Negotiating and Enacting Contracts for Business Networks

Peter Rittgen

School of Business and Informatics
University College of Borås - 501 90 Borås, Sweden
Phone: +46-33-435-5930 - Fax: +46-33-435-4007
peter.rittgen@hb.se

Abstract

Flexibility is a major issue in business today and one answer to that is to organize work in networks of companies rather than within a single business. Cooperation between organizations is more easily established and adapted than reorganization of internal company structures. But how do we manage flexibility in such a network without sacrificing the minimum stability that is required for economic success? We suggest a method for coordinating interaction in a business network that provides three levels of flexibility: local ad-hoc change, global negotiated change and planned change. This method is based on the negotiation and enactment of process-based interorganizational contracts.

Keywords: network coordination, interorganizational workflow, contract negotiation

1. Introduction

The increasing pace of changing market demands puts pressure on the flexibility of business processes. Many organizations find it difficult to constantly adapt their own structures to these changes while maintaining a sound economical basis. A certain stability, i.e. inflexibility, is even a prerequisite for successful business [1]. In this situation it is reasonable to locate flexibility outside a company's borders to some extent by changing and adapting the cooperations with business partners. This will lead us to a business network that works together to provide a common portfolio of complementary products and services. But by externalizing the problem of flexibility we have not yet solved it. Now we have to ask the question of how the coordination of such a network can be achieved in a flexible way.

In a business network the coordination effort is much higher than in a conventional supply chain where a company can focus on the relation to a few immediate major suppliers. In a network coordination is also required among suppliers, i.e., we move from a tree structure to a graph topology which implies a new balance between market and hierarchical coordination. The general problem has been studied in theory, most notably in Agency Theory [2-5] and Transaction Cost Economics [6-10]. Organizations in a supply chain are assumed to choose their organizational structure and network of trading partners in such a way that the sum of both costs is minimized. Regarding information technology, early work by Malone et al. [11] suggested that it will lower transaction costs and therefore, ceteris paribus, lead to an increase in market coordination. Later work posited that organizations will "move to the middle", i.e., to "more outsourcing, but from a reduced set of stable partnerships" [12] if non-contractible issues (e.g. quality and trust) play an important role. Empirical evidence [13] shows that companies often operate in a "mixed mode" blending aspects from both markets and hierarchies.

But the majority of these studies were performed in the context of conventional supply chains. In the face of a network topology the balance between hierarchical and market coordination needs to be readjusted: In the absence of a central coordination unit we typically use the contract as an instrument for coordination. Agency Theory suggests two principal forms of contracts, behavior-based contracts and outcome-based contracts. In the case of a behavior-based contract the principal checks each activity that the agent is supposed to perform, with outcome-based contracts the principal only checks whether the principal delivers the agreed output. If the costs for monitoring agent behavior are high, an outcome-based contract is often superior because an unobserved agent is assumed to shirk (i.e. underperform). An outcome-based contract can be seen as a special case of a behavior-based contract where the outcome is the only observable behavior of the agent. Because the costs for monitoring agent behavior have become marginal in many cases due to the omnipresence of information technology we will focus our investigation on behavior-based contracts incorporating models of the interorganizational process.

In this paper we suggest a method for coordinating interactions in a business network. We begin by outlining the method in section 2. Section 3 defines three levels of flexibility and their roles in network coordination. Sections 4 to 7 describe the steps M1-M4 of the method, respectively. Section 4 starts with the first step, "M1: Modeling the Interorganizational Process". It shows the relevant diagrams from the DEMO language and how they can be used to derive an initial model of the cooperation process. Section 5 deals with the second step, "M2: Negotiating the Contract". It describes the negotiation process, criteria for selecting an appropriate negotiation model and the execution of this model in the context of the given example.

In section 6, "Simulating the Process", we show how simulation can be used to identify process parameters and to improve process performance. In order to do so we need to derive a simulation model based on the agreed transaction models. The results from the simulation can then help to determine certain parameters such as unit prices. They can also give us feedback on how to change the models to make the process more efficient by "playing" with alternative process organizations.

Section 7, "Deriving the Workflow Models", describes how the transaction models of the contract can be enacted. This involves the development of the collaboration model and the business rules. The former is the stable part of the interorganizational workflow and can be used as a template to control a workflow engine. The latter specifies rules for dealing with exceptions. Together they allow us to manage the interactions between the network members on the operational level.

All examples and figures used in this paper are excerpts from the real models we designed in the course of a consulting project where we tested the feasibility of our approach. The section 8, "A Case Study", gives further details on this project. Last but not least we conclude this paper by summarizing the main arguments and outlining future research.

2. A METHOD FOR COORDINATING BUSINESS NETWORKS

The overall method that we suggest is divided into 4 steps (see Fig. 1) which are as follows:

M1: Initial modeling of the interorganizational business process as an interaction model

M2: Negotiating the frame contract as a set of transaction models (based on the interaction model)

M3: Simulating the process models to determine process parameters and to improve process performance

M4: Deriving the workflow models to govern the interaction in the business network

The next section explains this method in a general way. After that we discuss how this method can support flexibility on three different levels. Detailed descriptions of each step follow in later sections.

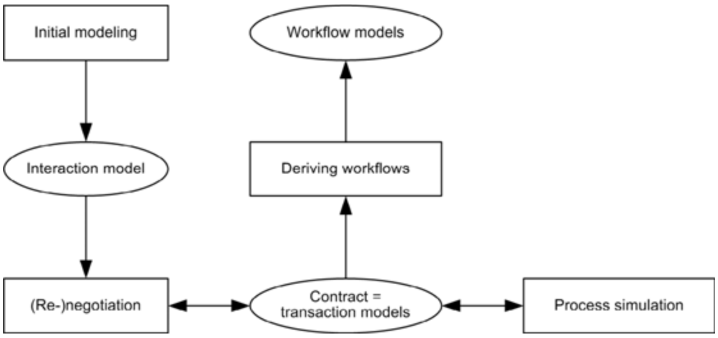

Figure 1: A method for negotiating and enacting a contract

A behavior-based approach requires a detailed analysis of the interactions between (and within) organizations. The processes in a business network are collaborative, i.e. they require that the actions in the different businesses are coordinated with each other. Coordination is essentially a communicative activity. The participants must communicate to determine what is supposed to be done next. This means that we need to look at the structure of the communication in order to understand how interorganizational activities are coordinated. For example, a simple coordination could consist of one partner making a request and the other promising to fulfill it. Together these two actions constitute a commitment, which is a mutual agreement on some activity to be carried out. The request and promise actions are communicative actions or language acts, as they are sometimes called. There is a substantial body of research dealing with language action, e.g. Speech Act Theory (Austin, 1962; Searle, 1969) and the Theory of Communicative Action (Habermas, 1984). These theories can help us to understand coordination and collaboration via the structure of human language that is at the root of these processes. The mentioned theories can contribute to the development of modeling languages that support the description of collaborative processes. Such modeling languages have indeed been developed, e.g. Action Workflow [14, 15] or Conversation for Action [16]. Among these modeling languages there is one called DEMO (Dynamic Essential Modeling of Organization) [17-19]. We have chosen this particular language because it represents the most elaborate and formalized approach which fits well with the principles of the method we suggest. More details about the advantages of DEMO can be found in the section "Step M1: Modeling the Interorganizational Process".

One of the models in DEMO is the interaction model which describes how organizations or organizational units interact with each other. This can form the basis for the development of more detailed models of collaboration called transaction models. The design process is structured and systematic which makes it less likely that (potentially important) details are overlooked. The resulting transaction models will then be the starting point for the negotiation process. If a business process analysis has not been performed we can, in principle, skip the initial step of modeling and start the negotiation process with an empty set of models. This implies that the process of modeling is interpreted as a pure negotiation process with the aim of reaching a consensus model (or models) among the participants. But empirical evidence shows that models of suitable quality can be designed more efficiently using the traditional chauffeured approach [20] or groups of subject matter experts supported by a modeling expert [21]. We therefore suggest an initial modeling step. Negotiation is then used to detail the transaction models. This is done by negotiators from each member of the business network with the help of a negotiation support system. Such systems already exist but have to be extended to cover the negotiation of full-blown models instead of individual selling contracts.

3. THE LEVELS OF FLEXIBILITY

The simplest form of flexibility is *local ad-hoc flexibility*. It concerns only two partners and it responds to a change in their mutual interaction that can be self-induced (e.g., finding a more efficient way to cooperate) or a reaction to a demand by other partners or customers. Such local change can be negotiated by the two affected partners alone which requires significantly less effort than a network-wide negotiation. The required change can be implemented by adding, removing or modifying the business rules alone as the relatively stable behavior in the collaboration model is not affected. This implies that the scope of ad-hoc change is limited but the speed of adaptation is high.

A more fundamental change can be effected by *global negotiated flexibility*. It involves far-reaching adaptations that affect the whole network and it responds to substantial changes in the network's environment. Such a change requires that all partners pass through the complete negotiation-enactment cycle once (steps M2, M4 and possibly M3). As a consequence the contract, the collaboration model and the business rules have to be adjusted. This requires some effort and as a result the adaptation speed is slower than in the ad-hoc case but the scope of global change is broad.

The highest degree of flexibility is *planned flexibility*. It concerns the whole network and it responds to expected changes in the near future (e.g., qualitative and quantitative changes in the demand). It implies that we have to reconsider the point of departure for the network as laid down in the Interaction Model (step M1). Simulation of the process with the expected parameter changes (step M3) can give valuable clues regarding process reorganization. Via negotiation we can then decide for a new process model (i.e., contract) in step M2 (thereby slightly rearranging the original order) and enact this contract according to step M4. The required effort is similar to that of setting up a new network so that a careful assessment of the opportunities and risks is essential.

4. STEP M1: MODELING THE INTERORGANIZATIONAL PROCESS

At the core of the language-action perspective is the Speech Act Theory by Austin and Searle [22, 23]. The central premise of this theory claims that language is not only, and not even primarily, a medium for exchanging information. Instead it is a means of action. Uttering something is actually doing something. We can instruct, direct, request, make commitments, promise, apologize, declare marriage and the like, all by just saying a few words. Each language action consists of an illocution describing the kind of action and a propositional content referring to the object of the action. [24] embedded this theory of speech acts into a social context whereby language action becomes social action. In his Theory of Communicative Action he argued that each action is determined by the roles that the actors play and the (power) relation they have towards each other. For this purpose he introduced validity claims implying that an actor makes such a claim, implicitly or explicitly, when performing a language action.

In an organizational setting communication is often aimed at the performance of a specific action to achieve some objective. Early templates for such goal-driven conversations are the conversation-for-action schema [16] and the action-workflow loop [14, 15]. More advanced examples of such frameworks are: Dynamic Essential Modeling of Organizations (DEMO) [17-19], action-based modeling [25], Business Action Theory and SIMM [26-28].

4.1. DEMO

The Language-Action Perspective offers many approaches some of which we have already mentioned. We have chosen DEMO because it offers transactional patterns not only in the meta-language but also as concepts in the modeling language itself. This allows us to distinguish between transactions (as complex communicative actions) and speech-acts (as elementary actions) which is essential for our approach (see section “Transaction Models”). In DEMO, all acts that serve the same purpose are collected in a *transaction* in which two roles are engaged: the *initiator* and the *executor*. Each transaction is assumed to follow a certain pattern which is divided into three sequential phases and three layers. The layers are: success, discussion and discourse. The success layer describes the actions that have to be carried out to complete the transaction successfully, i.e. assuming that nothing unexpected happens. It consists of the phases: *order* (O), *execute* (E) and *result* (R). If anything goes wrong on the success layer, the participants move to the discussion layer to decide on some action that resolves the conflict. This layer can be seen as a meta layer that allows for exception handling. If the conflict cannot be resolved on this level we move to the discourse layer (or meta-meta layer) where new social rules are established that govern problem solving on the discussion layer. For more details on these layers see [29].

4.2. DEMO’s Interaction Model

The interaction model shows actors and transactions. The actors are roles that are enacted by a person, an organizational unit or a whole organization. Fig. 2 shows the Interaction Model of our case. The main actors are the Logistics Provider, the Headquarters of the retailer and the Shop. The latter two maintain a very close, franchise-like relationship but are nevertheless organizations in their own right.

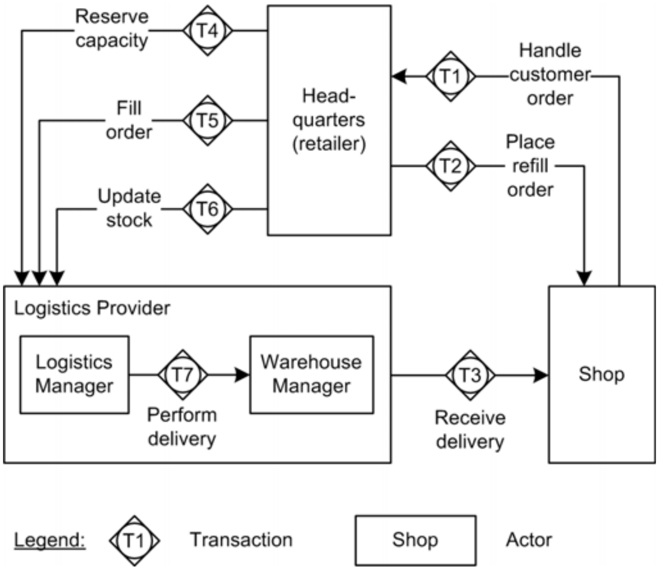

Figure 2: Interaction model

A transaction is represented by a diamond with an inscribed circle that contains the number of the transaction. An undirected arc connects it with the initiator, an arrow points from it to the executor. Fig. 2 describes the process of capacity reservation and order handling among these organizations. It starts when Headquarters reserve capacity for handling a certain amount of ordered items in advance of the actual order (T4). The logistics company (LogCom) allocates staff and space so that the reserved capacity can be provided at the time the respective order arrives. But the capacity required by the order might actually be higher or lower than the one that was reserved.

Orders can be initiated either by Headquarters or by the Shop. The former happens when the Shop is running low on certain products. Headquarters will in such a case suggest to the Shop to place a refill order (T2). For this purpose they send an order proposal containing the products in question which, after possible changes and/or additions is returned. If customers ask for specific products, the Shop can also place a so-called customer order (T1). Headquarters will forward both types of orders to LogCom (T5) who will then perform delivery (internal transaction T7) which includes picking items, packing them and handing them over to the carrier. The physical delivery is largely non-communicative and it is therefore not explicit in the Interaction Model. Instead the coordinative part is shown, namely that the Shop receives the delivery (T3) which confirms that LogCom have fulfilled their obligation. From time to time Headquarters will also ask for an update of the stock (T6). This is necessary to synchronize their warehouse management system with that of LogCom.

4.3. Transaction Models

A transaction is the smallest unit of activity that delivers a meaningful result. It can be negotiated independently and is therefore also the unit of modeling. Much of the detailed behavior that constitutes a business process is hidden inside each transaction. For the contract it has to be brought to light to operationalize it. A transaction in DEMO is made up of a number of speech acts and an objective action which follow a certain pattern. This pattern is not a rigid template that claims to fit every transaction. It is rather a guideline that describes a common conversational structure that can help us in analyzing a particular situation. In some cases it will describe the situation fairly accurately, in others we might have to revise it or even to develop a new one that is specific to that particular situation.

The pattern consists of the phases mentioned above: order, execute and result. The order conversation (O phase) has at least two elements: a *request* and a *promise* (see fig. 3) but longer negotiations (including a failure) are possible. If an agreement was reached in the order phase, the objective action (E phase) is executed and the result conversation (R phase) is entered. As a minimum this can consist of the speech acts *state* and *accept*. Fig. 3 summarizes these steps which are performed in the order that is indicated by the leading numbers. A model that contains only actors, speech acts and objective actions is called a speech act model. A speech act model that contains only actions and actors belonging to one transaction is called a transaction model.

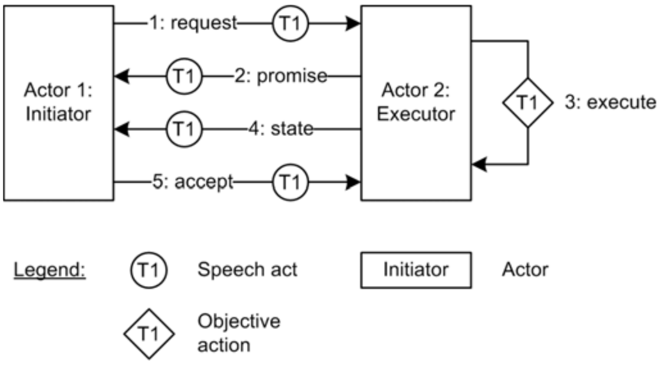

Figure 3:Speech act model of a transaction (transaction model of T1)

Fig. 4 shows the complete, minimum speech act model of the interaction model in fig. 2. As speech act models can be very complex for realistic cases we will usually refer to a set of transaction models instead. The section “Step M4: Deriving the Workflow Models” shows how these can support the development of contracts in general and the business rules and collaboration models in particular.

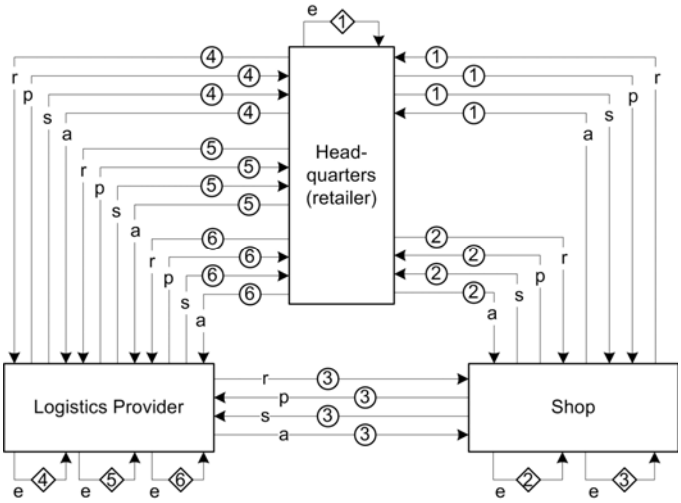

Figure 4: Complete speech act model of the interactions

5. Step M2: Negotiating the Contract

5.1. The Process of Negotiation

We define negotiation as the process whereby a group of two or more individuals tries to reach an agreement on the performance of future actions. The individuals are human beings that might act on behalf of organizations or on their own behalf. For the purpose of this paper we focus on electronic negotiations, i.e. negotiations that are supported by information and communication systems. They can be divided into three different types: bargaining, auction and agent negotiation [30]. Auctions are very common, especially in electronic commerce. They assume that the traded products or services can be described in detail and are hence comparable. The auction proceeds in the form of a bidding process where potential buyers can make (money) offers for a certain product or service. There are different models to organize the bidding process [31]. A comprehensive classification of negotiations with respect to auctions is provided by the Montreal taxonomy [32]. Agent negotiation means that an inanimate agent, i.e. a software artifact, carries out the process of negotiation on behalf of a principal, typically a human being. The principal delegates the task of negotiating to the agent by providing it with his or her preferences regarding the product or service to be procured. The agent has certain autonomy to act within the limits of these preferences. Some models for agent negotiation are given in [33]. The specification of preferences requires that the product or service in question can be described in detail. Hence both auctions and agent negotiation only work with standardized products / services.

The models we have discussed so far assume that most parameters of the contract are already predetermined and very few can actually be negotiated. Most often the only free parameter is the price. In many cases this restriction is not acceptable, i.e. we need more freedom in negotiating. This can, for example, happen if the product or service to procure is not standardized so that we have to negotiate many of its parameters. In such a case we need the third model, bargaining. In bargaining we assume that in principle all parts of a contract are negotiable, i.e. we start with an empty contract (although existing reference contracts or contract templates can be used as a starting point if desired). A number of bargaining models has been suggested such as the Three-Layer Architecture [34], SilkRoad [35], DOC.COM [36], MeMo Business Negotiation Support Metamodel [37], Protocols for Electronic Negotiation Systems [38], and the Generic Model [39]. To find a suitable negotiation model for business networks we must first identify the criteria that it should fulfill. Based on the characteristics of a business network mentioned above we have derived the following criteria as core criteria: Communication, documents, deontic logic (the logic of obligations, permissions and the like) and time. The 'implementation' of other criteria, e.g. the ones mentioned in [30], is subject to further research. The next section argues for the necessity of the chosen criteria.

5.2. Criteria for a Negotiation Model

In this section we discuss the criteria for choosing an appropriate negotiation model, i.e. one that fits the negotiation of frame contracts for interaction in a business network. The four chosen criteria are italicized and we argue why they are essential for this purpose. We then go on to identify a negotiation model that meets these criteria best. This is done with the help of a study that actually considered a larger set of criteria (but containing "our" criteria as a subset).

Communication is the primary instrument for social interaction in general and for negotiation in particular.

Negotiation consists basically of an exchange of messages between the negotiators. With these messages the negotiators create, modify and extend the contract, e.g. by making requests or commitments that ultimately lead to contractual obligations. It is therefore evident that a negotiation model for business network contracts must incorporate communication on a fundamental level. The importance of language for social action has been recognized early which led to the development of several theories, most notably Speech-Act Theory [22, 23] and the Theory of Communicative Action [24]. Many of the negotiation models that address the issue of communication are based on these theories.

The result of negotiation is a contract, which is obviously a *document*. But documents play an important role already during negotiation in the form of messages exchanged in negotiation and contract versions that are required to understand why the contract has developed in that particular way and to go back to an earlier version if something has gone wrong. Finally, d*eontic logic* is essential for reasoning about the obligations contained in a contract and *time* issues frequently arise in process contexts.

[30] has performed an evaluation and comparison of 11 negotiation models with respect to 11 criteria among which the above mentioned criteria can also be found. The closest match to the requirements for a business network negotiation model is represented by DOC.COM [36] which fulfills three of the four criteria fully and one, deontic logic, at least partially. We have therefore chosen to adopt this model for the purpose of our study. As deontic logic is an important issue we have decided to add respective functionality to the negotiation system. But there is yet another problem that needs to be solved. The objective of DOC.COM is to represent a negotiation about the execution of a process instance, e.g. the delivery of a particular item on a particular date. But negotiations regarding the governance of a business network concern process types, e.g. the general business logic of order processing. The resulting contract is called a frame contract as it regulates the interaction among network members regarding a significant number of orders over time. To enable such negotiations we have introduced a meta-layer into the negotiation language. Figure 1 shows the architecture of a method to govern a business network.

5.3. Negotiation in the Business Network

A business network consists of a number of network members. Each such member is typically an organization (i.e. a business) but could also be an individual who acts as an economic agent. Each member organization is represented by a negotiator who is entitled to carry out such negotiations and to sign a binding contract on behalf of the organization. This negotiator will interact with negotiators from the other network members via a negotiation support system (NSS). This system coordinates the negotiation activities such as making proposals, discussing proposals, voting on them and making the final decision. The NSS consists of a message component and a contract component. The former handles both the translation of "human" negotiation messages into the formal representation in DOC.COM and the presentation of recorded formal negotiations in a human-readable form. The contract component stores the binding negotiations, which together make up the contract and which are also stored in DOC.COM, and represents this contract in a way that is similar to conventional, written contracts. The specific NSS for DOC.COM is called Negoisst [40]. The next section describes how negotiation and contract formation proceed.

The left part of Figure 1 shows how the operation of a business network is supported. We assume that the process of negotiation has led to a contract that deals with all relevant issues of the cooperation. This could be the negotiation of a completely new frame contract, i.e. the set-up of a new business network, or adaptation to changes. The contract under consideration will in any case be subject to implementation which yields a description of the interactions between the members in a workflow language. The choice of this language depends on the workflow system that we choose. In principle any workflow system can be used that allows for the implementation of the workflow patterns identified in [41]. Most commercial systems qualify if we allow for workarounds and coding but there is little native support for many of the advanced patterns. For our prototype we used YAWL [42] because it provides most patterns, together with the YAWL Engine. YAWL uses the same serialization language as the negotiation and contract language DOC.COM, i.e. XML. This facilitates the implementation of the contract. YAWL makes also use of XQuery and XPath to extract data from XML input files and for generating XML output. This supports the integration with the enterprise application systems of the network members, most of which can import and export in XML format. The resulting workflow system is run on a coordination server. An overview and comparison of other languages for interorganizational workflows is given in [43].

5.4. From Negotiation to Enactment

The previous section has described the general method of governing a business network. In this section we describe how the procedures in that method are performed and what the results look like. For this purpose we consider a simple negotiation, the corresponding part of the contract and the resulting workflow net (enactment) in some detail. This example represents only a very small part of the case and just serves to illustrate the way our approach works. The complete example is shown in the next section on a more general level. The benefits of translating the contract into a workflow model are improved control and monitoring of the interorganizational process. The workflow engine makes sure that each network member knows what to do and delivers the specified results in time. It can also inform the affected party of delays or failures to comply with the agreed procedures.

Our case involves three business partners: A retail chain in the home decoration industry (RetCom), the shops of this chain and a logistics company (LogCom). RetCom want that LogCom take over the delivery of orders for them. Fig. 5 shows two steps in the respective negotiation between them. The representative from LogCom writes an email saying that they need a capacity reservation 2 weeks in advance of the order to be able to handle it. The negotiation support system helps with translating this request from the natural language to the internal, formal representation in DOC.COM:

REQUEST (Reserve_capacity[ORDER], t DATE[ORDER] – 14)

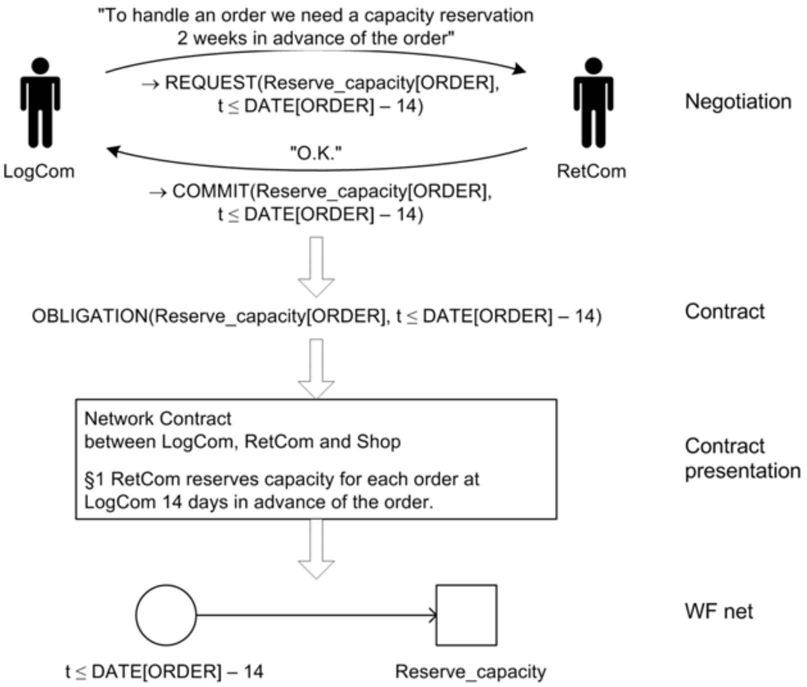
"To handle an order we need a capacity reservation 2 weeks in advance of the order"
→ REQUEST(Reserve_capacity[ORDER], t ≤ DATE[ORDER] – 14)
"O.K."
→ COMMIT(Reserve_capacity[ORDER], t ≤ DATE[ORDER] – 14)
LogCom
RetCom
Negotiation
OBLIGATION(Reserve_capacity[ORDER], t ≤ DATE[ORDER] – 14)
Contract
Network Contract
between LogCom, RetCom and Shop
§1 RetCom reserves capacity for each order at LogCom 14 days in advance of the order.
Contract presentation
WF net
t ≤ DATE[ORDER] – 14
Reserve_capacity

Figure 5: *From negotiation to workflow net (example)*

The keyword REQUEST indicates that LogCom would like to introduce a new action into their cooperation. The propositional content of this message tells us what that action is, namely the reservation of capacity for each order. The request also specifies a time restriction for this action, i.e. 14 days in advance of the order date. This message is stored in the message memory of the negotiation system so that it can be matched with RetCom's reaction to it. In this case RetCom fully agree with the action that was suggested by LogCom by answering with "O.K.". Again the NSS will help with translating this to the formal representation:

COMMIT (Reserve_capacity[ORDER], t DATE[ORDER] – 14)

The speech act COMMIT signals that RetCom agree to fulfill the request. A request that is followed by a commit with the same propositional content and restrictions leads to a binding obligation of the committing party towards the requesting party with respect to the content. An alternative reaction of RetCom could be:

COMMIT (Reserve_capacity[ORDER], t DATE[ORDER] – 7)

which would be interpreted as: "We agree to reserve capacity but we cannot do it earlier than one week in advance." Such a speech act does not create an obligation but constitutes a counter-offer. An acceptance of this counter-offer by LogCom would then create an obligation concerning the modified terms. In our example the original request is granted and a respective obligation is inserted into the contract:

OBLIGATION (Reserve_capacity[ORDER],
t DATE[ORDER] – 14)

The presentation component of the NSS can at any point in time display the contract that has been negotiated so far in a human-readable form (see Fig. 5). In the final step the obligation is translated to a corresponding workflow.

6. Step M3: Simulating the Process

A contract has to specify general terms and conditions that can be seen as static parameters that control the interaction between the trading partners. Examples for such parameters are pricing, terms of delivery, terms of payment and so on. Determining reasonable values for these parameters is difficult because they depend heavily on characteristics of the interorganizational process. This is particularly true when companies engage in cooperation for the first time and therefore lack prior experience. Let us consider, for example, the pricing of a logistics service, e.g. the handling of one unit of the customer's product. How much we charge for that depends, among other things, on how much it costs us to deliver this service, which in turn depends on the time it takes, the number of workers that are involved, resources that are used etc. One way of assessing the complex interaction of these factors is to simulate the respective business process. The usefulness of business process simulation has been studied thoroughly [44-49], particularly in an interorganizational context [50-52].

A simulation model is an abstracted, formal description of some real or imagined system. A simulation is an enactment of such a model that allows us

a) to observe the potential behavior of a system that does not (yet) exist, or

b) to observe the (potential) behavior of an existing system at a much faster pace and at lower costs than that of the real system and without disturbing it.

If an appropriate abstraction is chosen the results of the simulation will represent a fair approximation of the behavior of the real system (or the imagined system if it were built). With its help we can determine the performance characteristics of the business process. This data can then be used to support the design of the terms and conditions of the contract. For the development of the simulation model we use the approach described in [53]. It is based on a language-action model of the business process and proceeds in three steps:

S1: Designing the business process view,

S2: Designing the resource view,

S3: Designing the simulation model.

The first step involves the design of a flow-like view on the process that excludes the actors in favor of a more precise description of the execution logic. In the second step we develop a view that tells us which resources are required by each action and the final step results in the simulation model which is written in SimPy (Simulation in Python). To give the reader an idea of how this works, steps S1 and S2 are shown in detail in the following sections. As an example we use the pricing of product handling. This issue was given highest priority by both corporate partners in our project. We assumed that the costs of handling a product unit will play an important role in determining the price. We therefore took a closer look at transaction T7, Perform delivery, and the associated transactions T4, Reserve capacity, and T5, Fill order (see fig. 2). When decomposing the transactions into speech acts and productive actions we get the process view as shown in fig. 6.

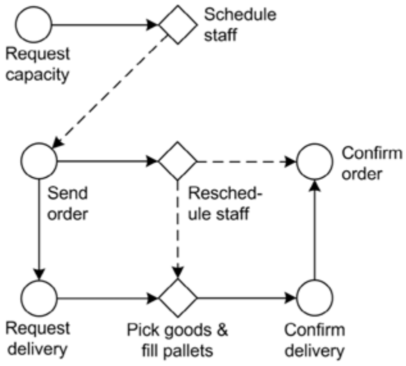

Figure 6: The process view of the interaction model

The business process starts with a request speech act for a certain capacity which is then used to schedule the warehouse staff accordingly (target action). Note that the promise part of the order conversation is omitted because the business rules force the Logistics Provider to accept each request (see next section). After that the availability of the capacity is confirmed (speech act: state). Due to the business rules in the contract (see next section) we can omit the speech acts "promise" and "accept" in this and later transactions. The next step in the process is the order that is sent by the customer. Observe that this action is not caused by the confirmation of the capacity because the customer might decide not to make use of the capacity and not send an order. Hence there is no causal relation. But on the other hand, the order cannot be sent without prior reservation of capacity which makes the relation conditional (dashed arrow). The order information is used to reschedule the staff depending on the actual package load. This might involve extra hours of the outbound staff or that inbound staff is reassigned. At the same time delivery is requested but it cannot be done before the staff is rescheduled, so that sufficient staff is available. The goods are then picked and each pallet is filled with the goods destined for a particular shop. When this has been done the pallets are picked up by the forwarder and delivery is confirmed. This allows us to confirm the completion of the order.

The work described so far was part of the initial business analysis where we also identified problems. A pressing one (from the point of view of LogCom) was related to the discrepancies between planned and actual capacities so we suggested doing a simulation of the process to determine how the transaction costs are affected. This required a resource view (step S2) to get a clearer picture of the resource use of each action. The result is shown in fig. 7.

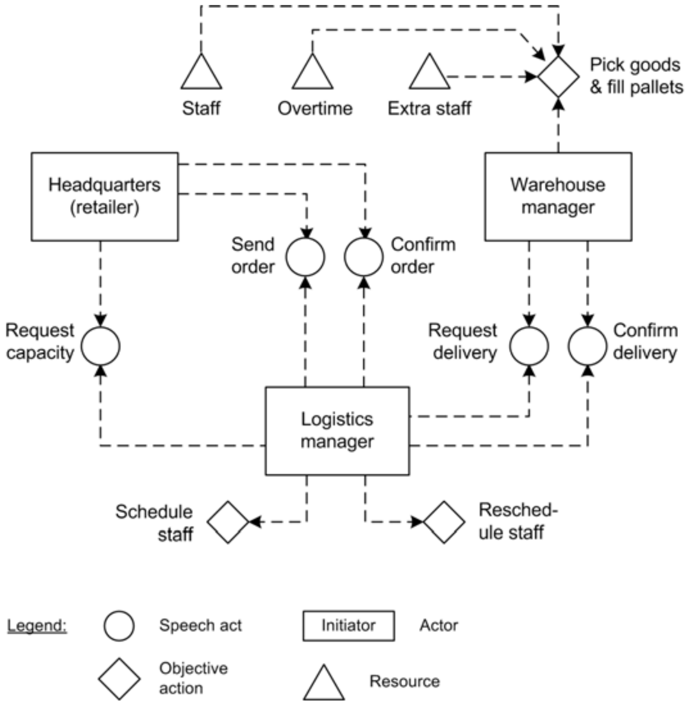

Figure7 : The resource view of the interaction model

The resource view shows the actors and objects that are involved in each action. We assume that an actor who is engaged in an action cannot perform another action at the same time. Most of the information contained in fig. 7 can be derived from the action and process views (figs. 2 and 6, respectively) in the following way: For each action in the process view, find the corresponding transaction in the action view and from there the actors involved (initiator and executor). These are the resources of the respective speech act. The initiator becomes the performer of the request and accept acts and the addressee of the promise and state acts. Likewise the executor will be the performer of the promise and state acts and the addressee of the request and accept acts. If the action is productive we drop the initiator and record only the executor as a resource.

In the example of fig. 7 this procedure yields an almost complete diagram with only three resources missing. These concern the action "Pick goods & fill pallets” that requires additional resources: the scheduled staff, extra staff that might be called in and overtime of the scheduled staff. The use of these resources is associated with certain costs. The time for filling the pallets depends on the actual number of packing units to be handled, the number of available staff (incl. extra staff), the overtime and the time required for handling a unit. The latter is assumed to be normally distributed with given μ and s. Packing units that cannot be handled during the week in question have to be treated in the following week which leads to delays and further overtime. The time for (re)scheduling is also normally distributed with given μ and s. All other actions are assumed to require a negligible time.

A run of the resulting simulation model yielded results for a full year (52 weeks), the first 6 of which are shown in table 1.

Table 1: Excerpt from the simulation results

Reserved capacity	Actual units	Deviation %	Staff	Extra staff	Overtime	Handled units	Total costs	Costs per unit
4841	7366	52%	10	4	29,28	7366	11.385,60 €	1,55 €
5099	4494	-12%	10	0	-40,48	4494	6.000,00 €	1,34 €
4684	4957	6%	9	0	36,56	4957	6.131,20 €	1,24 €
2203	1179	-46%	4	0	-65,68	1179	2.400,00 €	2,04 €
5374	7817	45%	11	3	65,36	7817	11.507,20 €	1,47 €
2525	3564	41%	5	1	45,12	3564	5.102,40 €	1,43 €

The results of this simulation can now be used to make the constant pricing model of the old frame contract variable. This implies that the price for handling a unit is no longer fixed but depends on the accuracy of the capacity forecasts, i.e. the difference between reserved and actual capacity. The advantages of a variable pricing model are twofold. On the one hand it encourages Headquarters to improve the quality of their estimates as inaccurate capacity forecasts will invariably lead to higher logistics costs. This in turn improves the planning situation for LogCom. On the other hand, if deviations do occur, LogCom will get compensation for their increased costs due to insufficient or unused capacity.

For example, to determine a reasonable price for each handled unit we can refer to the unit costs in table 1 and use these figures as the basis for the cost calculation. One approach might be to take the average unit costs as an input for simple cost-plus pricing. Others might be to consider seasonal variations or to make the price depend on the difference between actual and reserved capacity. In our case we applied the latter approach. Using the simulation results mentioned above we get a cost base of 1.34 € plus 3 cents for positive deviations in steps of 10 %, and 1.07 € plus 26 cents for negative deviations in steps of -10 % based on a linear regression on the full results.

7. Step M4: Deriving the Workflow Models

A contract is a formal representation of the cooperation between a number of organizations. It defines the roles that each party to the contract plays and the activities they perform in the context of the cooperation. In principle we could claim that the speech act model already contains most of the information necessary for the dynamic part but this approach is not sufficient for at least two reasons. Firstly this model is typically very complex for realistic cases as the example of fig. 4 (which contains only a small part of the overall model) indicates. It is therefore unsuitable for communicating knowledge about the obligations implied by this process structure to the respective parties. But one of the most important requirements of a good contract is that the parties signing it should be fully aware of its implications.

Secondly the speech act model is hard to implement. It does not give us any directions as to which of its activities are supported by information systems integration and which not. Both issues can be addressed by dividing the behavioral model into two components: *Business rules* and *collaboration model*. The latter is a detailed, workflow-like model of the cooperation. It is structurally very similar to the speech act model but it contains only a fraction of the actions. It shows only standard, routine behavior that can be performed or largely supported by information systems integration. This facilitates the enforcement of the contract.

The business rules then cover exceptional or non-routine behavior. This kind of behavior does not occur often enough to economically justify an integration of the involved information systems. Such behavior would also crowd the collaboration model. It can be better represented in form of a table. The next section describes the development and the use of the collaboration model and the business rules in detail.

7.1. Collaboration Model and Business Rules

When developing the contract we look at each transaction in turn. We first create a speech act model of the respective transaction as described in the section "Transaction Models". The result is a detailed model with all the steps that have to be performed in the transaction. Fig. 8 shows as an example the speech act model that corresponds to transaction T5.

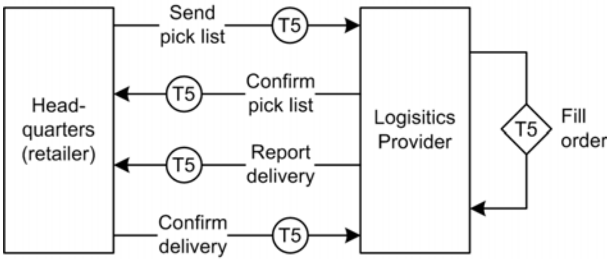

Figure 8: Speech act model of transaction T5

The aim of that transaction is to fill the order, i.e. to deliver the items contained in the order. It starts when Headquarters send a so-called pick list to LogCom. This list names the products to be picked (and delivered) and their quantities. The associated activity is a routine activity and the information is important for controlling the process of filling the order. It will therefore be entered into the collaboration model (see fig. 9). The information systems of Headquarters and LogCom are integrated in such a way that the list is sent electronically as a "pick file".

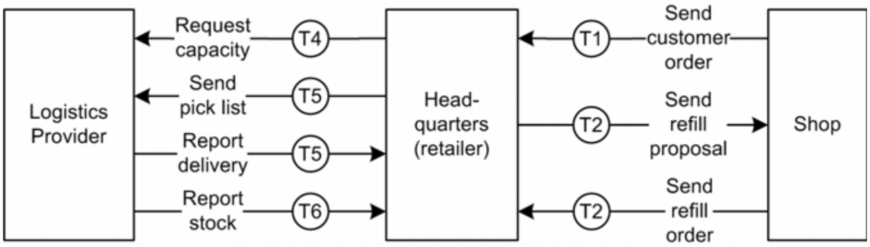

Figure 9: Collaboration model

The next step in transaction T5 is that LogCom confirms the receipt of the pick list. As the warehouse management system of Headquarters mirrors that of LogCom an out-of-stock situation cannot occur. LogCom only has to confirm that enough resources are available (staff, shelf space) to handle the order. As the reserved capacity (T4) is usually sufficient an explicit confirmation is not required but is per default assumed. The respective speech act does therefore not appear in the collaboration model. Instead we create a business rule that is activated in the case of an exception, i.e. if the required capacity does exceed the reserved one by more than the specified percentage value (see table 2, T5, promise). As a special arrangement has to be made for solving this problem in each specific case this activity cannot be supported by information systems integration. The logistics managers at both companies have to negotiate this solution.

Table 2: Business Rules

Trans-action	Phase	Business Rule
T1	promise	A request to deliver items is per default granted and hence not confirmed. In case of out-of-stock a respective notification is sent.
	state, accept	Covered by transaction T3
T2	state, accept	Covered by transaction T3
T3	request, promise	Covered by transaction T1 or T2
	accept	If 'confirm receipt' was O.K. no further message is sent. Otherwise the claim is processed (return/resend).
T4	promise	A request for a capacity (forecast of required capacity) is always accepted and hence not confirmed.
	state, accept	The provision of the requested capacity is guaranteed. Hence no confirmation is required.
T5	promise	The pick list is accepted per default, no confirmation is sent. If the amount of items to be picked exceeds the limit a special arrangement is made (rescheduling of warehouse staff / higher unit price).
	accept	This is implied by the receipt of the delivery. For missing or wrong items a complaint is sent and wrong items are returned to LogCom.
T6	request, promise	The retailer's warehouse system is updated via automatic daily stock report transmissions. Request/promise is therefore obsolete.
	accept	The receipt of the stock report is assumed. If transmission fails, manual trouble-shooting will be invoked.

The objective action "Fill order" is not considered in the contract because it concerns only internal behavior of LogCom. The next step in transaction T5 is that LogCom reports the delivery. This is a routine activity and Headquarters needs this information for billing purposes. It is therefore a part of the collaboration model. The final step, confirm delivery, is implied by the receipt of the delivery (T3). The exceptional case of a wrong delivery is handled by the business rule T5, accept. The same is done for the remaining transactions T1-T4 and T6 leading to the collaboration model in fig. 9 and the business rules in table 2.

7.2. Managing the Interorganizational Workflow

The examples so far were detailed but covered only a small part of the case. Here we give a complete account of the case without the details concerning negotiation. We primarily focus on the "old" architecture of the retail network and the result of applying the method described so far. We started our project with performing an analysis of the business processes between the companies we have already mentioned. These companies had an established business relationship based on a conventional contract. In the analysis we discovered the structure of their cooperation (see Fig. 10) and a number of problems such as: broken interaction patterns, missing business rules, unclear communication structures, different contract interpretations and excessive interpersonal communication. As a consequence the parties were unsatisfied with the current situation.

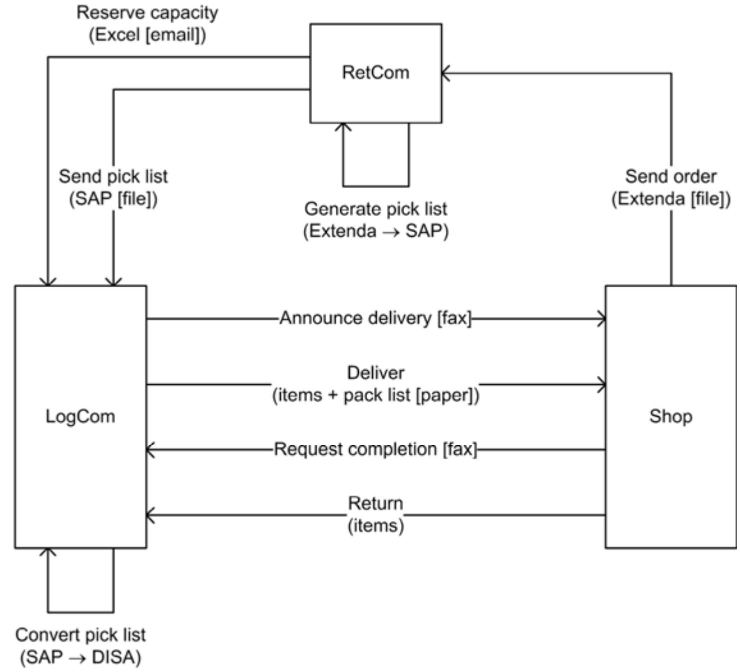

Figure 10: *Original architecture of the network*

To solve these problems we decided to create a new network to coordinate their interaction. We started with negotiating the formal contract. This was done in a seminar where the representatives of each organization were present and the seminar leader manually translated their requests and commitments into a formal representation according to the procedure described above. The reason for this is that the NSS does so far only support bilateral negotiations. We consider this as a technical restriction rather than a conceptual one and it should be possible to extend the NSS to multi-part negotiation. The implementation of the contract was done with the help of YAWL and the YAWL engine which was run on a coordination server that connects all parties. The conversions between the involved formats (SAP, DISA, Extenda and Excel) have been performed with the help of XML Script and the X-Tract XML Script processor. This led to the architecture depicted in Fig. 11.

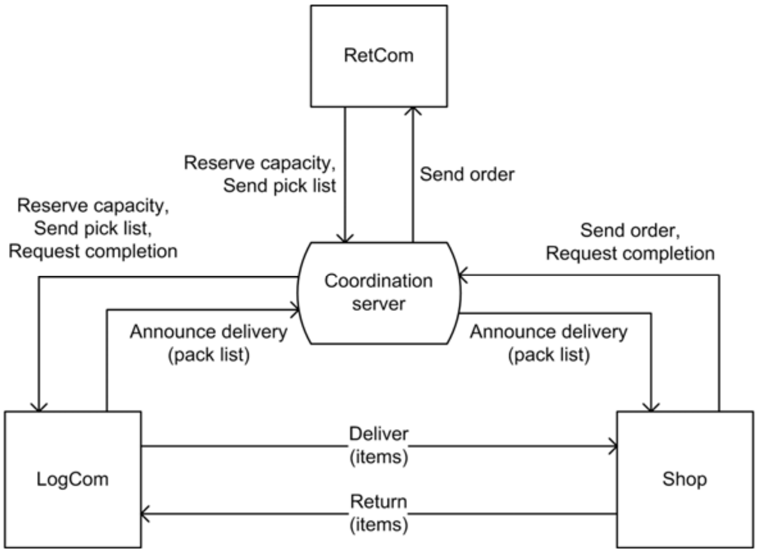

Figure 11: *New architecture of the network*

In the new architecture each network partner only exchanges messages with the coordination server. This reduces the complexity of the coordination considerably. The server takes care of forwarding messages to the right recipients, converting between formats, triggering time-controlled messages and so on. The new network architecture also offers ways to improve the efficiency of the communication. In our case, for example, the paper-based communication can be replaced by electronic messages, e.g. concerning the fax containing the pick list. The physical exchange between LogCom and the Shop can in this way be restricted to the exchange of the items themselves.

We have tested the coordination server with real data from the involved businesses covering their exchange during a whole year. This implies roughly 200 orders and 100 deliveries. We simulated the behavior of the each of the 3 parties manually based on the data. The messages produced by the server matched those that were actually sent according to the empirical data we were provided with by the companies.

8. A Case Study

The proposed method was tested in a project that concerned LogCom, the headquarters of the retail chain and more than 100 shops. In a business analysis we discovered that the network was unable to react in a flexible way to challenges in the environment that were primarily caused by both seasonal and unpredictable fluctuations in demand. One of the aims of that project was to improve the existing contract. As a consequence there was a need for changes that required a re-negotiation of the complete contract for the whole network. This was done with the help of the proposed method as detailed in the next paragraphs. The major changes involved the mending of broken patterns, exception handling, clarification of communication channels, improving understanding of the contract.

The project began by performing step M1. This was done in the form of process-modeling seminars that involved representatives from both companies and from our university. A seminar leader from our university took on the role of a facilitator following a chauffeured-session approach. A minute taker recorded the process models that were agreed upon with the help of a laptop computer. The sessions were also taped to allow for corrections of the material in the case of misunderstandings. Excerpts from the resulting models have already been shown in section 4. These models gave us a very detailed understanding of the current situation at the time and they also provided a basis for the re-negotiation of an improved process model.

The latter comprised step M2 and it was performed in a way similar to M1. We again used process-modeling seminars to detect problems with the old design and to come up with suggestions for a new design. Due to space restrictions the final models are not shown here but they eventually led, after performing steps M3 and M4, to the architecture shown in Fig. 11. The outcome of this step was discussed with representatives from both companies to make sure that the new version is approved by all parties to the contract.

In step M3 we built the simulation model that was shown in section 6. This allowed us to fine-tune some of the parameters of the contract such as pricing, penalties and allowable deviations from forecast capacities. The simulation model was derived from the process models developed in steps M1 and M2 and was run with data that was supplied by the companies, Excel sheets from RetCom and DISA printouts from LogCom. Some of the results have been shown in section 6. Further details may not be disclosed as they concern trade secrets of the involved companies.

The final step M4 was then performed manually in the absence of an appropriate tool. Such a tool can, however, be built to support future applications of this method by ourselves and also to promote the use of this method by other business process consultants. We translated the new process models (M2, M3) to the workflow language YAWL and executed them on the YAWL engine with real data supplied by the companies. Message conversions were done with the help of XML Script and the X-Tract XML Script processor. Running the new architecture was done offline to avoid disturbing the actual operations of the companies. The purpose of this procedure was to validate the method by reproducing the actual communications between the companies. For this purpose we compared the "simulated" messages with the ones that were actually sent by the organizations or their IT systems. All messages concerning the roughly 200 orders and 100 deliveries coincided. The following paragraphs describe the improvements that could be achieved by the application of this method.

There was a pattern of interaction when establishing the framework contract and another one when realizing the business transaction. The interaction pattern that glues framework contract and business process was thus broken. This had the effect that Headquarters could not be sure of the capacity that will be available at the time of order and LogCom did not reserve the required capacity. The estimates made by Headquarters were therefore neither informative nor directive and hence did not imply mutual commitments. Following the proposed method we arrived at a contract that could be translated into a workflow description and therefore ensure that the agreed and performed activities coincided.

The daily routine work was dominated by "negotiating" on an interpersonal level ways to deal with exceptional situations that were not provided for in the original frame contract. This led to a substantial amount of extra work. During re-negotiation these exceptions were considered and implemented as business rules. Under the old network organization it was also unclear in many cases who was supposed to communicate with whom regarding which issue. Business rules take also care of that now.

Different interpretations of the old contract by the parties led to expectations that were not fulfilled. This is avoided now by a clearer, formal contract that allows the parties to trace all activities to the agreement to understand why something is done in a particular way. Minor problems in the interaction between two partners that have been overlooked in negotiation or that have turned up later can be solved now by making ad-hoc changes to those business rules that concern the respective parties.

9. Related Research

Our paper touches on a number of issues that have been addressed by existing research. Primarily we can mention the area of formalized or electronic contracts that have been studied extensively by e.g. [54-56]. Most of these approaches aim at transactional contracts, i.e. they cover only a single business transaction such as a sale. Our approach targets frame contracts that cover a number of similar transactions, i.e. a complete business process.

Another related issue is that of collaborative modeling [57-61]. Our approach can be seen as a specialization of these methods for the specific situation of multi-organizational modeling.

Group negotiation and negotiation support systems are also relevant. They have been studied by e.g. [30, 35, 37, 38, 40]. Again the focus of these approaches is mainly on transactional contracts so that they cannot be used in the context of model negotiation.

Interorganizational workflows are the subject of the following publications: [43, 62, 63]. They are typically also process-based but do not assume the necessity of negotiating these process models.

10. Conclusions

We introduce a method that supports flexibility of interorganizational processes by transforming business networks on three different levels: local ad-hoc change, global negotiated change and planned change. Ad-hoc flexibility concerns local changes that affect only two partners. The suggested method supports this by allowing for bilateral negotiations that result in adding, removing or modifying the local business rules but leave the collaboration model intact (step M2 and part of M4). This implies a limited scope of ad-hoc changes but the speed of adaptation is high. If a higher degree of freedom is required we employ global negotiated flexibility. This is facilitated by applying the negotiation-enactment cycle of the method once (steps M2, M4 and possibly M3) for the whole network. As this is quite costly it is only applicable in situations that require massive, network-wide adaptations in response to substantial changes in the market or other environmental conditions (e.g. new legislation). The highest degree of flexibility, planned flexibility, can be achieved by rethinking the whole network. This makes sense if future developments are to be anticipated. It involves all the steps M1-M4 with possible iterations of steps M2 and M3. Especially simulation is important here to be able to assess the impact of expected changes on the network. The effort equals that of creating a new network and a careful estimate of the opportunities and risks is paramount.

At the same time the described method also supports the set-up of a new network. This is achieved by designing and enacting contracts for business networks. We interpret such contracts as transaction models that govern the interaction between network members. Contract design is divided into two phases, an initial modeling phase to develop an interaction model of the cooperation that can serve as a starting point for deriving a set of transaction models. This preliminary set of models is then refined in the second phase of negotiation. The result is a detailed contract describing the workflow of the cooperation. An intermediary phase, simulation, is subsequently employed to parameterize and optimize the process. The fourth and final phase, contract enactment, consists of the development of workflow models to manage the cooperation at an operational level.

A contract that is developed in a formal and methodical way facilitates the implementation of the procedures and the enforcement of the rules. This makes it easier to react to changes in a flexible way. The research we have performed so far shows the feasibility of the approach. A confirmation of the results in a larger field study remains to be done. In the areas of electronic commerce and virtual organizations there is a growing interest in more formalized contracts [54-56]. The existing approaches are often technology-driven and there is a need to complement them with rigorous approaches that have a strong business orientation. But formalized contracts (e.g., eContracts) are not only beneficial in electronic commerce or automated transactions. They can also make a substantial contribution towards the support of interorganizational business processes and workflows [62]. Our research takes a step in this direction.

References

[1] M. T. Hannan, J. H. Freeman, Organizational Ecology, Harvard University Press, Cambridge, MA, 1989.

[2] A. A. Alchian, H. Demsetz, Production, information costs and economic organization, *American Economic Review* 62(5): 777-795, 1972.

[3] M. C. Jensen, W. H. Meckling, Theory of the firm: Managerial behavior, agency costs and ownership structure, *Journal of Financial Economics*, 3: 305-360, 1976.

[4] S. Ross, The economic theory of agency: The principal's problem, *American Economic Review*, 63(2): 134-139, 1973.

[5] R. Wilson, The theory of syndicates, *Econometrica*, 36: 119-132, 1968.

[6] R. H. Coase, The nature of the firm, *Economica* 4: 386-405, 1937.

[7] B. Klein, R. Crawford, A. Alchian, Vertical integration, appropriable rents, and the competitive contracting process, *Journal of Law and Economics* 21: 297-326, 1978.

[8] O. E. Williamson, Markets and Hierarchies, Free Press, New York, 1975.

[9] O. E. Williamson, The modern corporation: Origins, evolution, attributes, *Journal of Economic Literature*, 19: 1537-1568, 1981.

[10] O. E. Williamson, The Economic Institutions of Capitalism, Free Press, New York, 1985.

[11] T. W. Malone, J. Yates, R. I. Benjamin, Electronic Markets and Electronic Hierarchies, *Communications of the ACM* 30(6): 484-497, 1987.

[12] E. K. Clemons, S. P. Reddi, M. C. Row, The Impact of Information Technology on the Organization of Economic Activity: The “Move to the Middle” Hypothesis, *Journal of Management Information Systems* 10(2): 9-35, 1993.

[13] C. P. Holland, A. G. Lockett, Mixed Mode Network Structures: The Strategic Use of Electronic Communication by Organizations, *Organization Science* 8(5): 475-488, 1997.

[14] P. J. Denning, R. Medina-Mora, Completing the Loops, *Interfaces* 25(3): 42-57, 1995.

[15] R. Medina-Mora, T. Winograd, R. Flores, F. Flores, The Action Workflow Approach to Workflow Management Technology. In *Proceedings of the Conference on Computer-Supported Cooperative Work CSCW’92*, J. Turner, R. Kraut (Eds.), pages 281-288, ACM, New York, 1992.

[16] T. Winograd, F. Flores, Understanding Computers and Cognition: A New Foundation for Design, Ablex, Norwood, NJ, 1986.

[17] J. L. G. Dietz, Understanding and modeling business processes with DEMO. In *Proceedings of the 18th International Conference on Conceptual Modeling ER '99*, J. Akoka, M. Bouzeghoub, I. Comyn-Wattiau, E. Métais (Eds.), pages 188-202, Springer, Berlin, 1999.

[18] J. L. G. Dietz, N. Habing, The Notion of Business Process Revisited. In *Proceedings of the OTM Confederated International Conferences, CoopIS, DOA, and ODBASE*, R. Meersman, Z. Tari (Eds.), pages 85-100, Springer, Berlin, 2004.

[19] K. Liu, L. Sun, J. Barjis, J. L. G. Dietz, Modelling dynamic behaviour of business organisations - extension of DEMO from a semiotic perspective, *Knowledge-Based Systems*, 16(2): 101-111, 2003.

[20] G. Fox, The Challenge of Convergence. In *Proceedings of the 28th Annual Hawaii International Conference on System Sciences*, pages 485-492, 1995.

[21] D. L. Dean, R. E. Orwig, D. R. Vogel, Facilitation Methods for Collaborative Modeling Tools, *Group Decision and Negotiation*, 9: 109-127, 2000.

[22] J. L. Austin, How to Do Things with Words, Oxford University Press, Oxford, 1962.

[23] J. R. Searle, Speech Acts - An Essay in the Philosophy of Language, Cambridge University Press, London, 1969.

[24] J. Habermas, The Theory of Communicative Action 1 - Reason and the Rationalization of Society, Beacon Press, Boston, 1984.

[25] E. Lehtinen, K. Lyytinen, An Action Based Model of Information Systems, *Information Systems*, 11(4): 299-317, 1986.

[26] G. Goldkuhl, Generic business frameworks and action modelling. In *Communication Modeling - The Language/Action Perspective, Proceedings of the First International Workshop on Communication Modeling*, F. Dignum, J. Dietz, E. Verharen, H. Weigand (Eds.), Springer, Berlin, 1996.

[27] G. Goldkuhl, M. Lind, Developing e-interactions – A framework for business capabilities and exchanges. In *Proceedings of the 12th European Conference on Information Systems*, 2004.

[28] G. Goldkuhl, A. Röstlinger, Joint elicitation of problems: An important aspect of change analysis. In *Human, Organizational, and Social Dimensions of Information Systems Development*, D. Avison, J. Kendall, J. Degross (Eds.), North-Holland, Amsterdam, 1993.

[29] V. E. v. Reijswoud, PhD thesis, TU Delft (Delft), 1996.

[30] F. Köhne, M. Schoop, D. Staskiewicz, A Meta Model for Electronic Negotiations - Comparison of existing approaches. In *Proceedings of the 12th Research Symposium on Emerging Electronic Markets (RSEEM 2005) "Governance of Electronic Markets"*, Y. H. Tan (Ed.), Vrije Universiteit Amsterdam, Amsterdam, 2005.

[31] M. Bichler, A Roadmap to Auction-based Negotiation Protocols for Electronic Commerce. In *Proceedings of the 33rd Hawaii International Conference on Systems Sciences*, page 6018, IEEE Computer Society Press, Los Alamitos, CA, 2000.

[32] M. Ströbel, C. Weinhardt, The Montreal Taxonomy for Electronic Negotiations, *Group Decision and Negotiation*, 12(2): 143-164, 2003.

[33] F. Dignum, U. Cortés, Agent-Mediated Electronic Commerce III. Springer, Berlin, 2001.

[34] D. K. W. Chiu, S. C. Cheung, S. Till, A Three-Layer Architecture for E-Contract Enforcement in an E-Service Environment. In *Proceedings of the 36th Hawaii International Conference on System Sciences*, page 74.1, IEEE Computer Society Press, Los Alamitos, CA, 2003.

[35] M. Ströbel, Design of Roles and Protocols for Electronic Negotiations, *Electronic Commerce Research*, 1(3): 335 - 353, 2001.

[36] M. Schoop, C. Quix, DOC.COM: a framework for effective negotiation support in electronic marketplaces, *Computer Networks*, 37(2): 153-170, 2001.

[37] A. de Moor, H. Weigand, Business Negotiation Support: Theory and Practice, *International Negotiation*, 9(1): 31-57, 2004.

[38] G. E. Kersten, S. E. Strecker, K. P. Law, Protocols for Electronic Negotiation Systems: Theoretical Foundations and Design Issues. In *E-Commerce and Web Technologies: Proceedings of the 5th International Conference, EC-Web 2004, Zaragoza, Spain, August 31-September 3, 2004*, K. Bauknecht, M. Bichler, B. Pröll (Eds.), Springer, Berlin, 2004.

[39] P. Mathieu, M.-H. Verrons, A generic model for contract negotiation. In *AISB'02 Symposium on Intelligent Agents in Virtual Markets*, The Society for the Study of Artificial Intelligence and Simulation of Behaviour, Imperial College of Science, Technology and Medicine, University of London, 2002.

[40] M. Schoop, A. Jertila, T. List, Negoisst: a negotiation support system for electronic business-to-business negotiations in e-commerce, *Data & Knowledge Engineering*, 47(3): 371-401, 2003.

[41] W. M. P. v. d. Aalst, A. H. M. t. Hofstede, B. Kiepuszewski, A. P. Barros, Workflow Patterns, *Distributed and Parallel Databases*, 14(1): 5-51, 2003.

[42] W. M. P. v. d. Aalst, A. H. M. t. Hofstede, YAWL: Yet Another Workflow Language, *Information Systems*, 30(4): 245-275, 2005.

[43] M. Bernauer, G. Kappel, G. Kramler, W. Retschitzegger, Specification of Interorganizational Workflows - A Comparison of Approaches. In *Proceedings of the 7th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2003)*, pages 30-36, 2003.

[44] G. M. Giaglis, R. J. Paul, V. Hlupic, Integrating Simulation in Organisational Design Studies, *International Journal of Information Management* 19(3): 219-236, 1999.

[45] V. Hlupic, S. Robinson, Business Process Modelling and Analysis Using Discrete-Event Simulation. In *Proceedings of 1998 the Winter Simulation Conference*, D. J. Medeiros, E. F. Watson, M. Manivannan, J. Carson (Eds.), The Society for Computer Simulation, San Diego, CA, 1998.

[46] R. J. Paul, G. M. Giaglis, V. Hlupic, Simulation of Business Processes, *American Behavioral Scientist*, 42(10): 1551-1576, 1999.

[47] R. J. Paul, A. Serrano, Simulation for Business Processes and Information Systems Design. In *Proceedings of the 2003 Winter Simulation Conference*, S. E. Chick, P. J. Sánchez, D. Ferrin, D. J. Morrice (Eds.), pages 1787-1796, Institute of Electrical and Electronics Engineers, Piscataway, NJ, 2003.

[48] R. J. Paul, A. Serrano, Collaborative Information Systems and Business Process Design Using Simulation. In *Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04)*, page 10009.1, 2004.

[49] J. H. Weyland, M. Engiles, Towards Simulation-based Business Process Management. In *Proceedings of the 2003 Winter Simulation Conference* S. E. Chick, P. J. Sánchez, D. Ferrin, D. J. Morrice (Eds.), pages 225-227, Institute of Electrical and Electronics Engineers, Piscataway, NJ, 2003.

[50] C. Chandra, A. V. Smirnov, N. Chilov, Business Process Reengineering of Supply Chain Networks Through Simulation Modeling and Analysis. In *Proceedings of the Second International Conference on Simulation, Gaming, Training and Business Process Reengineering in Operations*, pages 345-349, 2000.

[51] G. M. Giaglis, G. I. Doukidis, R. J. Paul, Simulation Assessment of Inter-Organisational Business Process Re-engineering Initiatives. In *Proceedings of the First International Conference on Simulation, Gaming, Training and Business Process Reengineering in Operations*, 1996.

[52] G. M. Giaglis, R. J. Paul, G. I. Doukidis, Simulation for Intra- and Inter-Organisational Business Process Modelling, *Informatica*, 21(4): 613-620, 1997.

[53] P. Rittgen, Deriving Simulation Models from Business Process Models, *INFOCOMP Journal*, 4(3): 23-31, 2005.

[54] S. Artyshchev, H. Weigand, Contract-Based Interoperability for E-Business Transactions. In *Interoperability of Enterprise Software and Applications*, D. Konstantas, J.-P. Bourrières, M. Léonard, N. Boudjlida (Eds.), Springer, Berlin, 2005.

[55] Z. Milosevic, P. F. Linington, S. Gibson, S. Kulkarni, J. Cole, Inter-organisational collaborations supported by E-Contracts. In *Proceedings of the IFIP 18th World Computer Congress "Building the E-Service Society: E-Commerce, E-Business, and E-Government"*, W. Lamersdorf, V. Tschammer, S. Amarger (Eds.), 2004.

[56] H. Weigand, L. Xu, Contracts in E-Commerce. In *Proceedings of the Ninth IFIP TC2/WG2.6 Working Conference on Database Semantics "Semantic Issues in E-Commerce Systems"*, R. Meersman, K. Aberer, T. S. Dillon (Eds.), 2003.

[57] P. v. Bommel, S. J. B. A. Hoppenbrouwers, H. A. E. Proper, T. P. v. d. Weide, Exploring Modelling Strategies in a Meta-modelling Context. In *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops - OTM Confederated International Workshops and Posters, AWESOMe, CAMS, COMINF, IS, KSinBIT, MIOS-CIAO, MONET, OnToContent, ORM, PerSys, OTM Academy Doctoral Consortium, RDDS, SWWS, and SebGIS, Proceedings, Part II, Montpellier, France*, *Vol. 4278* R. Meersman, Z. Tari, P. Herrero (Eds.), pages 1128-1137, Springer, Berlin, Germany, 2006.

[58] P. J. M. Frederiks, T. P. v. d. Weide, Information Modeling: the process and the required competencies of its participants, *Data & Knowledge Engineering*, 58(1): 4-20, 2006.

[59] S. J. B. A. Hoppenbrouwers, L. Lindeman, H. A. Proper, Capturing Modeling Processes - Towards the MoDial Modeling Laboratory. In *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops - OTM Confederated International Workshops and Posters, AWESOMe, CAMS, COMINF, IS, KSinBIT, MIOS-CIAO, MONET, OnToContent, ORM, PerSys, OTM Academy Doctoral Consortium, RDDS, SWWS, and SebGIS, Proceedings, Part II, Montpellier, France*, *Vol. 4278* R. Meersman, Z. Tari, P. Herrero (Eds.), pages 1242-1252, Springer, Berlin, Germany, 2006.

[60] S. J. B. A. Hoppenbrouwers, H. A. Proper, T. P. v. d. Weide, Formal Modelling as a Grounded Conversation. In *Proceedings of the 10th International Working Conference on the Language Action Perspective on Communication Modelling (LAP`05), Kiruna, Sweden*, G. Goldkuhl, M. Lind, S. Haraldson (Eds.), pages 139-155, Linköpings Universitet and Högskolan i Borås, Linköping and Borås, 2005.

[61] A. Persson, Enterprise Modelling in Practice: Situational Factors and their Influence on Adopting a Participative Approach. Department of Computer and Systems Sciences, Stockholm University, 2001.

[62] H. Weigand, W. J. A. M. van den Heuvel, Cross-organizational workflow integration using contracts, *Decision Support Systems*, 33(3): 247-265, 2002.

[63] M. Shen, D.-R. Liu, Coordinating Interorganizational Workflows based on Process-Views. In *Proceedings of the 12th International Conference on Database and Expert Systems Applications (DEXA'01)*, *Vol. 2113*, pages 274-283, Springer, Berlin, 2001.