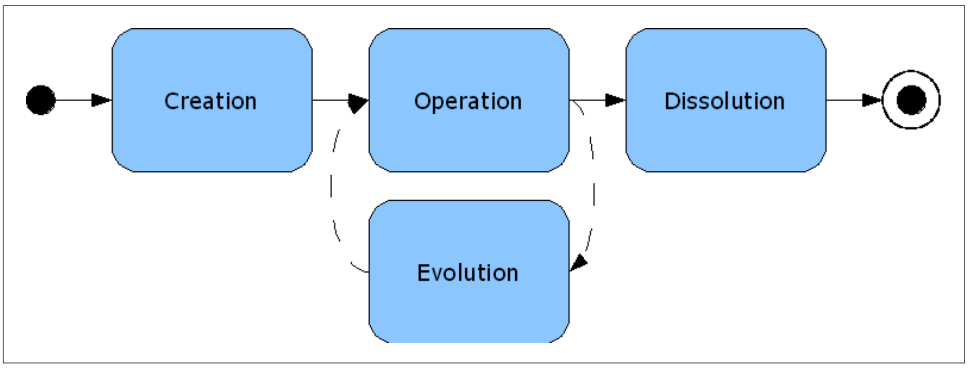
The organisation and functioning of RBVOs' activities are ensured by a community of intelligent agents that automate the procedures and operations of RBVOs (Mařík and McFarlane, 2005). In the RBVO defined by Roberts, the agents serve as assistants for human decision makers; in the agent system each participating SME is represented by its agent, which is able to undertake predefined automated decision-making support on behalf of the SME or enables a user to interact with the system on behalf of the SME. Other possible roles played by agents in VOs are defined, for example, in Hodik *et al.* (2007).

2.3 *The VO life cycle*

The VO life cycle and its phases have been described many times in previous works (*e.g.*, Fischer *et al.*, 1996). The basic phases (see Figure 1), which are included or extended in most definitions, are:

- the creation phase, which is the first phase after discovering a business opportunity. During this phase the VO is created: the VO task is defined, the VO team is formed and then the VO is initiated.
- the operation phase, which contains all the value-adding processes of the VO. In some cases there is a need for an evolution (also called adaptation) of the VO during this phase, *e.g.*, in the case of initiation of new VO members.
- the dissolution phase, which finalises and evaluates the VO operation and potentially opens future cooperation. When the task of the VO is accomplished, the VO operation may be evaluated.

Figure 1 The VO life cycle – the three basic phases and the optional evolution/ modification phase (see online version for colours)



Targeting these main life cycle phases, various authors split them into more phases and define additional subphases as well. The first main phase is creation. Fischer *et al.* (1996) distinguishes two phases of the creation process. In the first phase, the product is defined and the related business process is separated from the product is defined and the related set of business processes is formulated. In the second one, the team of VO members is negotiated and formed. In this second phase (*i.e.*, negotiation and formation, and supporting it with multiagent technology), Fischer distinguishes four subphases:

1 identification of potential partners

2 generation of alternative mappings from partners to individual business processes

3 evaluation of strategic interests and risk

4 finalisation of partners and mapping to partial processes.

A similar concept is presented by Tagg (2001), who has identified three phases of the life cycle for VEs: VE development (establishment phase), business development, and operational. The first and second phases correlate to the two phases defined by Fischer. During these two phases, the VO is created and set up during the establishment and business development phases. These phases consist of negotiations between potential members, checking for potential partners' credibility and authenticity, and contracts (tasks and responsibility allocation). The establishment phase is launched not only at the beginning of the VO; it also covers the expansion and adaptation of a VO team. The operational phase covers doing business and VO performance monitoring and management.

Extending Fischer's work, Faisst replaces the creation phase with three other ones: identification, formation and design (Faisst, 1997). The works of Fischer and Faisst are referred to by Rocha and Oliveira (1999) and Preece (2001), who point out four phases of the VO life cycle: identification of needs, partners selection, operation and dissolution. The creation phase is also extended by Van Wijk *et al.* (1998), who defines seven steps of the VO life cycle: modification of strategy, cooperation strategy, weighing cooperation alternatives, selection of partners, design and integration, management, dissolution and evaluation.

In work related to competency cells, Neubert also mentions the VO creation process (Neubert *et al.*, 2001). When a cell discovers a business opportunity (attracts a customer's production task), the first step that is done in creating a cooperation network is production planning. The cell creates a production plan or subcontracts a specialised cell to do it. The next steps are searching for partners and cooperation formation. The output of the second step (searching for partners) is a set of the potential configurations of a network. From this set the best possible configuration is chosen after negotiation with the agents of potential partners in the third step (cooperation formation). When a suitable configuration of cells is found, they are requested to confirm the obligations.

Most of the authors focus on the creation phase. On the other hand, there are authors like Camarinha-Matos and Afsarmanesh, who concentrate on the operation phase as well. They have defined a life cycle that consists of four phases: creation, operation, modification and dissolution (*e.g.*, in Camarinha-Matos and Afsarmanesh, 1998). The modification phase contains significant adaptations of VOs that cannot be executed during the operation phase. This phase is also called the *evolution phase* (Camarinha-Matos and Afsarmanesh, 2001).

There are projects, such as CONOISE/CONOISE-G, which concentrate on the operation phase as well (Shao *et al.*, 2004). The 'happy path' of their VO life cycle consists of the following phases: formation, operation and dissolution. These phases may be extended by *perturbation*, which is applied in case of some of these events:

- significant deviation is identified and its implications are eliminated by the VO adaptation/evolution
- the VO manager has identified a provider who can provide some service that is included in the plan of the VO in a better way (quality, price, *etc.*) than the provider already included in the VO team, and has negotiated a substitution.

2.4 Cooperation, coordination and commitments in competitive environments

The VO establishment is based on an agreement on the cooperation of individual partners. The concept of *social commitments* was introduced by Wooldridge and Jennings (1999). This concept may be applicable in some VO domains, but it does not address the problem of the unilaterally advantageous dropping of commitments. In most VO domains, an explicit employment of rewards and penalties is needed as a clear qualification of the utilities that the party gains or loses. The concept of such an explicit utility evaluation is then a part of the commitments; the party providing a service commits not only to perform appropriate actions (in order to gain the promised utility which is its motivation), but also to provide compensation in case of failure (*e.g.*, a compensation of the profit lost to the other party). The most complete approach on the commitments in the competitive environment has been presented by Sandholm and Lesser (2001) as Levelled Commitment Contracts (LCCs), which include an explicit utility evaluation in the form of a contract price and penalties. In order to provide a complete decision-making mechanism, the authors applied several significant restrictions (*e.g.*, the utility function needs to be identical for all participants, the opportunity-cost business probability function for every agent is common knowledge, *etc.*). These assumptions are limiting (Bíba and Vokřínek, 2006) and basically prevent the direct deployment of LCC in a real application. Nevertheless, LCC introduces a basis for the notion of commitments in competitive environments.

An SLA introduces a formalisation of a business relationship (or a part of a business relationship) between two parties (most often between a provider and a customer), which is a key concept for service management (Trienekens *et al.*, 2004). Usually it specifies the delivery of products or services for a certain price, meeting specified deadlines and quality requirements together with financial guarantees and other contract terms. It may concern continuous, discrete or one-shot service/goods deliveries. For an RBVO, it represents a description of work flows, schedules, resource allocations, participant roles, prices, sanctions, guarantees, legacy-related and other contract management and coordination issues. The SLA introduces a consistent (possibly reduced) electronic form of the contract signed by the contract parties as a paper document (the reduction may concern mainly nontechnical/financial parts) expressed in a machine-readable language (most often in XML, which is nowadays considered as an interoperable business information exchange format).

2.5 VO/RBVO formation mechanisms

There exist various methods of negotiating and coordinating team actions. Lomuscio *et al.* (2003) define negotiation as "…the process by which a group of agents communicate with one another to try to reach agreement on some matter of common interest". They define two components of the negotiation mechanism: the negotiation strategies and the negotiation protocol. The former defines the lists of actions that individual agents have planned to reach their desired goals. The latter (the protocol) defines rules for messages that are allowed in the message sequence. The rules:

- restrict the allowed types of messages (the 'performative' in the Foundation for Intelligent Physical Agents (FIPA)[1] messages)
- provide constraints for the message content
- define time-outs for receiving the message.

The most popular negotiation and coordination methods are derived from the Contract Net Protocol (CNP) and from the auctions. An introduction of the most important ones follows.

2.5.1 Contract Net Protocol

The Contract Net Protocol (CNP) is one of the most popular negotiation protocols ever used in Multi-Agent Systems (MAS). It comes from economics and is used in communities of altruistic as well as self-interested agents. The CNP was described by Smith (1980), who described a single-shot protocol for requesting and selecting a provider of a product or service in a group of one coordinator (who requests) and one or more participants (who may provide the needed item) (Smith, 1980). In the beginning of the session, the coordinator requests for offers from the participants and the interested ones reply with their offers. The coordinator evaluates the received offers and chooses the most suitable participant(s) or dissolves the session. Finally, if one or more offers are chosen, the coordinator grants the business to participants offering them.

In its basic form, the CNP provides a lot of freedom in each step of the interactions and the obligation to fulfil the contract defined in the call is not required in the basic CNP (*e.g.*, acceptance of proposals depends on the proposals themselves and the actual state of the coordinator at the moment of the proposals evaluation). The CNP fits well in collaborative environments where there is one subject evaluating the possibilities and the others are providing the most suitable offers for the call. In environments in which preference is somehow explicitly expressed (*e.g.*, by money) and/or in environments with competitive participants the CNP must be extended by rules and features, *e.g.*, known from the auctions (Ovcharenko *et al.*, 2006).

2.5.2 Auction

The auction is a method of optimal reallocation of resources according to the actual demand and supply, which are usually measured by monetary units. Many types of auctions exist; they vary in features like bid adaptation possibility, number of sellers and buyers, discrete or continuous evaluation of bids, number of criteria for evaluating a bid and others. The definitions of basic auctions are usually provided for a negotiation about a single issue with invariable features. Basically two auction mechanisms are possible: the one-shot and the iterative.

In the case of the former one, there is only one round of a negotiation. It means that the negotiation coordinator announces a proposal to which the participants respond by obligatory offers. Then the coordinator evaluates the received offers and announces the winning offer(s). There are two basic types of one-shot auctions; they differ in the price that the winner has to pay:

1 in the first-price-sealed-bid auction, the winner pays the price that he/she proposed

2 in the second-price-sealed-bid auction, the price to be paid is defined by the second best proposal.

The second-price-sealed-bid is usually called the Vickrey auction. Although it is an application of the Vickrey auction to the single-item single-unit domain, it is not the only Vickrey auction. The Vickrey auction is naturally a single-item multi-unit. For one kind of commodity (single item), it provides its redistribution of the commodity according to the match of the curves of supply and demand.

The iterative auctions are the English auction, where the price of the auctioned issue is being increased until only one participant is paying for it (for the reverse auction, the price is being decreased until only one interested provider remains), and the Dutch auction, where the price is too high (low in reversion auction) at the beginning of the negotiation and then it is being decreased (increased in the reverse auction) until a participant accepts it. When the English and the reverse English auctions are combined together, the Double auction is created. In that auction there are groups both of participants interested in selling and those interested in buying. The sellers overbid themselves by decreasing the required price, while the buyers increase it. When some selling and buying bids match, the auction is successfully finished. A very special subgroup in the group of iterative auctions are Continuous auctions, in which the bids are evaluated online and when some of them match, the exchange is executed. Independent of an identified match, the auction continues to identify another match of bids.

The iterative auctions are more complicated than the single-shot auctions. In case of a one-criteria iterative auction, where each proposal may by evaluated by a number, *e.g.*, price, the solution is clear: the one with the highest (not dominated) offer is the winner (actually, the evaluation requires a comparability of each two values from the domain of definition to which the evaluation is mapped and the comparability must be transitive). Individual offers depend on the type of auction and the preferences of the participants. In case of a multicriteria description of the proposals such evaluation becomes incredibly difficult and therefore an iterative multicriteria auction is the most complicated one.

2.5.3 Legal Agreement Protocol

The Legal Agreement Protocol (LAP) (Perugini *et al.*, 2007) extends the Provisional Agreement Protocol and is based on Australian contract law. The protocol allows an M:N negotiation which is split into several phases. The first phase allows a not-binding negotiation (the agreed conditions do not imply any commitment for any of the parties) which enables the parties to reach a mutually advantageous compromise. The next phase consists in a binding negotiation over a binding offer (which can be accepted or rejected).

Once a contract is established, it is possible to terminate it in several ways: by fulfilling the contract (this does not require communication), unilateral decommitment under agreed penalties given by the agreement, mutual agreement about cancelling the contracts without penalties and a contract breach (not solved by the protocol – to be resolved per curriam). One instance of the protocol is started for each task (a single-task negotiation) and multiple tasks are negotiated independently in concurrent protocol instances (*i.e.*, multiple contracts). The LAP allows flexible negotiations including backtracking, withdrawing offers, temporary rejections, *etc.* (it is possible to implement various search algorithms such as depth-first search and A*). Decommitments are not negotiated upon, but are carried out unilaterally by informing the other party about the decommitment. The protocol does not directly support contract renegotiation – it is covered by cancellation of the contract or decommitment while new contract conditions are negotiated in a new contract. The protocol assumes safe message delivery and the absence of communication is involved as an interaction option of the protocol (the protocol explicitly considers timeouts). The authors of the LAP have proved various properties of the protocol, *e.g.*, the protocol is free of a communication deadlock (communication is always terminated, though the matter of mutual commitments and their status is not considered) (Perugini *et al.*, 2007).

2.5.4 Competitive Contract Net Protocol

The Competitive Contract Net Protocol (C-CNP) (Vokřínek *et al.*, 2007) is a FIPA-like protocol designed for flexible contracting in a competitive environment (*e.g.*, e-commerce and VOs) and aims at covering the whole contract life cycle, specifically:

- the contract conclusion phase
- the optional decommitment phase
- the contract termination phase.

Not all the parties involved in a multiround negotiation of commitments need to be addressed by Call-for-Proposal (CFP) messages. The protocol allows participants to impose their proposals (based on third-party information) onto an already running negotiation. The 1:N negotiation is held in a pairwise manner. During the execution phase any of the parties involved in pairwise commitments may attempt to decommit from the contract. The multiround decommitment negotiation on conditions of dissolving the cooperation may end up either by the decommitting party backing off (the contract returns to normal) or dropping the commitments under the payment of a penalty (the penalty may be fixed during the contract-conclusion negotiation or may remain open and be adjusted in time). Finally, in the termination phase, the results are evaluated with respect to the agreed commitments. Eventually, penalties for noncompliance with commitments are negotiated. The message content is assumed to describe the contract as a whole, *i.e.*, full and explicit descriptions of commitments (*i.e.*, not merely the task assignment, but also resource allocation, quality of service, schedules, *etc.*), rewards and sanctions are provided (such message content may be, for example, an SLA). Thus, the negotiation is also assumed to be multiattribute rather than single-attribute. The multiround manner of the protocol allows multiple, simultaneously running negotiations as well as multilevel ordering of subsequent protocols (*i.e.*, a participant of a C-CNP may become a coordinator of another subsequent C-CNP negotiation, *e.g.*, for outsourcing).

2.5.5 Renegotiable Competitive Contract Net Protocol

The Renegotiable Competitive Contract Net Protocol (RC-CNP) (Bíba *et al.*, 2008) extends the C-CNP through renegotiation phases, provides means for a fully flexible contracting in competitive environments and enables a consistent evolution of commitments starting at their creation and terminating with their fulfilment, adaptation or breach (even a partial breach) under punishment (payment of a penalty) – *i.e.*, it covers a complete commitment life cycle within a group of mutually committed agents (the commitments are pairwise between coordinator and participants). The protocol allows M:N multiattribute negotiations and it can be extended by not-binding phases. Moreover, it is capable of dealing with a possible temporary communication inaccessibility by utilising communication timeouts and related default transitions (as well as a possible synchronisation backtracking).

3 The RBVO Formation Protocol

The protocols and approaches presented in the previous section give solid bases for any negotiation-based cooperation establishment. In this work, we focus on the RBVO formation. It is based on a negotiation between independent actors that are willing to cooperate. Individual actors are motivated to join the VO to increase their business opportunities and to be able to participate in larger-scale contracts. An RBVO is a mutually agreed consortium of such individuals that is formed to cover a complex business opportunity. This business opportunity is introduced by the *coordinator*, which leads the negotiation with several potential *participants* – partners selected to cover some part of the business opportunity.

The protocols described before are not sufficient or are too complex for practical usage in the domain of RBVO formation. Thus we have designed the RBVO Formation Protocol, inspired by CNP and C-CNP, which allows the implementation of arbitrary business strategies or auctions. It is targeted at the negotiation protocol component of the negotiation mechanism (Lomuscio *et al.*, 2003). It does not address the planning step introduced in Neubert *et al.* (2001), but focuses on the searching for partners and cooperation formation steps. This work also builds on the results of other works (*e.g.*, Rocha and Oliveira, 1999; Preece, 2001; Van Wijk *et al.*, 1998) that focus on establishing a cooperation using the concept of VO as well.

In the RBVO domain, we understand the business opportunity introduced by the coordinator as the prepared production plan and RBVO formation as Neubert's configuration of the network.

3.1 Protocol description

The RBVO Formation Protocol supports a contract negotiation on several levels. It consists of three phases:

1 the search for potential partners (prenegotiation of contract)

2 negotiation of SLAs with selected partners and establishment of the RBVO (one partner or a small number of partners that together cover the required competencies)

3 (execution and) dissolution of the RBVO.

The first two phases enable multiround negotiations and concern the creation phase of an RBVO life cycle. Both the coordinator and the individual potential partners are allowed to withdraw from the negotiation for any reason. During the first two phases, the final RBVO configuration is agreed together with the related commitments (given by pairwise or multiparty SLAs) for all the parties. The final phase concerns the simplified execution and termination phases of the RBVO life cycle.

A sequence diagram of the RBVO Formation Protocol is shown in Figure 6 and the hybrid state diagram is in Figure 7. The individual phases (see Figure 2) are described in detail as follows:

Phase 1 the search for potential partners

The first phase aims at a prenegotiation of the contract conditions with possible partners (equipped with the required competencies) with respect to the ratings of their offers so that the number of partners selected for detailed negotiations of SLAs is reduced and the best candidates are chosen. The negotiation is started by sending a *Collaboration Request* (CR) message as a CFP. The CR describes the decomposed tasks, respective competencies required for their accomplishment and constraints (*e.g.*, geographical location of a potential candidate, due dates). The coordinator and the candidates enter into a pairwise multiround bargaining (in a *propose/counter-propose* manner) in which they try to agree on preliminary cooperation rules. Thus, the coordinator obtains several possible configurations of the resulting RBVO. The coordinator decides on the best configuration and sends the respective candidates *preliminary-accept* messages containing SLA proposals. The other candidates are not rejected immediately, but should remain in the first phase of the bargaining process while the preselected candidates enter the second phase. The waiting candidates may get their chance if the coordinator fails to reach agreements on SLAs with some of the preselected candidates. The granularity of information in CRs is generally less fine than in SLAs (some of the attributes may be irrelevant to negotiate upon until the pre-agreement is reached). Both the coordinator and the candidates are allowed to terminate the negotiation for any reason by sending *refuse* participation (candidates) or *reject* participation (coordinator) messages.

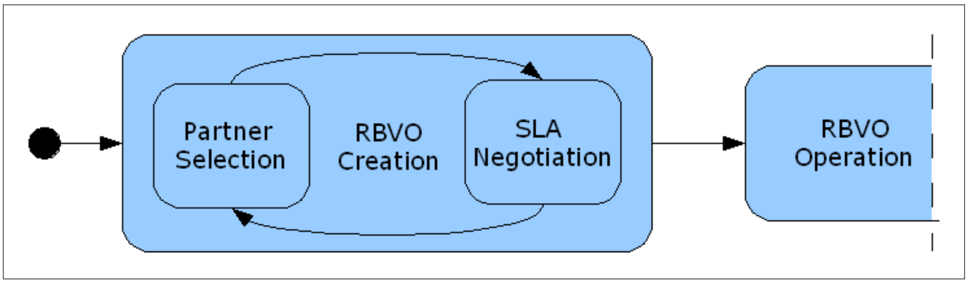
Phase 2 negotiation of SLAs and establishment of the RBVO

The coordinator and the candidates preselected for the negotiation of the detailed terms and conditions of the resulting SLAs try to reach a final agreement reflecting various aspects like time schedules, qualitative parameters, prices and penalties. The pairwise multiround bargaining (again in a *propose SLA/counter-propose SLA* manner) may be terminated by any of the parties provided a mutually advantageous compromise on the contract conditions appears unreachable. In such a case, some of the waiting candidates, equipped with respective competencies can be invited to the final negotiation by a preliminary accept message from the coordinator. As soon as all SLAs are finalised, *confirm-SLA* messages are sent to the candidates and the RBVO is practically established. The waiting candidates that are not included in the RBVO receive *reject-participation* messages. The RBVO is created and the execution and termination phases of the RBVO life cycle can take place.

Phase 3 (execution and) dissolution of the RBVO

As soon as a participant accomplishes its tasks, it sends an *SLA-done* message to the coordinator. The coordinator terminates the cooperation by confirming the dissolution of the RBVO.

Figure 2 The RBVO life cycle supported by the RBVO Formation Protocol (see online version for colours)



The RBVO Formation Protocol, as designed, does not provide direct support for the evolution (also named modification or perturbation) phase of the VO life cycle. The evolution may be invoked during the operation (execution) phase in case of incidents that endanger accomplishment of the RBVO mission or in case of an opportunity to increase the efficiency of the RBVO.

On the other hand, such a situation may be understood as a very special case of RBVO formation with additional constraints. The constraints are twofold:

1 Existing tasks which have been already started or finished can no longer be modified.

2 Existing tasks which have not been started can be modified.

The assignment of the latter ones may be modified. The modification process consists of:

- the substitution of the tasks to be modified by new decompositions considering the already fulfilled/finished tasks
- the invocation of both the prenegotiation and SLA negotiation phases of the protocol to find partners to take on the alternative obligations defined by the new decomposition subset.

The alternative obligations are applied to the involved participants as well as to the coordinator. It means that the coordinator may be obliged to pay a penalty in case of the cancellation of the contract, although the partner has not started working on it yet – the partner may have already booked resources for it and even if the partner will not use them, the related costs have to be paid.

This modification process influences the assignment only of the tasks included in the set to be modified. The other tasks are not influenced but they may generate constraints (especially for the available starting date and due dates) that the coordinator must take into account during the evaluation of offers received from potential participants. Actually, the list of tasks considered for the assignment modification may vary many times during this phase because it is affected by the ongoing negotiation.

3.2 Timeouts and accessibility synchronisation issues

According to Lomuscio *et al.* (2003) the timeouts for the messages should be defined. Not all messages have to be secured by timeouts. The critical messages are the collaboration request, propose, counter-propose, pre-accept, SLA propose and SLA counter-propose messages.

Due to the nature of the distributed environment, there are potential problems with timeouts and synchronisation of the protocol for each participant. An inconsistency of the protocol state can be caused by:

- a *message loss*, when a party sends the message and the other party does not receive it
- a *protocol breach* caused either intentionally by a party (*e.g.*, when it is unwilling to communicate with a certain party or an unexpected message is received) or by a software bug
- *communication inaccessibility*, when a party is not able to send messages to others
- a *timeout*, when some message is sent or received after the deadline (*e.g.*, the participation proposal).

All the above occasions lead to an inconsistency in the protocol state of the coordinator and one or more participants. The protocol has been designed to operate in dynamic environments, where it is often not possible to ensure that the participants will follow the protocol without intended or non-intended breaches (*e.g.*, because of timeouts or lost messages). Mainly, the initial CR may remain unanswered by the addressed participants for any of the presented reasons. In this case, the coordinator continues to execute the protocol and the participants have to synchronise their state. To keep the business logic consistent, the following suggestions are made:

- Missing received communication is understood as *refuse, reject* or *refuse SLA sent* (depending on the current protocol state).
- If the coordinator receives a message that breaches the protocol, it ignores it.
- If a participant receives an unexpected message, it has to synchronise its state and continue according to the protocol.

3.3 Protocol extension by exceptional messaging

In some cases, there is a need to adapt the protocol to enable more freedom for the business strategies. For example, the timeout of some proposals can be shorter than the timeout for the CR. In this case, the coordinator has to move the protocol state with a particular partner and continue with a *pre-accept* or *counter-proposal* message before another partner sends the initial proposal. In fact, this is not a protocol breach or inconsistency. Individual coordinator-to-participant relations are not affected, but due to the selected business strategy, the coordinator has to decide the proposal evaluation before all the proposals are received.

Let us discuss this issue with the following example. The coordinator is looking for a rent-a-car provider. Its business strategy is to look for the lowest available price with a desired daily rate under 200 euros. There are three potential participants: *A*, *B* and *C*; so the coordinator starts the negotiation by sending a CR to all of them with the timeout of one day. Participant *A* answers immediately with a proposal for 250 euros per day with a one-day validity. Participant *B* answers in one hour with a proposal for 180 euros, but this proposal is valid for one hour only. Participant *C* sends no proposal within the next hour. At this moment, the coordinator has to decide either to wait for the last proposal and potentially lose the best offer or to finish the negotiation by accepting the proposal of Participant *B*. The coordinator chooses the second option and sends a *pre-accept* message to Participant *B* and continues to the SLA agreement. When the SLA is confirmed (between the coordinator and Participant *B*), the coordinator sends *reject* to Participant *A*. According to the protocol, it is not possible to send *reject* to Participant *C*, but if Participant *C* sends a proposal in the future it should certainly be rejected as well.

This example illustrates the need for exceptional protocol handling due to complex business strategies. We introduce *in-advance messages* for handling this. Such a message can be understood as a default message that will be used as a response to another party's activity. This type of functionality is mainly needed at the coordinator side, so the usual in-advance messages are *counter-propose*, *pre-accept* and *reject*. In-advance messages are handled by the protocol controller to ensure consistency and correct termination of the protocol.

Another type of exceptional message can be provoked by changes during negotiation. In general, the negotiation is performed in a dynamic environment, where individual agreements affect each other. An example would be as follows: The coordinator receives a proposal from another participant not involved in the negotiation yet. It sends a *counter-propose* message, but meanwhile (as more proposals are received) the first proposal becomes satisfactory. At this moment, the coordinator is willing to accept the proposal, but this leads to a protocol breach. To help with this issue, we introduce *instant messages* that can be sent even if it is not the particular actor's turn. The possible instant messages for the coordinator are *counter-proposal*, *pre-accept* and *reject*. For the participant, the possible instant message is *proposal*.

Both types of messages are summarised in Table 1. In-advance and instant messages address the need for exceptional messaging. The protocol is not breached by using those messages, but the synchronisation issue arises (see Section 3.2).

Table 1 Exceptional messages of the RBVO Formation Protocol

Action	*Message type*	*Actor*
Counter-propose	In-advance	Coordinator
Pre-accept	In-advance	Coordinator
Reject	In-advance	Coordinator
Counter-propose	Instant	Coordinator
Pre-accept	Instant	Coordinator
Reject	Instant	Coordinator
Propose	Instant	Participant

4 Auctions implementation using the RBVO Formation Protocol

The process of RBVO creation consists of the following steps:

Step 1 the search for partners

Step 2 obligations finalisation during RBVO establishment.

The auctions usually concentrate on the latter phase and they do not solve the former one; the potential partners are identified before or during the negotiation initiation. Another difference between the RBVO Formation Protocol and classical auctions is that the auctions mostly define convergent criteria for the proposals and counterproposals offered during the negotiation; the RBVO Formation Protocol does not limit the composition and evaluation process of proposals and counterproposals. Therefore the RBVO Formation Protocol may implement almost any auction mechanism type for the negotiation. This is done by applying the rules defining the chosen auction type to the RBVO Formation Protocol.

In the RBVO Formation Protocol, the auction mechanisms may be applied during the RBVO establishment phase. Many auction mechanisms exist and their suitability for negotiation is domain dependent. Here we provide examples of how the auction mechanisms may be included in the RBVO Formation Protocol:

- The one-shot auctions (*e.g.*, first-price-sealed-bid or second-price-sealed-bid auctions) are implemented by one round of SLA negotiations, when the SLA proposal is provided by the coordinator to the participants, which consequently respond with counterproposals. Such counterproposals are evaluated according to the used auction rules and the winner(s) is (are) announced by the final SLA proposal(s) that they agree and that are finally confirmed by the coordinator.
- The English iterative auction may be implemented through continuous SLA proposing being conducted by the coordinator. During the auction process, the coordinator adapts the offer in order to slightly improve its own potential profit, as it would be if the proposed SLA is confirmed by all parties. The negotiation continues until the group of interested participants is decreased to the minimal group included in the formed RBVO. The interested participants' group is expected to be reduced because increasing the potential profit of the coordinator negatively influences the potential profit of the participants in the RBVO. Therefore they leave the negotiation because the conditions under negotiation may become disadvantageous for them. The last participant(s) standing is (are) the winner(s).
- The Dutch iterative auction may be implemented similarly to the English auction, but the first proposal given by the coordinator is not profitable for any participant to accept it. Then the proposed conditions are continuously adapted in order to make them more interesting for the potential participants until the group of participants covering all the requirements for the RBVO establishment is identified by their acceptance of the proposals given by the coordinator.

5 The PANDA case study

The presented RBVO Formation Protocol has been utilised by the multiagent system prototype within the PANDA[2] project, which aims at collaborative process automation in the ERP/CRM[3] industry. It facilitates the creation of international e-collaborations based on RBVO formation using sector-specific SLAs and a community of intelligent agents. The individual value chain actors act as service providers mainly for consulting, software implementation, installation and customisation, training and maintenance. The motivation is to find the most suitable consortium of service providers to meet customer requirements such as cost, experience in the industrial domain and appropriate ERP solution, geographical location and language.

The system is composed of a set of distributed partner agents and a central or distributed platform that supports services. The central platform services provide public data and they are considered to be fully accessible for the agents (that are online). The agents operate in a semi-accessible mode – it means the agents can be inaccessible for some time (*e.g.*, any agent can be offline or turned off).

The PANDA system prototype is composed of the following components:

- portal – a unified user interface that incorporates all the user interactions with the whole system
- central services – data and directory services providing competency taxonomies, profiles, *etc.*
- agent platform-supporting services – web-services-based FIPA-compliant platform implementation provides agent management, online and offline agent communication channel and directory services
- partner agents – distributed components representing the companies operating in the system. Every company deploys one agent that performs negotiations on behalf of its owner.

The PANDA intelligent agents play the role of the partners' representatives for (semi-)automated negotiation that supports the e-business acceleration in the ERP value chain domain. Although inspired by real business, the outcome of the PANDA intelligent agents system should be generalised for the wider domain and be adaptable for any value chain and different domains.

The generic constrains of the PANDA case should be summarised as follows:

- The system should support independent, self-interested, geographically distributed players registered within the chain.
- Every member can introduce new business opportunities and start the RBVO formation.
- The RBVO is formed upon peer-to-peer negotiations.
- Every member is able to use private preferences and constraints during negotiations.
- The coordinator (the one who starts the RBVO formation) is able to evaluate the cooperation proposals using private preferences and rules. It is also able to choose the type of auction and evaluation method.
- The actors' responses are based on dynamic private knowledge.

The main goal of the PANDA multiagent system is to provide negotiation-based matchmaking methods, taking into account limited information provision, multicriteria evaluation of proposals and private preferences and metrics for each participant. Using public data, previous experience or reputation mechanisms, the coordinator starts the negotiation (using the defined communication protocol – the RBVO Formation Protocol) with a preselected set of potential participants, which provides the highest probability of matching the coordinator preferences after the negotiation phase. The peer-to-peer negotiation is used for gathering semiprivate knowledge to be able to select the appropriate partners (proposals are based on participants' private preferences and availability and cannot be evaluated without negotiation) and construct the RBVO. The finalisation of the matchmaking is done by the RBVO proposals evaluation and there is a possible backtracking when the evaluation gives nonsatisfactory results.

A typical CR contains global constraints (*e.g.*, validity of the CR, owner, deadline, overall indicative budget) and a list of tasks. Each of the tasks contains more detailed constraints (*e.g.*, the service required, starting date, end date, language, location, resource amount required) and provider constraints (*e.g.*, required expertise domains, ERP modules expertise, reputation). By way of illustration, the CR for RBVO can by composed of the following tasks:

Task 1 implementation in Germany, required expertise in the food industry and sales management ERP module

Task 2 consulting in Germany in English, required expertise in sales management ERP module

Task 3 support in Germany in English, required expertise in the food industry.

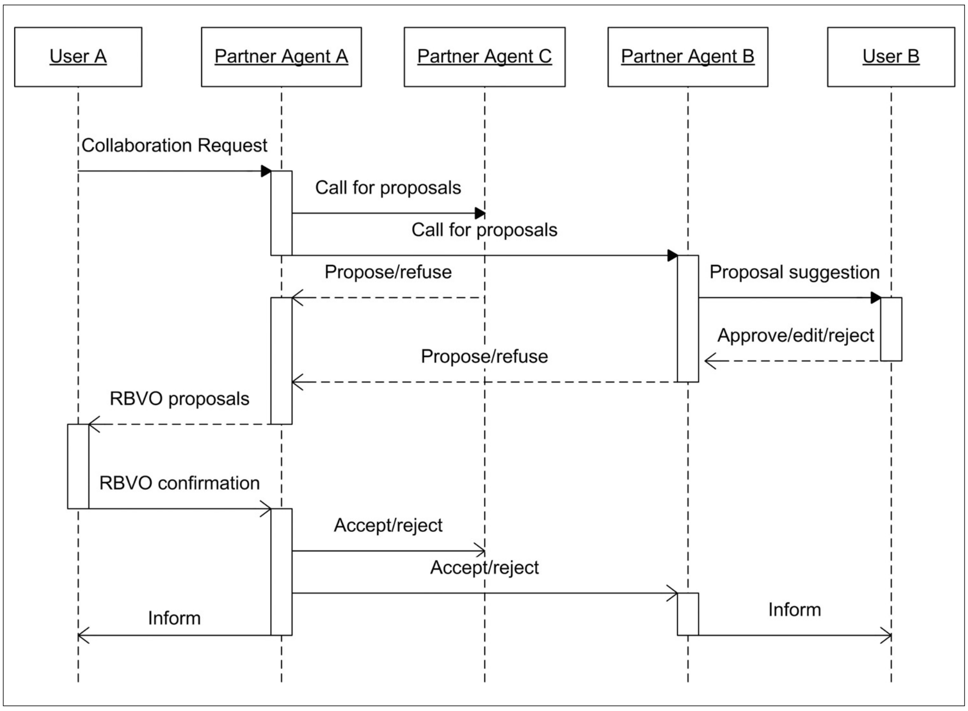
The coordinator is looking for a consortium of participants – an RBVO covering all three tasks in the best condition. The evaluation metrics of the RBVO is private for the coordinator.

5.1 Deployment of the RBVO Formation Protocol in PANDA

The negotiation-based matchmaking is based on the interaction of the coordinator and the potential participants, where all actors follow their own private strategies. The interaction is based on the presented RBVO Formation Protocol. The coordinator introduces a business opportunity and starts to negotiate the potential RBVOs. It facilitates the multiattribute/multicriteria evaluation of the potential RBVOs combined from the received proposals. The goal is to find a pareto-optimal set of RBVO clusters with respect to user-defined constraints and preferences (represented by rules, weights, *etc.*). Such RBVO clusters are dynamically created from a combination of proposals and appropriate counterproposals are generated to converge to the desired solution.

The example of the RBVO Formation Protocol utilisation is given in Figure 3. There are three Partner Agents (*A*, *B* and *C*) and two involved Users (*A* and *B*). The figure illustrates the first stage of the RBVO Formation Protocol with no counterproposals. Company *A* (represented by Partner Agent *A* and User *A*) is the coordinator and Companies *B* and *C* are the participants. While Company *C* is represented by a fully automatic agent with no user interaction, Company *B* is represented by a semi-automated agent in the role of an assistant. Also, the final selection of the best RBVO is approved by the human user.

Figure 3 A simplified example of RBVO Formation Protocol utilisation in PANDA



Let us discuss a realistic scenario based on the CR defined above. Let Participant *B* propose to provide all three services for an overall price of 1000 euros. Participant *C* proposes the provision of Task 1 for the price of 500 euros. There are two potential RBVOs constructed from this proposal:

1 all three tasks will be provided by Participant *B* for 1000 euros

2 Task 1 will be provided by Participant *C* for 500 euros and Tasks 2 and 3 will be provided by Participant *B* for less than 1000 euros.

For the second RBVO, the coordinator does not have enough information to evaluate the price correctly, so the counterproposal should be made to Partner *B* for the provision of Tasks 2 and 3. In the worst case, all the combinations have to be examined. The RBVO Formation Protocol allows such bargaining to reach the most suitable solution, but the logic of the bidding is within the scope of the implemented business strategy. When the final RBVO is selected, the coordinator sends the respective pre-accept messages and continues with the SLA negotiation phase. In the PANDA demonstration prototype, the SLA negotiation is done mainly offline or in the form of text document attachments. The second and third phases of the protocol usually follow the simplest, straightforward, message sequence: *SLA propose*, *agree SLA*, *confirm SLA*, *SLA done*.

5.2 Rule-based matchmaking strategies using the RBVO Formation Protocol

As described before, the RBVO Formation Protocol can be used for the utilisation of any type of auction mechanism. In the PANDA prototype, the business logic is provided by the agent Knowledge Processing Module. It utilises the Drools engine – a business rule management system provided by JBoss.[4] During the negotiation driven by the RBVO Formation Protocol, the Knowledge Processing Module executes a set of business rules to create a proposal as a participant, and evaluate the obtained proposals and generate counterproposals as a coordinator. This approach enables the abstraction of the business logic layer from the protocol and implementation of any kind of business strategy on top of the RBVO Formation Protocol. Table 2 shows an example of the business rules implemented in the prototype. The overall optimisation strategy is to get the cheapest possible RBVO while all the defined constraints are satisfied.

Table 2 Example of business rules implemented by the PANDA partner agent

Business rules
I need to finish the negotiations in x days
Only consider partners that offer services in the same country as the requested services (Y/N)
Only consider services offered by local partners (Y/N)
Reject listed sellers – blacklist (Y/N)
Don’t show me consortia > x partners
Offer discount x% for specified buyer
I will always start my bidding at my listing price, irrespective of the project budget (Y/N)
I do not want to participate in projects less than x euros
I do not want to participate in tasks less than x Mandays
I do not want to participate in projects (*i.e.*, set of tasks offered to me) less than x Mandays
I do not want to participate in projects more than x euros
Reject listed buyers – blacklist (Y/N)
Give a discount x% in my reply that include multiple tasks

The whole negotiation matchmaking logic in PANDA is supported by three components:

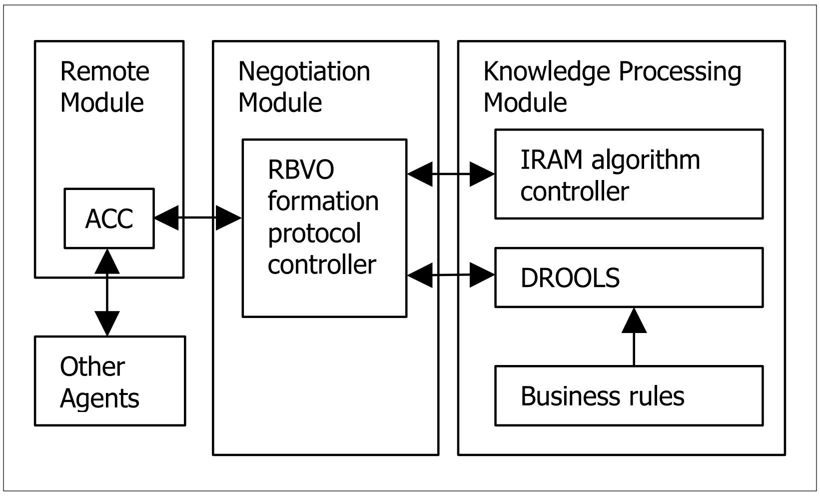
1 the RBVO Formation Protocol as a negotiation frame

2 business rules as a proposal generation and RBVO evaluation tool

3 the Incrementally Refined Acquaintance Model (IRAM) algorithm as an optimisation strategy.

The iterations of negotiation towards RBVO formation are executed by the IRAM algorithm (Pěchouček *et al.*, 2008), which provides fast convergence to the optimum while minimising the loss of private information. It enables one to minimise the number of counterproposals and (thus the information obtained) needed to reach the best RBVO. The algorithm is also able to provide the best known solution in every iteration step. The schema of the agent processing logic is in Figure 4. The agent communicates via the *Remote Module* and Agent Communication Channel (ACC) with other agents according to the RBVO Formation Protocol controlled by the *Negotiation Module*. The *Knowledge Processing Module* behaviour depends on the agent's role in the particular negotiation. The business rules can differ for the coordinator and participant and the IRAM algorithm is utilised by the coordinator only.

Figure 4 The simplified structure of the PANDA prototype partner agent



5.3 Agent system implementation

There are several technologies for integrating web services and agent platforms. One of the approaches is to enable the transparent cooperation of agents operating on an existing agent platform and web services (Esteban, 2007; Overeinder *et al.*, 2008). Such an integration is usually gateway based[5] – the service invocation or agent messaging is transparently translated and passed to the other side. So the agents can be invoked as web services (only simple protocols like request-respond are supported) and agents can communicate with web services in the agent manner. This approach is suitable for the integration of the existing agent system and web services and takes advantages (and also disadvantages) from both – the agent platform and web services technology.

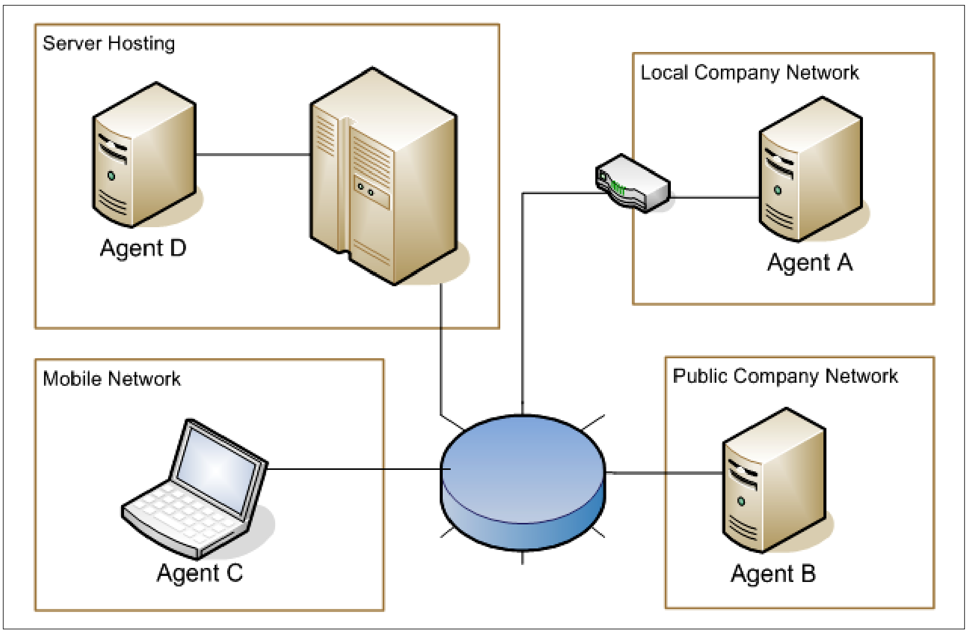
Since the PANDA agent system has been designed to work in an open internet environment from the beginning, the focus has been placed on the maximisation of the system's stability and robustness. To keep all the main features like FIPA compliance, openness, stability, usage of up-to-date standards and technologies, we have designed and implemented the web-service-based agent platform.

The agent platform complies with the FIPA standards, while all the platform services are implemented as web services (see Figure 8). The agent programming interface provides access to all the necessary interactions with the platform (Directory Services, Agent Management System and ACC). On top of the agent platform, the RBVO formation protocol controller has been implemented. Through this layer, the agents no longer need to implement standard message handling (in the manner of sendMessage and handleMessage methods), but the high-level methods are available for controlling the protocol.

For example, when a new negotiation for RBVO formation has to be started, the agent invokes (a standard JAVA method call) the method of the protocol controller with the CR as a parameter. According to the CR, the protocol controller invokes the directory service of the agent platform (the JAVA method call redirected to the WS invocation) and receives the addresses of potential collaborators. Then the protocol controller prepares the CFP and calls the sendMessage method of the platform (the JAVA method call redirected to the WS invocation – the direct invocation of the messaging service of online agents and the invocation of offline platform message service when needed). The messaging web service of the receivers puts the CFP message into the incoming message buffer. Then the protocol controller obtains the CFP, creates a new instance of the RBVO Formation Protocol for this call, and the preparation of a proposal is started. In the case of offline messaging, the agent checks the offline message box (WS invocation) upon starting, and then periodically, to handle temporal connection problems or a network configuration with a limited peer-to-peer communication.

The described implementation of the agent platform allows the development of lightweight independent agents with the ability to operate while being widely distributed across the internet in various network settings (see Figure 5). The web service nature of the platform guarantees agents' full operation as long as the web service invocation of the agent platform services is possible (the agents operate in the passive mode). When the invocation of the web services deployed on each agent (mainly distributed ACC) is enabled, the agents are able to communicate directly in full peer-to-peer manner (the agents operate in the active mode). The agent platform also provides several optional tools for message sniffing, logging and interactions inspection and visualisation.

Figure 5 Example of the deployment of the PANDA agents (see online version for colours)



Note: The agents can be hosted on the server and run on the public network, on the private network (behind address translation), or on the mobile devices.

Figure 6 RBVO Formation Protocol sequence diagram

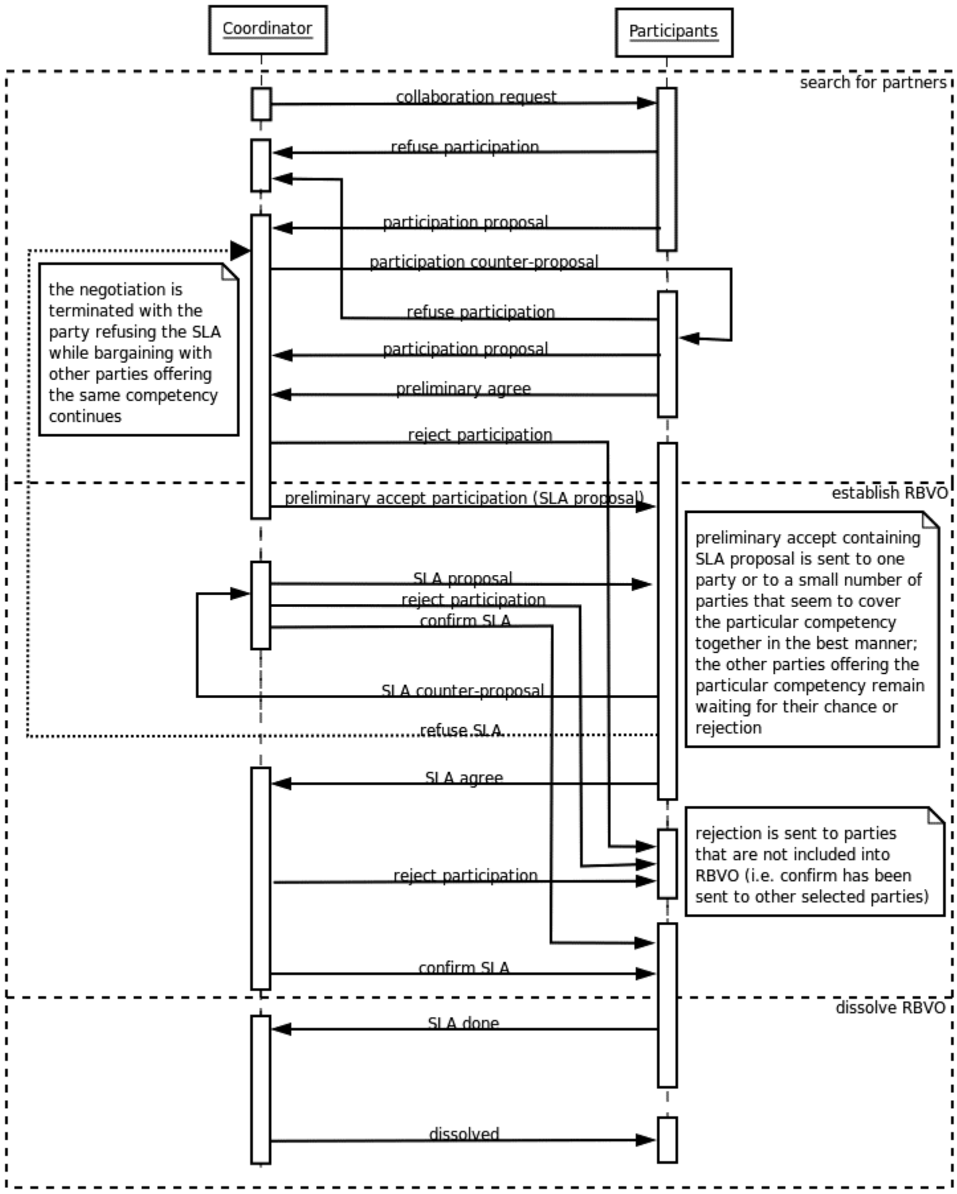


Figure 7 RBVO Formation Protocol hybrid state diagram (see online version for colours)

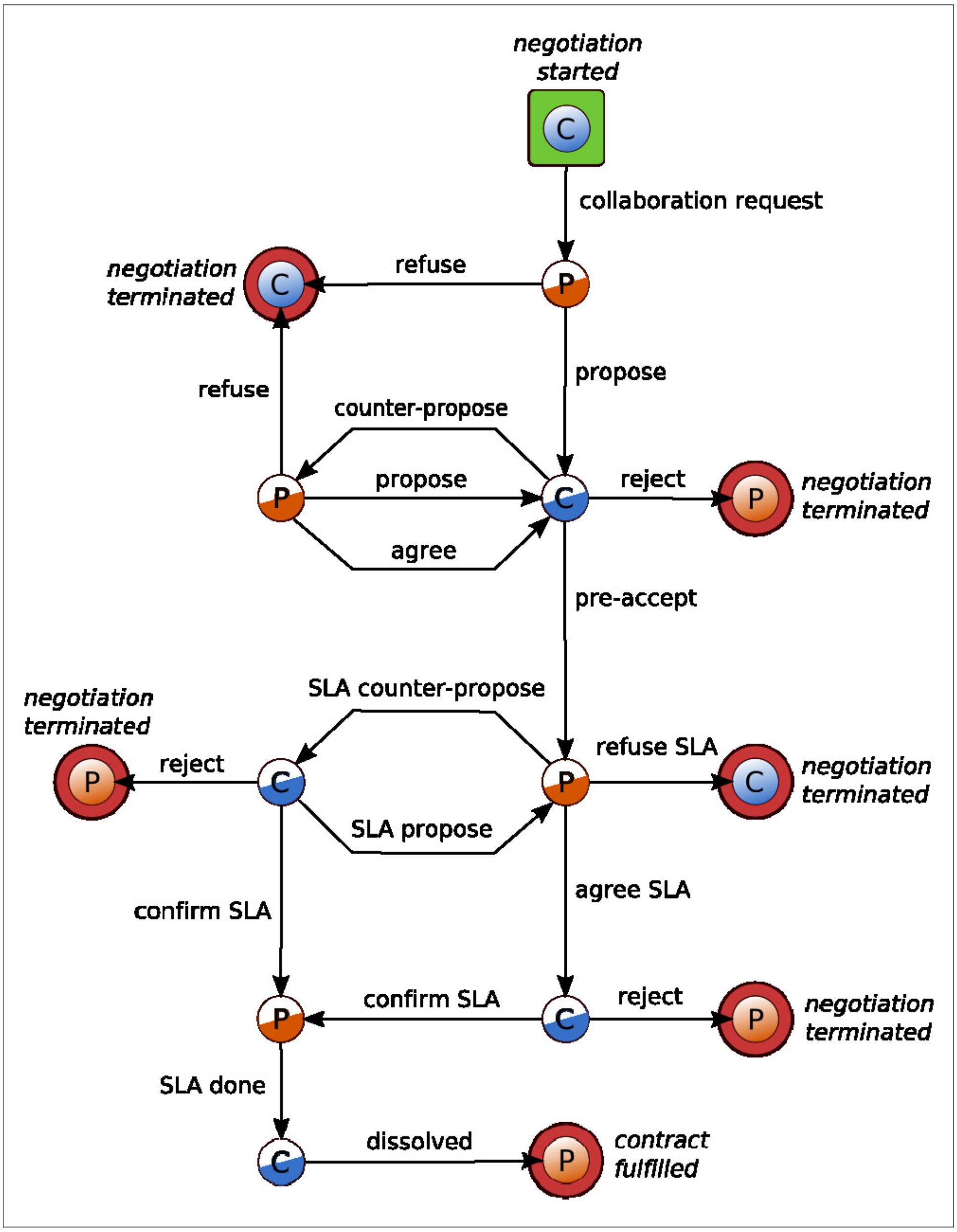
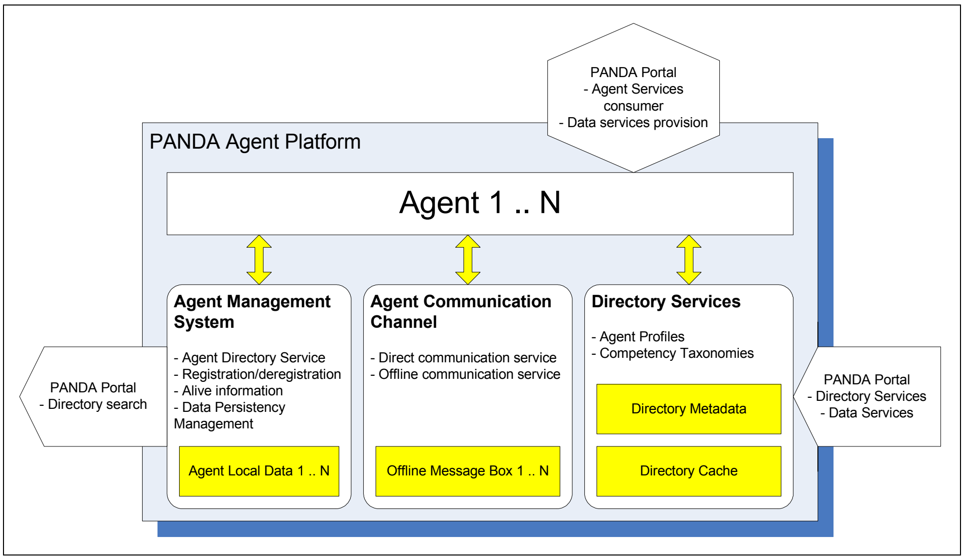


Figure 8 The structure of the PANDA Agent Platform (see online version for colours)



Note: There are also horizontal interactions with the PANDA Portal (transparent to agents) and vertical interactions for the integration of individual agents with other systems of the agent's owner.

The PANDA Partner Agent application is able to act in the community of agents as a coordinator and/or participant at the same time (this means it is able to start RBVO formation or participate in an RBVO formation initiated by another agent). It is empowered by the implementation of the RBVO Formation Protocol, rule engine and set of private business rules adjusted by the owner (human user). For the user interactions, it is equipped with the JAVA graphical user interface. In the case of server hosting, when direct application access is not possible, the agent can be controlled by the remote Graphic User Interface (GUI) using the *PANDA Agent Service.* This service is optionally deployed on each agent and enables one to access agent capabilities remotely. Although the service does not provide all the features like the application GUI (such as the CR editor, business rules editor, various RBVO filtering and sorting, *etc.*), it allows one to create any kind of custom user interface or integration of the agent in the internal business processes and applications of the company. This web service provides the following methods:

- submitCr – enables the submission of a collaboration request to the agent. The method also allows the definition of the set of business parameters for this CR
- getRbvoProposals – provides the list of proposed RBVOs for a particular CR
- getCrStatus – provides the status of a collaboration request
- confirmRbvo – enables the selection of an RVBO and signals the agent to close the collaboration request
- getParticipBizRuleParams – enables acquisition of the list of business rules parameters actually used on the participant's (collaborator's) side

- setParticipBizRuleParams – enables updating of the list of business rules parameters actually used on the participant's (collaborator's) side
- getObtainedNegotiations – enables acquisition of the list of negotiations that the agent participates in
- submitObtainedNegotiation – enables updating of a negotiation (that is in pending or empty status).

To have a complete picture, there is a possibility to integrate the agent with the private company data stores. There is a web service interface designed to enable the *Knowledge Processing Module* of the agent to ask an external service for the creation of proposals for incoming CRs. The idea is to substitute the Drools engine used by agents internally with any proposal creation mechanism. Using this interface, the agent is enabled by the possibility to create collaboration proposals based on real data and standard intracompany processes.

The implementation of the PANDA agent system has been performed in the JAVA programming language. The platform services have been deployed on the TOMCAT server beside the Portal and Central Services. The Partner Agents use native JAVA JAX-WS[6] implementation as the service container. Each agent runs as a separate JAVA application and is able to be deployed 'any-place', where JAVA 1.6SE or a better runtime environment is installed. It needs at least an outgoing internet connection (for web services calls) and optional incoming connection. The distribution (and also updating) of the agents is provided using Java Web Start Technology.[7]

6 Conclusion

The proposed protocol has been designed for RBVO formation, but it is also possible to deploy it to other domains of VOs which employ the concept of SLAs. The protocol allows for reflecting the conditions of real competitive environments as well as negotiation scalability and complexity and is a support for human-assisted negotiation.

The first phase of the protocol focuses on the multiround prenegotiation of the contract conditions between the partners. This phase is finished by a preliminary agreement or a participation refusing/rejection and can be fully or partially automated (agents negotiating on behalf of their owners). The second part contains pairwise multiround bargaining of the agreements. The result of this part is a set of SLAs or participation rejections. The third part is focused on RBVO dissolution and does not offer any special terminating conditions (*e.g.*, penalties, quality of service delivered) or RBVO execution/evolution control.

A possible improvement of the presented RBVO Formation Protocol consists in the adoption of features of the C-CNP Protocol in its decommitment and termination phases. In fact, the proposed RBVO Formation Protocol can be used instead of the contracting phase of the C-CNP. The decommitment phase of C-CNP or (the more complex) renegotiation phase of the RC-CNP should directly address the needs of the evolution (modification/perturbation) phase of the VO life cycle.

The RBVO Formation Protocol has been successfully deployed in the PANDA prototype and tested in the real demonstration of the system. There are about 38 partner agents utilising the protocol deployed on various sites (hosted on the central platform server or distributed on the user partners' servers or PCs in seven different geographical locations); 12 of them are able to play the role of coordinator and thus initiate RBVO formation. The partner agents are able to provide five different services with various constraints like languages, countries, industry domains, ERP module expertise, reputation, price and availability. In the worst case, there is a maximum of 24.3 million potential RBVOs for nonconstrained five-task CR if all of the companies offer all of the services. The RBVO Formation Protocol empowered by the IRAM algorithm and business logic captured by rules provides efficient (semi-)automated cooperation establishment and effective collaboration support.

Acknowledgements

The work is partially funded by the European Commission's FP6 programme within the projects PANDA (contract no. 027169) and ECOLEAD (contract no. FP6 IP 506958), and the Network of Excellence for Innovative Production Machines and Systems (No. 500273-2). It is also supported by the Czech Ministry of Education grant 6840770038.

Any opinions expressed in this paper do not necessarily reflect the views of the European Community. The community is not liable for any use that may be made of the information contained herein.

References

Aubrey, B. (1991) 'Les conditions de possibilités du réseau', in I. Orgogozo (Ed.) *Les paradoxes du management*, Les Editions d'Organisation.

Bíba, J. and Vokřínek, J. (2006) 'Agent contracting and reconfiguration in competitive environments', *Cybernetics and Systems*, Austrian Society for Cybernetics Studies, Vol. 2, pp.527–532.

Bíba, J., Vokřínek, J. and Hodík, J. (2008) 'Renegotiable competitive contract net protocol', Research Report, Prague: CTU FEE, Department of Cybernetics, The Gerstner Laboratory, GL 194/08, ISSN: 1213-3000.

Camarinha-Matos, L.M. and Afsarmanesh, H. (1998) 'Virtual enterprises: life cycle supporting tools and technologies', in A. Molina, J.M. Sánchez and A. Kusiak (Eds.) *Handbook of Life Cycle Engineering: Concepts, Tools and Techniques*, Kluwer Academic Publishers, pp.535–571.

Camarinha-Matos, L.M. and Afsarmanesh, H. (2001) 'Virtual enterprise modeling and support infrastructures: applying multi-agent system approaches', in M. Luck, V. Mařík, O. Štěpánková and R. Trappl (Eds.) *Multi-Agent Systems and Applications*, *9th ECCAI Advanced Course ACAI 2001* and *Agent Link's 3rd European Agent Systems Summer School, EASSS 2001*, Prague, Czech Republic, 2–13 July, Selected Tutorial Papers, Springer, Vol. 2086 of Lecture Notes in Computer Science, pp.335–364.

Capó, J., Lario, F.C. and Ortiz, A. (2004) 'Proposal of an advanced model for supply chain management in construction industries, based on the virtual enterprise', in K-D. Thoben, K.S. Pawar and F. Weber (Eds.) *Proc. of the Tenth Int. Conf. on Information and Computation Economies ICE-2004*, Centre for Concurrent Enterprising, pp.395–402.

Esteban, L.S. (2007) 'Agent communication using web services, a new FIPA message transport service for Jade', *Multiagent System Technologies. 5th German Conference, (MATES-2007)*, 24–26 September, Leipzig, Germany: Springer, Vol. 9.

Faisst, W. (1997) 'Information technology as an enabler of virtual enterprises: a lifecycle-oriented description', *Proceedings of the European Conference on Virtual Enterprises and Networked Solutions*, Paderborn, Germany, April.

Fischer, K., Müller, J.P., Heimig, I. and Scheer, A-W. (1996) 'Intelligent agents in virtual enterprises', *Proceedings of the First International Conference on the Practical Application of Intelligent Agents and Multi Agent Technology (PAAM)*, Blackpool, UK, pp.205–223.

Gruber, M. and Nöster, M. (2005) 'Investigating structural settings of virtual organizations', in K.S. Pawar, F. Weber, K-D. Thoben and B. Katzy (Eds.) *Proc. of the 11th Int. Conf. on Concurrent Enterprising ICE-2005*, Centre for Concurrent Enterprising, June, pp.245–252.

Hodik, J., Vokrinek, J., Biba, J. and Becvar, P. (2007) 'Competencies and profiles management for virtual organizations creation', *Multi-Agent Systems and Applications V, 5th International Central and Eastern European Conference on Multi-Agent Systems, CEEMAS 2007*, Leipzig, Germany, 25–27 September, Springer-Verlag, ISBN: 978-3-540-75253.

Lomuscio, A.R., Wooldridge, M. and Jennings, N.R. (2003) 'A classification scheme for negotiation in electronic commerce', *Group Decision and Negotiation*, January, Vol. 12, No. 1, pp.31–56.

Mařík, V. and McFarlane, D.C. (2005) 'Industrial adoption of agent-based technologies', *IEEE Intelligent Systems*, Vol. 20, No. 1, pp.27–35.

Molina, A., Flóres, M. and Caballero, D. (1998) 'Virtual enterprises: a Mexican case study', in L.M. Camarinha-Matos, H. Afsarmanesh and V. Mařík (Eds.) *Intelligent Systems for Manufacturing: Multi-Agent Systems and Virtual Organizations, Proceedings of the BASYS – 3rd IEEE/IFIP International Conference on Information Technology for Balanced Automation Systems in Manufacturing*, Prague, Czech Republic, Kluwer, Vol. 130 of IFIP Conference Proceedings, August, pp.159–170.

Neubert, R., Langer, O., Görlitz, O. and Benn, W. (2001) 'Virtual enterprises – challenges from a database perspective', *ITVE '01: Proceedings of the IEEE Workshop on Information Technology for Virtual Enterprises*, Washington, DC: IEEE Computer Society, pp.98–106.

Ovcharenko, S., Alibhai, Z., Ng, C., Gruver, W.A. and Sabaz, D. (2006) 'Implementation of a wireless distributed intelligent system', in V. Mařík, W.A. Gruver, M. Pěchouček and L. Přreučil (Eds.) *IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications (DIS'06)*, Los Alamitos, CA: IEEE Computer Society, June, pp.207–212.

Overeinder, B.J., Verkaik, P.D. and Brazier, F.M. (2008) 'Web service access management for integration with agent systems', *Proceedings of the 2008 ACM Symposium on Applied Computing SAC '08*, Fortaleza, Ceara, Brazil, 16–20 March, New York, NY: ACM, pp.1854–1860.

Pěchouček, M., Doubek, J., Vokřínek, J. and Rehák, M. (2008) 'Incrementally Refined Acquaintance Model (IRAM) for request based virtual organization formation', *Distributed Human-Machine Systems*, Praha: CTU Publishing House, pp.182–187, ISBN: 978-80-01-04027-0.

Perugini, D., Jarvis, D., Reschke, S. and Gossink, S. (2007) 'Distributed deliberative planning with partial observability: heuristic approaches', *Integration of Knowledge Intensive Multi-Agent Systems 2007*, Waltham, MA: IEEE, pp.407–412, ISBN: 1-4244-0945-4.

Petersen, S.A., Divitini, M. and Matskin, M. (2001) 'An agent-based approach to modelling virtual enterprises', Special issue on 'Enterprise Modeling', *International Journal of Production, Planning and Control*, Vol. 12, No. 3, pp.224–233.

Preece, A. (2001) 'Supporting virtual organisations through knowledge fusion', *International Journal of Intelligent Systems in Accounting, Finance, and Management*, Vol. 10, pp.25–37.

Roberts, R., Svirskas, A. and Matthews, B. (2005) 'Request Based Virtual Organisations (RBVO): an implementation scenario', *Collaborative Networks and their Breeding Environments*, Springer, Vol. 186 of IFIP, pp.17–25.

Rocha, A.P. and Oliveira, E. (1999) 'An electronic market architecture for the formation of virtual enterprises', *PRO-VE 99 IFIP/PRODNET Conference on Infrastructures for Industrial Virtual Enterprises*, October.

Sandholm, T.W. and Lesser, V.R. (2001) 'Leveled commitment contracts and strategic breach', *Games and Economic Behavior*, April–May, Vol. 35, Nos. 1–2, pp.212–270.

Shao, J., Gray, W.A., Fiddian, N.J., Deora, V., Shercliff, G., Stockreisser, P.J., Norman, T.J., *et al.* (2004) 'Supporting formation and operation of virtual organisations in a grid environment', *Proceedings of the UK e-Science All Hands Meeting (AHM)*, pp.376–383.

Smith, R.G. (1980) 'The contract net protocol: high level communication and control in a distributed problem solver', *IEEE Transactions on Computers*, December, Vol. 29, No. 12, pp.1104–1113.

Tagg, R. (2001) 'Workflow in different styles of virtual enterprise', *ITVE '01: Proceedings of the IEEE Workshop on Information Technology for Virtual Enterprises*, Washington, DC: IEEE Computer Society, pp.21–28.

Trienekens, J.J.M., Bouman, J.J. and van der Zwan, M. (2004) 'Specification of service level agreements: problems, principles and practices', *Software Quality Journal*, March, Vol. 12, No. 1, pp.43–57.

Van Wijk, J., Geurts, D. and Bultje, R. (1998) 'Seven steps to virtuality: understanding the virtual organisation processes before designing ICT support', *Objects, Components and the Virtual Enterprise '98. An Interdisciplinary Workshop*, http://www.cs.tcd.ie/ Virtues/ ocve98/ proceedings/ index.html.

Vokřínek, J., Hodík, J., Bíba, J., Vybíhal, J. and Pěchouček, M. (2007) 'Competitive contract net protocol', *LNCS 4362 – SOFSEM 2007: Theory and Practice of Computer Science*, Berlin: Springer-Verlag, pp.656–668.

Wiendahl, H-P. and Lutz, S. (2002) 'Production in networks', *CIRP Annals – Manufacturing Technology*, Vol. 51, pp.573–586.

Wooldridge, M. and Jennings, N.R. (1999) 'The cooperative problem-solving process', *Journal of Logic and Computation*, August, Vol. 9, No. 4, pp.563–592.

Notes

1 The Foundation for Intelligent Physical Agents (FIPA) is an IEEE Computer Society standards organisation that promotes agent-based technology and the interoperability of its standards with other technologies, http://www.fipa.org.

2 http://www.panda-project.com/

3 Enterprise Resource Planning/Customer Relationships Management (ERP/CRM).

4 http://labs.jboss.com/drools/

5 For example, the one of the most used agent platforms JADE provides the web-services integration gateway WSIG for transparent integration of agents and WS, jade.tilab.com/doc/ tutorials/JADE_WSIG_Guide.pdf.

6 See http://java.sun.com/webservices/ for more details.

7 See http://java.sun.com/products/javawebstart/ for more details.