A Contract-Based Architecture for Business Networks

Peter Rittgen

ABSTRACT: Contracts coordinate the interactions between organizations in a business network. Conventional frame contracts often result in ambiguous agreements with operational details unspecified, making it hard to coordinate the network effectively and efficiently. This paper suggests a structured method for designing agreements and ensuring that their terms are observed in business transactions. Based on the assumption that governing a business network means managing the workflow between the member organizations, the contract consists primarily of process models at different levels of detail. A method is proposed for negotiating these models between the member organizations of the network and enacting them with the help of an interorganizational workflow system.

KEY WORDS AND PHRASES: Business networks, e-contracting, enactment, frame contract, interorganizational workflow, modeling, negotiation.

There are currently two conflicting trends in business cooperation. First, competitive pressures force companies to concentrate on their core competencies and outsource other activities. Second, customers demand complementing products and services from one source. The latter point seems to suggest an increased amount of "insourcing." The solution in both cases if for companies to cooperate more closely, each concentrating on its area of expertise, but jointly offering a complete suite of related products and services that are well matched (one face to the customer). But this scenario represents an enormous challenge both in terms of organization and regarding the information system support.

In a business network, organizations strive to provide complex products and services by coordinating their activities in an "intelligent" way. The coordination effort is much higher than in a conventional supply chain. In the latter, an individual company can focus on managing its relationship to a few immediate major suppliers for creating a product or service, whereas in a business network this is not enough, since coordination among the suppliers is also required. Theoretically we move from a tree structure to a graph topology, which implies that we have to strike a new balance between market and hierarchical coordination. The underlying general problem is quite old, and several theories have been advanced to explain the use of various forms of coordination, most notably agency theory [4, 40, 69, 88] and transaction cost economics [16, 44, 85, 86, 87]. These theories can be used to determine the internal and external coordination costs [33]. High external costs favor centralization, high internal costs promote decentralization. It is typically assumed that organizations in a supply chain choose their organizational structure and network of trading partners in such a way that the sum of both costs is minimized. There has also been some debate on the impact of information technology (IT). Early work by Malone, Yates, and Benjamin suggested that IT will lower transaction costs and therefore, *ceteris paribus*, lead to an increase in

International Journal of Electronic Commerce / Summer 2008, Vol. 12, No. 4, pp. 119–150.

1086-4415/2008 $9.50 + 0.00.
DOI 10.2753/JEC1086-4415120405

market coordination [49]. Later work posited that organizations will "move to the middle"—that is, to "more outsourcing, but from a reduced set of stable partnerships" [15]—if non-contractible issues (e.g., quality and trust) play an important role. Empirical evidence shows that companies often operate in a "mixed mode" blending aspects from both markets and hierarchies [36].

But most 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, the contract is typically used as an instrument for coordination. Agency theory suggests that there are two principal forms of contracts, one behavior-based and the other outcome-based. If the costs for monitoring agent behavior are high, an outcome-based contract is favored. This is because an unobserved agent, knowing there are no consequences to fear (moral hazard), is assumed to shirk (i.e., underperform). Thanks to the omnipresence of information technology, the costs for monitoring agent behavior have become marginal in many cases. In consequence, the discussion here focuses on behavior-based contracts incorporating models of the interorganizational process.

The overall method is divided into three phases (*see Figure 1*):

1. Negotiating the frame contract as a model of the interaction between the organizations.
2. Simulating the process models to determine process parameters and improve process performance.
3. Deriving the workflow models (collaboration models) that govern interaction in the business network.

These phases are described very briefly in the following paragraphs and more thoroughly in the subsequent sections.

A behavior-based approach requires a model of the interactions between the organizations. A significant number of interactions serve to coordinate productive activities. Coordinative actions are communicative actions and require a communicative perspective (language-action perspective) on the process. Dynamic essential modeling of organizations (DEMO) is a suitable method for this task. Its interaction model describes how organizations interact with each other on the transactional level. This forms the basis for the development of more detailed models of interaction at the elementary business-act level (collaboration models).

The process of modeling is interpreted as a negotiation process with the aim of reaching a consensus model among the participants. Empirical evidence shows that models of suitable quality can be negotiated efficiently using either the traditional chauffeured approach or groups of subject matter experts supported by a modeling expert [17, 24]. The negotiators are representatives from each member of the business network. Negotiation is supported 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.

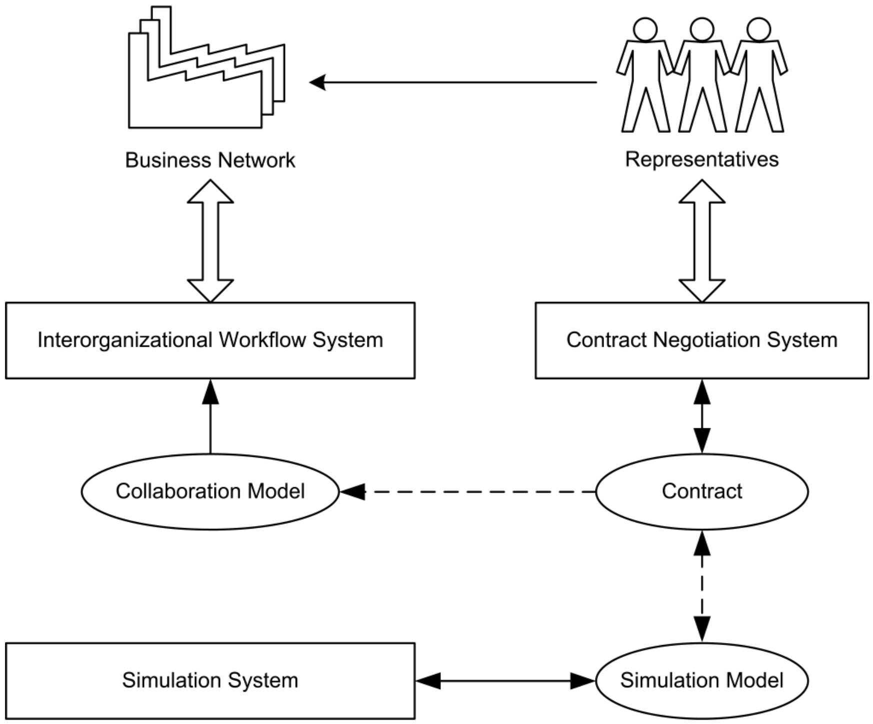

Figure 1. A Method for Negotiating and Enacting a Contract

As will be shown further on, simulation can be used to identify process parameters and improve process performance. In order to do so it is necessary to derive a simulation model based on the agreed collaboration models. The results from simulation can help to determine certain parameters, such as unit prices. They can also provide feedback on how to change the models to make the process more efficient by "playing" with alternative process organizations.

Enacting the contract involves the development of a collaboration model that can be used as a template to control a workflow engine. This makes it possible to manage the interactions between the network members on the operational level.

All the examples and figures used in this paper are excerpts from real models designed in the course of a consulting project that tested the feasibility of the proposed approach. The case study gives further details on this project.

Modeling the Interorganizational Process

At the core of the language-action perspective is Austin and Searle's speech act theory [7, 74]. Its central premise is that language is not only a medium for exchanging information but a means of action. Uttering something is actually doing something. We can instruct, direct, request, make commitments, promise, apologize, propose marriage, and the like, all by just saying a few words. Every language action consists of an illocution describing the kind of action and a propositional content referring to the object of the action. Habermas embedded this theory of speech acts into a social context whereby language action becomes social action [34]. In his theory of communicative action, he argued that every action is determined by the roles the actors play and their (power) relations toward one other. He suggested that an actor performing a language action is making a validity claim, whether implicitly or explicitly. An order, for example, makes a claim to authority. If the actor issuing the order has no authority over the addressee, the latter is under no obligation to obey, and thus the order becomes meaningless.

In an organizational setting, communication is often aimed at the performance of a specific action to achieve an objective. Templates for such goal-driven conversations, such as the conversation-for-action schema and the action-workflow loop [19, 52, 89], provide a stable framework for the analysis of organizations in general and business processes in particular. Dynamic essential modeling of organizations (DEMO) [20, 21, 48], action-based modeling [47], business action theory, and SIMM [29, 30, 31] are all more sophisticated examples of such frameworks.

This work shows that organizational behavior is deeply rooted in language action. All coordination is essentially communicative. Language is what enables us to build and maintain organizations. We use it to delegate, report, inform, negotiate, sanction, hire, train, and so on. The importance of communication is even more obvious in an interorganizational context where we cannot rely on a common structure when coordinating activities that cross the boundaries between organizations. Thus a modeling approach that addresses communicative action is needed.

DEMO

The language-action perspective offers many approaches, some of which have already been mentioned. DEMO has been chosen for the purposes of the present discussion because it offers transactional patterns not only in the meta-language but also as concepts in the modeling language. This makes it possible to distinguish between transactions (as complex communicative actions) and business-acts (as elementary actions), which is essential for the proposed approach. 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.* Transactions are assumed to follow a certain pattern that is divided into three sequential phases and three layers: success, discussion and discourse. On the success layer the phases are *order* (O), *execute* (E), and *result* (R). If anything goes wrong on the success layer, the participants can decide to move to the discussion or discourse layer. For details on these layers, see Reijswoud's paper [65].

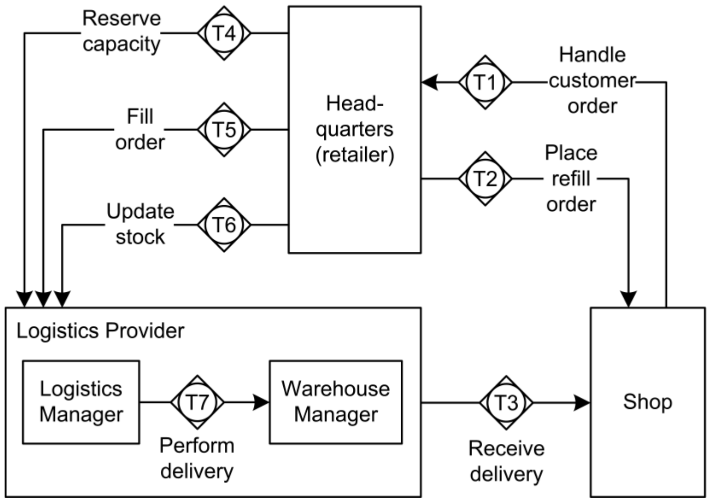

Figure 2. Interaction Model

Interaction Model

The interaction model shows actors and transactions. The actors are roles enacted by a person, an organizational unit or a whole organization. Figure 2 shows the interaction model of our case. The main actors are the Logistics Provider, the Headquarters of the retailer, and the Shop.

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. The name of the transaction, which coincides with that of the objective action, is also shown. This was included to enhance readability; it is not a feature of the original DEMO language. Figure 2 describes the process of capacity reservation and order handling among these organizations. It starts when Headquarters reserves capacity for handling a certain amount of ordered items six months, two months, and two weeks in advance of the actual order (T4). These reservations represent forecasts whose accuracy increases the closer they are to the date of delivery. The logistics company (LogCom) allocates staff and space so that the reserved capacity can be provided when the order arrives. But the capacity required by the order may actually be higher or lower than what is reserved.

The product assortment consists of basic-range products and seasonal products. The latter are distributed according to turnover quota and are not part of the order process. Orders for basic-range products 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 this case suggest that the Shop place a refill order (T2). For this purpose it sends an order proposal containing the products in question, which is returned after any necessary changes or additions are made. 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). The delivery to the Shop will then be performed by LogCom (internal transaction T7), which includes picking items, packing them, and handing them over to the carrier. The actual delivery is largely noncommunicative and material, so it is not explicit in the Interaction Model. Only the coordinative part of it is represented—namely, the Shop receiving the delivery (T3). This consists of the arrival of the goods and a confirmation. The arrival of the goods is a material action that also has a communicative function: Through it LogCom states that it has performed the delivery and thereby fulfilled its obligation. The confirmation can be accompanied by a complaint if items are missing or the wrong ones have been sent.

Periodically Headquarters will also ask for an update of the stock (T6). This is necessary because it runs its own "virtual" warehouse management system which is not integrated with the "physical" warehouse management system of LogCom. As will be described below, transaction models can be derived from the interaction model. This is an intermediate step toward the two dynamic parts that make up the contract—the business rules and the collaboration model.

Collaboration Models

Much of the detailed behavior that constitutes a business process is hidden inside every transaction. This has to be brought to light for the specification of the contract, because it constitutes the content of the business rules and the collaboration model. A transaction in DEMO is made up of a number of speech acts and an objective action that follow a certain pattern. The pattern is not a rigid template that fits every transaction. It is rather a guideline that describes a common conversational structure that can help in analyzing a particular situation. In some cases it will describe the situation fairly accurately, in others it might be necessary to revise it or even to develop a new one that is specific to the particular situation.

The pattern consists of the phases mentioned above: order, execute, and result. The actagenic conversation (O phase) has at least two elements: a *request* and a *promise* (*see Figure 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 factagenic conversation (R phase) is entered. At a minimum this can consist of the speech acts *state* and *accept.* Figure 3 summarizes these steps, which are performed in the order indicated by the leading numbers. The same notation is used for the actors as in the interaction model. A speech act is represented by a circle containing the number of the respective transaction. An arrow goes from the performer via the circle to the addressee. The performer is the one who makes the utterance, the addressee is the "listener." The arrow is annotated by the name of the speech act, which can be preceded by a sequence number. An objective action is represented by a diamond containing the number of the respective transition. The arrow starts and ends at the executor. A model that contains only actors, speech acts, and objective actions is called a collaboration model.

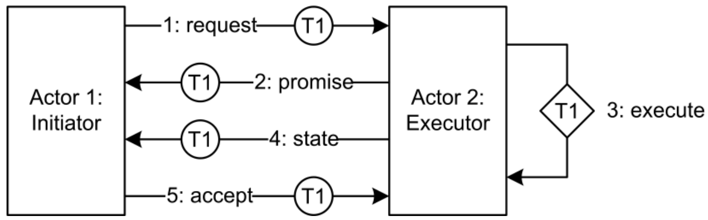

Figure 3. Collaboration Model of a Transaction.

Modeling as a Negotiation Process

Given the abundance of the literature on modeling, one would expect that the process of modeling—that is, how models are actually created—would be well understood. And indeed, methods for developing, such as business process models, can be found with ease [5, 9, 12, 31, 53, 60, 68, 71, 84]. Most of these methods operate on a coarse-grained level by specifying, for example, the order in which diagrams should be developed. Some also provide guidelines on how to create a specific diagram, especially in object-oriented modeling [10]. But in practice the use of a method is often reduced to the use of its notation, as the former is fraught with problems [39]. Therefore it is necessary to study the modeling process from a descriptive rather than a prescriptive perspective to find out what actually happens when people model and to support these activities with appropriate tools.

There are very few descriptive approaches to understanding the modeling process [56, 76, 82, 83]. Most of them are situated either in management science or in information systems, where a single expert modeler creates a formal model of some part of a business. The studies identify sets of general heuristics for successful modeling without going down to the level of the concrete steps performed in creating models. The results are hardly applicable to general business modeling for a number of reasons. First, a business model is rarely developed by one expert working alone but instead is produced by a team made up of representatives of the respective business(es) and externals. Second, the problem domain of general business modeling is often less well structured and formal languages are of limited use. Third and last, the goal of providing tool support for collaborative modeling requires the identification of detailed steps.

To understand the modeling process, it was necessary to assume that two factors are predominant in model creation: the internal reasonings of each modeler, and the conversations between modelers and within the group. A think-aloud process-tracing methodology was used to get access to the former [22, 76], where the observers speak out what they are currently thinking. A preliminary coarse-grained categorization was developed based upon theories in the pertinent literature, particularly in organizational semiotics. The upper four "rungs" of the semiotic ladder were used—syntactic, semantic, pragmatic, and social—referring, respectively, to the structure of a sign system (e.g., a language), the meaning of the signs, their use, and the norms of a community [77]. An initial coding phase within this framework revealed that the syntactic

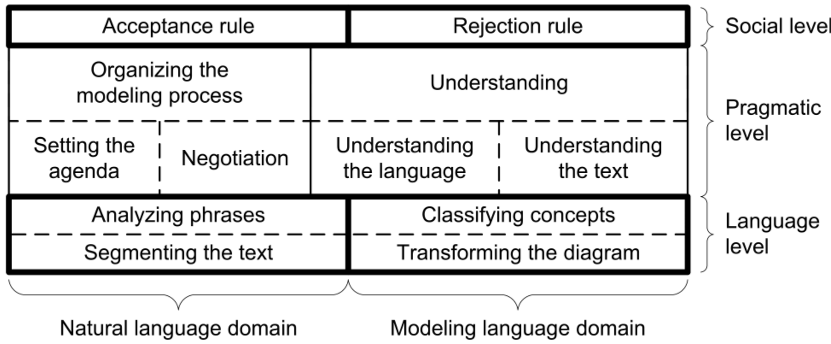

Figure 4. Levels and Domains

and semantic levels, which together make up the language level, are divided into the natural language domain and the modeling language domain depending on the kind of language used to describe the business.

Activities on the pragmatic level were classified as either understanding or organizing the modeling process. The former term was further refined into ″understanding the language″ and ″understanding the text.“ The latter can be divided into ″setting the agenda″ and ″negotiation.“ The social level consists of rules for acceptance and rejection in the negotiation. A detailed discussion of these categories can be found in the respective sections. The results are summarized in Figure 4.

Three experiments were conducted involving a total of 26 groups of two or three informatics students each over a period of three years. The students were provided with a textual description of four business processes in a hospital and were asked to model these processes with the help of two different modeling languages chosen freely from a set of four languages: ARIS-EPC [71], FMC-Petri nets [42], UML [58, 59], and DEMO [20]. An architecture of a system that can support the modeling process was derived based on the results of these experiments.

The main coding of the material was carried out within the framework stipulated by Figure 4. Examples of the procedure are shown further on. The results are presented here in the order of the levels from top to bottom.

Social Level

The social norms within a modeling team are mainly made up of rules for determining whether a proposal is accepted or rejected. The rules do not have to be logical complements that allow for situations where a proposal can be neither rejected nor accepted but requires further convincing to decide one way or the other. A termination rule was occasionally applied to force a decision if a negotiation got stuck, that is, when there were no more changes in the individuals' convictions over an extended period. Two types of rules were witnessed:

- *Rules of majority,* where a certain number of group members (e.g., more than half) had to support or oppose a proposal in order for the whole group to accept or reject it. A tie-break rule was sometimes specified (e.g., for the case of an equal number of supporters and opponents). The tie-break could involve seniority issues.
- *Rules of seniority,* where the weight of a group member's support or opposition was related to the member's status within the group. This status could be acquired (e.g., by experience) or associated with a position to which the member was appointed. For example, very frequently the other members regarded a more experienced modeler as their leader; this member made decisions on behalf of the others, who in such cases filled the role of consultants.

The rules were sometimes set up explicitly before the group began its work or in an early phase of the work. In most cases, however, they emerged as the result of each member's behavior. Individuals making regular contributions of high quality were likely to acquire seniority. Homogeneous teams more often used majority rules.

Pragmatic Level

On the pragmatic level two distinct types of behavior were discovered, each of which can be classified in two sub-categories (the abbreviations of the categories are used as indices of the respective coded terms later on):

- *Understanding,* which concerns the text of the case description (index UT) or the (modeling) language (index UL).
- Organizing the modeling process, which involves two types of activities: *setting the agenda* (index SA) and *negotiation* (index N).

Understanding was established by questions and answers. If the respondent could not provide clarification, an assumption was made. For details see Table 1.

The participants in the study used agendas as an instrument for roughly structuring the modeling session. These were introduced at the beginning and adapted during the session if necessary. On the whole, most groups started by reading the case description completely and then organized their work around the flow of the text. For further details refer to Table 1.

The majority of the activities on the pragmatic level were associated with negotiation, though (*see Table 1*). This was surprising, because modeling is typically pictured as an intuitive act largely the product of a creative brain (e.g., a consultant) that may receive some input from other stakeholders in the modeling process (e.g., domain experts from the respective departments).

Based on the foregoing results, it is possible to draw interesting conclusions for the design of a system that supports modeling, as will be treated further below. This concludes the discussion of the pragmatic level. The next section proceeds with the semantic level.

Activity	Coding
Modeler *m* makes proposal *p*.	**propose**$_N$ (*m*, *p*)
Modeler *m* withdraws proposal *p*.	**withdraw**$_N$ (*m*, *p*)
Modeler *m* consents to proposal *p*.	**support**$_N$ (*m*, *p*)
Modeler *m* objects to proposal *p*.	**challenge**$_N$ (*m*, *p*)
Modeler *m* delivers argument *a* to support *p*.	**argue_for**$_N$ (*m*, *p*, *a*)
Modeler *m* delivers argument *a* to challenge *p*.	**argue_against**$_N$ (*m*, *p*, *a*)
Modeler *m* proposes *p*′ instead of *p*.	**counter**$_N$ (*m*, *p*, *p*′)
Modeler *m* needs clarification on issue *q*.	**ask**$_{UT/UL}$ (*m*, *q*)
Modeler *m* provides possible answer *a* to question *q*.	**assume**$_{UT/UL}$ (*m*, *q*, *a*)
Modeler *m* gives definite answer *a* to question *q*.	**clarify**$_{UT/UL}$ (*m*, *q*, *a*)
Add activity *a* to agenda as item *n*.	**add**$_{SA}$ (*a*, *n*)
Perform next activity on agenda.	**perform**$_{SA}$

Table 1. Generic Activities on the Pragmatic Level.

Semantic Level

The semantic level is concerned with the concepts of the modeled domain—in the present case, business interaction. It is also called the conceptual level. The concepts are often closely linked to the ones found in the language used for modeling the domain. The modeler expresses the perceived or constructed reality in terms of the concepts provided by the language, whether a natural language or a modeling language. This implies that the chosen language both enables and restricts the modeler in respect to having and expressing certain thoughts. Of the four languages of the study, DEMO was the one targeted on business interaction. The results of coding the modeling activities associated with this language and the types of concepts identified in this domain (= *Classifying Concepts,* index *CC*) are shown here. The foundational concepts are: actor, action, transaction, and business act. The relational concepts between them are listed in Table 2 without the surrounding **classify** activity, for convenience. By phrase is meant a text fragment (e.g., from the case description). The index *AP* refers to *Analyzing Phrases,* the main activity of the natural language domain).

This completes the activities on the semantic level. The next section deals with the syntactic level.

Syntactic Level

On the syntactic level a distinction is made between the natural language and modeling language domains. In the latter the diagrams of the respective language are built (*Transforming Diagrams, TD*). They consist of nodes, edges that connect them, and labels attached to both. In general, the foundational concepts are represented by nodes and the relational concepts by edges, but this is not necessarily true for all languages (e.g., Petri nets). In the natural language domain, the text is segmented into useful units for the analysis on

Activity/Concept	Coding
Phrase p_1 is considered equivalent to phrase p_2.	**interpret**$_{AP}$ (p_1, p_2)
Phrase *p* is considered an instance of concept *c*.	**classify**$_{CC}$ (*p*, *c*)
Actor *x* performs objective action *a*.	**execute**$_{CC}$ (*x*, *a*)
Initiator *i* starts transaction *tr* with executor *e*.	**transact**$_{CC}$ (*i*, *tr*, *e*)
Locutor *l* directs communicative action *c* to addressee *d*.	**perform**$_{CC}$ (*l*, *c*, *d*)
Communicative action *c* is an utterance with illocution *il* and propositional content *pc*. Possible illocutions are request, promise, state, and accept.	**utter**$_{CC}$ (*il*, *pc*)
Transaction *tr* is decomposed into order phase *o*, execution phase *e*, and result phase *r*.	**split**$_{CC}$ (*tr*, *o*, *e*, *r*)
A party commits to the execution of *a* in time interval *t* (after *a* unilateral request or a promise).	**commit**$_{CC}$ (*a*, *t*)
Execution of *a* in time interval *t* is requested.	**request**$_{CC}$ (*a*, *t*)
Execution of *a* in time interval *t* is promised.	**promise**$_{CC}$ (*a*, *t*)
Execution of *a* in time interval *t* is stated.	**state**$_{CC}$ (*a*, *t*)
Execution of *a* in time interval *t* is accepted.	**accepted**$_{CC}$ (*a*, *t*)

Table 2. Generic Activities and Concepts on the Semantic Level.

the semantic level (*Segmenting the Text, ST*). The generic activities on this level are listed in Table 3.

Architecture of a Modeling Support System

Analysis of the modeling sessions showed that modeling is a relatively well structured process. Thus it is worthwhile to consider a system that can support this process. Modeling consists of a limited number of well-defined activities on all levels of the semiotic ladder. Further research will certainly reveal more activities, but from the experience of the three experiments that yielded a decreasing number of new ones, it seems likely that the total number of activities will converge with respect to a given domain (e.g., business interaction). The activities identified so far can therefore be assumed to be stable in that domain. But the findings will not carry over to another domain because of the domain specificity of the language level. The other levels are likely to work, though.

Analysis of the workflows on the pragmatic level revealed a structure that goes beyond the mere identification of generic activities. The negotiation process was found to actually follow a certain pattern, as shown in Figure 5.

The pattern consists of an initial and reject state at the top, a state where acceptance is favored (upper left corner), a state where rejection is favored (upper right corner), a recursive substate for negotiating a counter-proposal (lower right corner), and an accept state (lower left corner). Each of the states allows for a set of pragmatic activities that take the negotiation to a different state. The parameters concerning the modeler who performs the activity and the argument (if present) are omitted. In general, any modeler can perform any activity, but there are a few rules to be observed. A modeler making a proposal is implicitly assumed to support it and is the only one who may withdraw it. A different modeler can bring up a counter-argument, but the

Activity	**Coding**
Text fragment *t* is the unit of analysis. The fragment can be: text, section, sentence, nominal phrase, verbal phrase, or word.	$\textbf{focus}_{ST}$ (*t*)
Introduce a new node *n* of type *t*.	$\textbf{introduce}_{TD}$ (*n*, *t*)
Attach label *l* to node *n*.	$\textbf{label}_{TD}$ (*n*, *l*)
Remove node *n*.	$\textbf{remove}_{TD}$ (*n*)
Connect node n_1 to node n_2 with edge *e* of type *t*.	$\textbf{connect}_{TD}$ (n_1, n_2, *e*, *t*)
Attach label *l* to edge *e* at place *p*.	$\textbf{label}_{TD}$ (*e*, *l*, *p*)
Remove edge *e*.	$\textbf{remove}_{TD}$ (*e*)

Table 3. Generic Activities on the Syntactic Level.

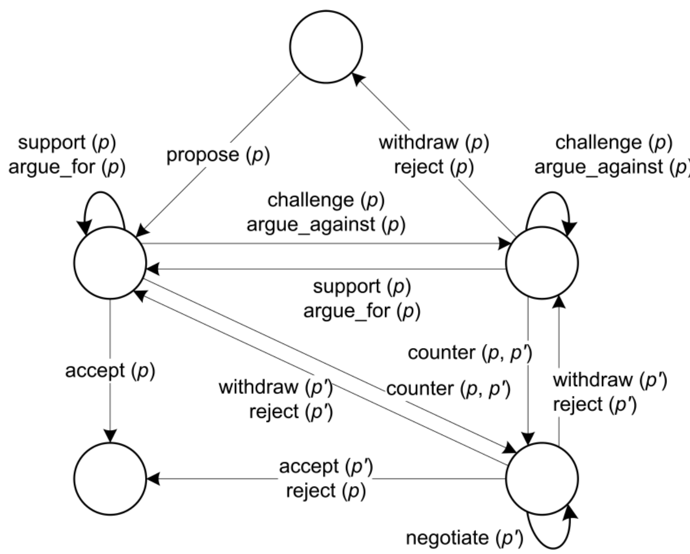

Figure 5. Negotiation Pattern

proponent of the original proposal can also make a counter-proposal (e.g., to accommodate counter-arguments).

The negotiation component of a modeling support system can be controlled with the help of the pattern in Figure 5. An equally strong pattern of activities was not discovered on the other levels. This will affect the kind of support a tool can provide at the language level.

Figure 6 shows a potential architecture of a modeling support system that makes use of existing technology as far as possible. At the bottom there are a text-analysis tool and a modeling tool. The former is used to structure the text (e.g., a case description) to prepare it for phrase analysis and concept classification. Available tools for text analysis can be adapted to this purpose, among them the RST-tool [50, 51, 57] or NoDoSE [3] or tools for qualitative data analysis [23]. The text-analysis tool must have an interface to the modeling tool so that a link can be established between phrases of the text and the concepts of the modeling language.

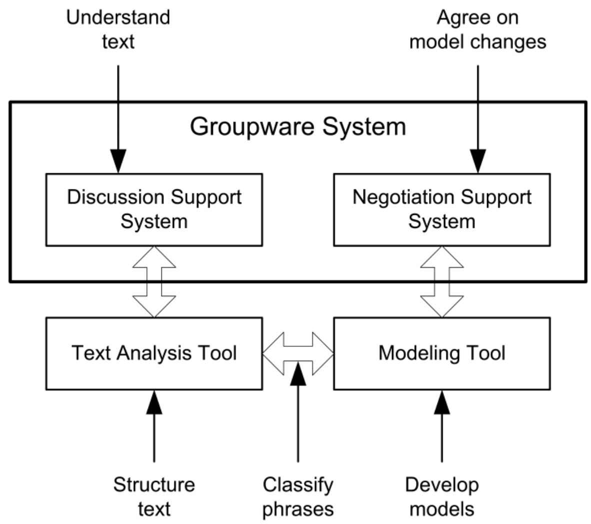

Figure 6. Architecture of a Modeling Support System

The modeling tool provides functionality for creating, checking, and managing models in a particular language. Existing tools for business process modeling (e.g., ARIS, ADONIS, Enterprise Architect, Pro*Vision* Enterprise, MEGA Process, casewise Corporate Modeler, iGrafx Process, WebSphere Business Modeler) can be utilized if appropriate interfaces to the other components can be furnished.

A groupware system placed on top of these tools deals with the pragmatic and social levels [8]. This system needs two components, a discussion-support system and a negotiation-support system (NSS). Existing groupware systems already provide most of the infrastructure required for supporting discussions (e.g., functions for brainstorming, prioritizing suggestions). The purpose of the discussion component is to help the group to understand the text by discussing unclear or ambiguous phrases. Therefore it needs to have an interface to the text-analysis tool so that the group members can make annotations to the respective phrases and adapt the structure of the text accordingly.

The second component of the groupware system is the NSS. It is part of the groupware system because the negotiation is done in the group and can be seen as a special kind of discussion. The purpose of the NSS is to make sure that the group approves every change to the model. Therefore it needs a close link to the modeling tool. An existing NSS might be used, but it will probably need adaptation because such systems are usually tailored for different kinds of negotiations (e.g., sales contracts for pre-defined goods) [18, 72, 73].

Negotiation in the Business Network

The left part of Figure 1 shows how the operation of a business network is supported. It assumes 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 for a new business network). On the other hand, negotiation can also be about business network maintenance (adapting to the loss of members, incorporating new members, replacing departing members, reacting to changed requirements, etc.). The contract under consideration will in any case be subject to implementation, which yields a description of the interactions between the members in some workflow language. The choice of this language depends on the workflow system selected to coordinate the workflow between members. In principle, any workflow system can be used that allows for the implementation of the workflow patterns identified in [2]. Most commercial systems qualify if one allows for workarounds and coding, but there is little native support for many of the advanced patterns. In a prototypical environment, it can be useful to employ YAWL [1], which provides all but one pattern, together with the YAWL Engine. YAWL also makes use of XQuery and XPath to extract data from XML input files and for generating XML output. This supports 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 example based on a YAWL implementation is given further on, under the heading "Managing the Interorganizational Workflow." An overview and comparison of other languages for interorganizational workflows can be found in [11].

From Negotiation to Enactment

The preceding sections describe the general method of governing a business network. This section describes how the procedures in the method are performed and what the results look like. For this purpose the discussion considers a simple negotiation, the corresponding part of the contract, and the resulting workflow net (enactment) in some detail. The example represents only a very small part of the case and just serves to illustrate the way the proposed approach works. The complete example is shown in the next section on a more general level.

The 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 wants LogCom to take over the delivery of orders. Figure 7 shows two steps in the negotiation between them. The representative from LogCom writes an e-mail saying that it needs a capacity reservation two weeks in advance of the order to be able to handle it. The negotiation-support system translates this request from natural language to the internal, formal representation:

```
REQUEST (Reserve_capacity[ORDER], t ≤ DATE[ORDER]-14)
```

The keyword REQUEST indicates that LogCom would like to introduce a new action into the cooperation. The propositional content of this message tells us what that action is—namely, the reservation of capacity for each order. The

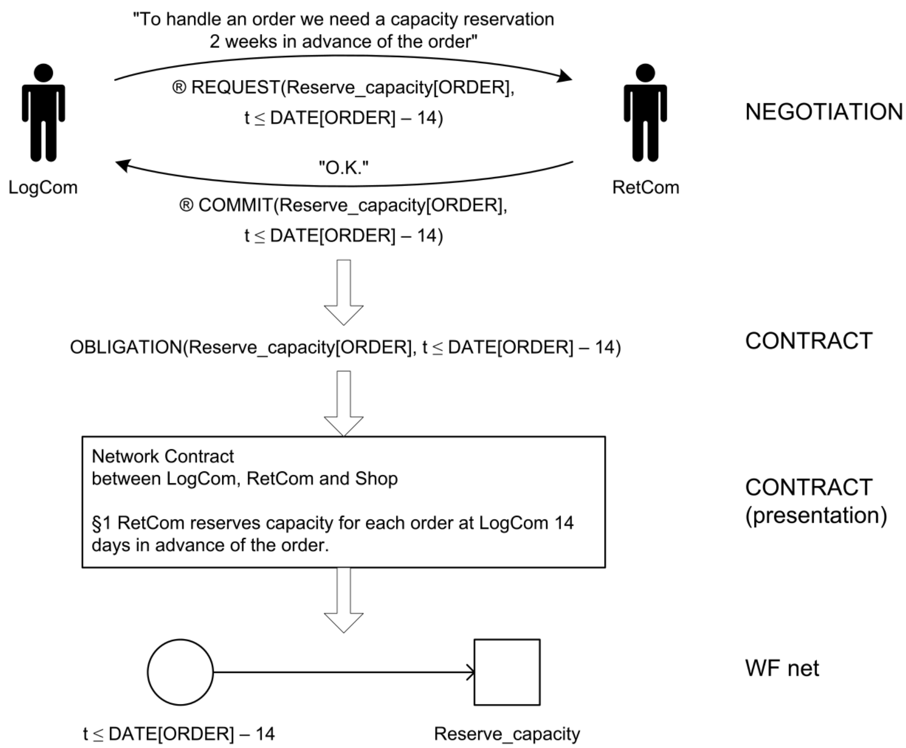

Figure 7. From Negotiation to Workflow Net (example)

request also specifies a time restriction for this action— 14 days in advance of the order date. The message is stored in the message memory of the negotiation system so that it can be matched with RetCom's reaction. In this case RetCom fully agrees with the action suggested by LogCom by answering "O.K.." Again the NSS will translate this to the formal representation:

```
COMMIT (Reserve_capacity[ORDER], t ≤ DATE[ORDER]-14)
```

The speech act COMMIT signals that RetCom agrees 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 to the requesting party with respect to the content. An alternative reaction of RetCom could be:

```
COMMIT (Reserve_capacity[ORDER], t ≤ DATE[ORDER]-7)
```

This 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 the example, the original request is granted and the appropriate obligation is inserted into the contract:

```
OBLIGATION (Reserve_capacity[ORDER], t ≤ DATE[ORDER]-14)
```

At any point in time, the presentation component of the NSS can display the contract that has been negotiated so far in human-readable form (*see Figure 7*). In the final step, the obligation is translated to a corresponding workflow.

Simulating the Process

A contract has to specify general terms and conditions that can be seen as parameters controlling the interaction between the trading partners. Pricing, terms of delivery, and terms of payment are all examples of such parameters. Determining reasonable values for these parameters is difficult because they depend on characteristics of the interorganizational process. This is particularly true when companies engage in a cooperation for the first time and therefore lack prior experience. Consider, for instance, the pricing of a logistics service (e.g., handling one unit of the customer's product). How much is charged depends, among other things, on how much it costs to deliver the requested service, which in turn depends on the time it takes, the number of workers involved, the resources used, and so on. One way of assessing the complex interaction of these factors is to simulate the business process. The usefulness of simulation has been studied thoroughly [28, 35, 62, 63, 81], particularly in an interorganizational context [14, 26, 27].

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 to observe—

1. the potential behavior of a system that does not (yet) exist, and
2. the (potential) behavior of an existing system at a much faster pace and lower cost 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 of the imagined system if it should be built). This makes it possible to determine the performance characteristics of the business process. These data can then be used to support the design of the terms and conditions of the contract. The development of the simulation model use the approach described in [66]. It is based on a language-action model of the business process and proceeds in three steps:

1. Designing the business process view
2. Designing the resource view
3. 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. The second step develops a view that tells us which resources are required by

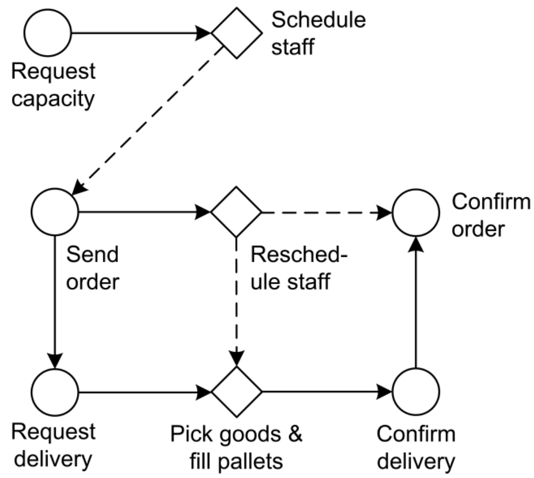

Figure 8. The Process View of the Interaction Model

each action. The final step results in the simulation model, which is written in SimPy (Simulation in Python). To give an idea of how this works, steps 1 and 2 are shown in detail in the following sections. The pricing of product handling is used as an example because it was the issue given highest priority by both corporate partners in our project. Based on the assumption that the costs of handling a product unit will play an important role in determining the price, a closer look was taken at transaction T7, Perform delivery, and the associated transactions T4, Reserve capacity, and T5, Fill order (*see Figure 2*).

Decomposing the transactions into speech acts and productive actions gives the process view shown in Figure 8. The business process starts with a request for a certain capacity which is then used to schedule the warehouse staff accordingly (target action). Note that the promise part of the actagenic conversation is omitted because the business rules force LogCom to accept each request (as will be discussed in the next section). After that the availability of the capacity is confirmed (speech act: state). Due to the business rules in the contract (see the next section) the speech acts "promise" and "accept" can be omitted in this and later transactions. The next step in the process is the order sent by the customer. Observe that this action is not caused by the confirmation of the capacity because the customer may decide not to make use of the capacity and not send an order. Hence there is no causal relation. On the other hand, the order cannot be sent without prior reservation of capacity, which makes the relation conditional (dashed arrow).

The information in the order is then used to reschedule the staff depending on the actual package load. This might involve requiring the outbound staff to work extra hours or reassigning the inbound staff. At the same time, the delivery is requested, but it cannot take place before the staff for this task is rescheduled so that sufficient staff is available. The goods are then picked from the shelves and loaded on the pallets filled with goods destined for that particular shop. Once this has been done, the pallets are picked up by the forwarder and the delivery is confirmed. This also makes it possible to confirm the completion of the whole order.

The process view approaches the more conventional activity-based view of a business process, but there is still a major difference. The latter only contains objective actions— activities performed in the real world to achieve some business goal—whereas the former also includes communicative actions performed in the social world for coordinative purposes. One way to get a true activity-based view would be to abstract from the communicative actions by formally considering them to be empty actions and removing them from the model. Another way would be to transform the whole business model from the language-action perspective to the activity perspective. This approach is described in [67].

The work described so far was part of the initial business analysis that also identified problems and goals. Since one of the most pressing problems from the standpoint of LogCom pertained to the discrepancies between planned and actual capacities, a simulation of the relevant parts of the overall process was done to determine how the deviations affected transaction costs. To do so it was necessary to complement the business process view with a resource view (step 2) to get a clearer picture of the use of resources by the actions in the process. The result is shown in Figure 9.

The resource view shows the actors and objects involved in each action. It is assumed that an actor engaged in an action cannot perform another action at the same time. Most of the information in Figure 9 can be derived from the action and process views (Figures 2 and 8, 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 both 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, the initiator is dropped and only the executor is recorded as a resource.

In the example of Figure 9, this procedure yields an almost complete diagram with only three resources missing. These concern the action "Pick goods and fill pallets," which 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 (including extra staff), the overtime, and the time required for handling a unit. The latter is assumed to be normally distributed with given μ and σ. 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 σ. All other actions are assumed to require a negligible time.

Several runs of the resulting simulation model gave the average values of reserved units, requested units, deviation, number of regular and extra staff, overtime, handled units, total costs, and unit costs for each of the 52 weeks of a year (*see Table 4*). These results were used, among other things, to make the constant pricing model of the old frame contract variable. This implies that the

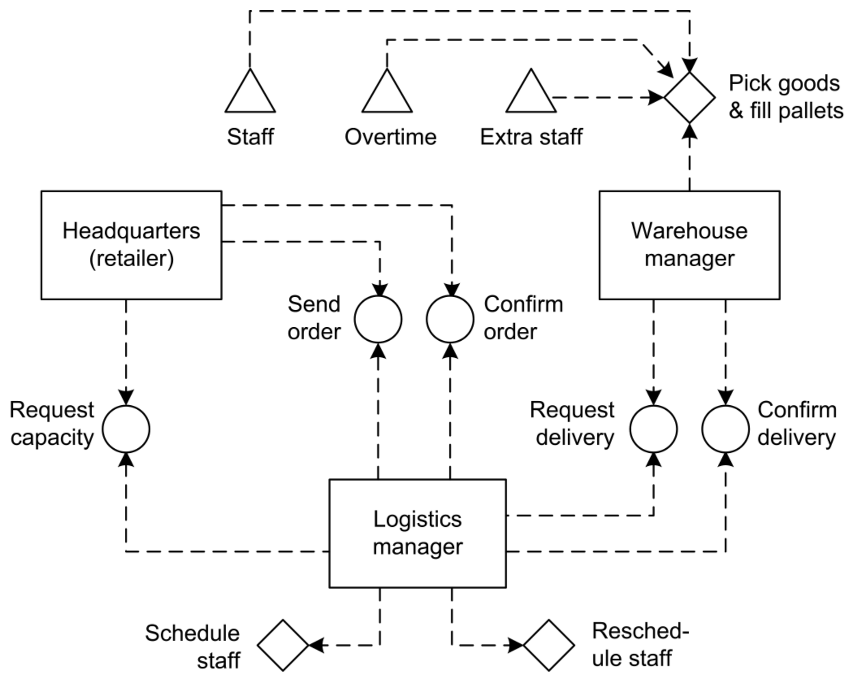

Figure 9. The Resource View of the Interaction Model

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 its estimates, since 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 the increased costs due to insufficient or unused capacity.

Whereas modeling the interorganizational process only touches public behavior at the interface between companies, the development of a simulation model requires a very detailed model of the organization's internal processes. Many organizations might not feel comfortable about disclosing information of this kind to their business partners [70]. In light of this, it was decided to exclude this procedure from the collaborative part and involve only the affected party, in the present case LogCom. The simulation is also an optional part of the method, for it may not be reasonable to develop so detailed a model under all circumstances. The estimated benefit in terms of economies of scale should outweigh the effort spent on the procedure.

Deriving the Workflow Model (Collaboration Model)

A contract is a formal representation of the cooperation between a number of organizations. It defines the role of each party to the contract and the activities

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

Table 4. Excerpt from Simulation Results.

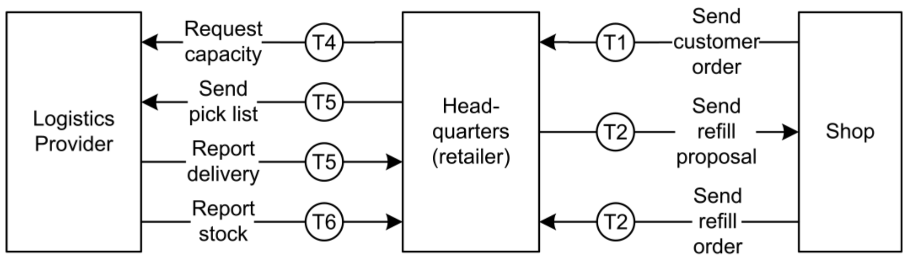

Figure 10. Collaboration Model

they perform in the context of the cooperation. In developing a contract, each transaction is looked at in turn. First, a collaboration model of the transaction is created. The result is a very detailed model with all the steps that have to be performed in the course of the transaction (*see Figure 10*).

Let us consider transaction T5, for example (*see Figure 2*). The aim of the transaction is to handle the order by delivering the ordered items. It starts when Headquarters sends a so-called pick list to LogCom. This list indicates the products to be picked (and delivered) and the respective quantities. The information systems of Headquarters and LogCom are integrated in such a way that the list is sent electronically as a "pick file."

According to the general interaction pattern (*see Figure 3*), the next step in T5 is that LogCom promises to fill the order. But as the warehouse systems of Headquarters and LogCom are synchronized, an out-of-stock cannot occur. This step can therefore be omitted from the collaboration model. The objective action of handling the order is not considered in the contract because it only concerns the internal behavior of LogCom. The next step in transaction T5 is that LogCom reports the delivery. The final step, confirm delivery, is again omitted because it is implied by the receipt of the delivery (refer to T3).

The same procedure is followed for the remaining transactions. This leads to the complete collaboration model shown in Figure 10.

Managing the Interorganizational Workflow

The examples so far have been detailed but only cover only a small part of the case. Here we give a complete account of the case without the negotiation details. The focus is primarily on the "old" architecture of the retail network and the result of applying the method described so far. The project began with an analysis of the business interaction between the companies already mentioned. They already had an established business relationship based on a conventional frame contract. The analysis discovered the structure of the co-operation (*see Figure 11*) and a number of problems, such as broken interaction patterns, missing business rules, unclear communication structures, different contract interpretations, and excessive communication. As a consequence the parties were not satisfied with the current situation.

The approach adopted to solve these problems entailed the creation of a new network to coordinate their interaction. The first step was to negotiate

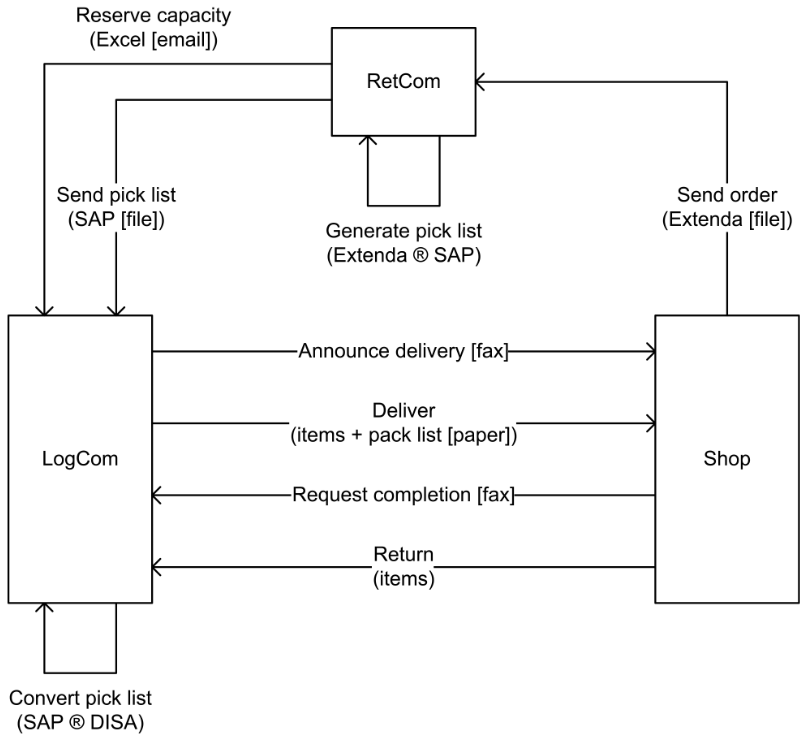

Figure 11. Original Architecture of the Network

the formal contract. This was done in a seminar where the representatives of each company were present and a seminar observer recorded the oral negotiation in writing and translated the requests and commitments into a formal representation by hand in accordance with the procedure described above. The reason for this is that the existing NSSs only support bilateral negotiations. The contract was implemented with the help of YAWL and the YAWL engine as run on a coordination server that connected all parties. The conversions between the involved formats (SAP, DISA, Extenda, and Excel) were performed with the help of XML Script and the X-Tract XML Script processor. This led to the architecture depicted in Figure 12.

In the new architecture, each network partner only exchanges messages with the coordination server. This considerably reduces the complexity of coordination. The server, among other things, forwards messages to the appropriate recipients, converts between formats, and triggers time-controlled messages. The new network architecture also offers ways to improve the efficiency of the communication. In the present 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.

The coordination server has been tested with real data from the involved businesses covering their exchanges during a whole year. This implies roughly

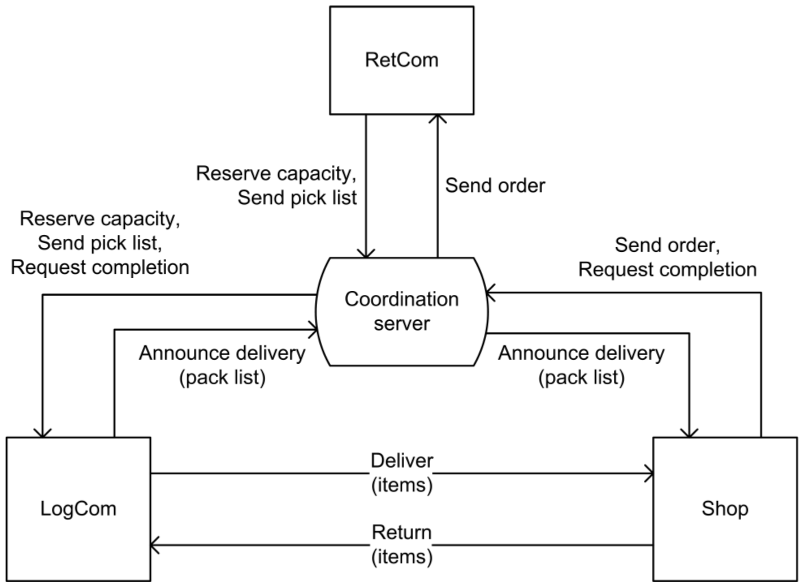

Figure 12. New Architecture of the Network

200 orders and 100 deliveries. The behavior of each of the three parties was simulated manually based on the data. The messages produced by the server matched those that were actually sent according to the empirical data provided by the companies.

A Case Study

The approach described above was tested in a project that involved representatives from both LogCom and their customer, a retail chain. One aim of the project was to improve the existing contract. The approach proposed in this paper made it possible to develop a proposal for a new contract based on a thorough analysis of the interorganizational business process. The vagueness of the old contract had led to a series of problems:

1. *Broken patterns:* In a business transaction, each business act should be related to another in a pattern of initiative and response, and the sequence of business acts must be followed so that the pattern is not broken. Going back to the empirical setting, Headquarters supplied estimates (as an initiative) without getting a response. Thus there was a one pattern of interaction when establishing the framework contract and another when realizing the business transaction, thereby breaking the interaction pattern that glues framework contract and business process. As a result, Headquarters did not know what capacity would be available at the time of order and LogCom did not reserve the required capacity. The estimates made by Headquar-

ters were neither informative nor directive and thus did not imply mutual commitments. As a consequence, the contract had to be specified in such a way that would encourage the parties to keep the patterns intact.
2. *Business rules:* There were no rules that guide the interplay between the overall framework contract and the embedded business transactions. Such rules are necessary to regulate the details of interaction and provide infrastructural support, such as IT systems.
3. *Indistinct communication structures:* It was often unclear who was to communicate with whom regarding which issue.
4. *Lack of trust:* Different interpretations of the contract by the parties led to unfulfilled expectations. This led in turn to a lack of trust in subsequent transactions.
5. *Excessive communication:* A considerable amount of personal interorganizational communication was spent on everyday work. This was necessary only because routine procedures had not been adequately specified in the framework contract.
6. *High transaction costs:* Ad-hoc solutions to exceptional problems increased transaction costs.

Using the approach introduced in the preceding sections, a proposal was developed for a new contract that addressed issues 2 to 6. Excerpts from this contract have been presented in the course of this paper. Further details will not be given here, but the implications of the changes will be discussed.

The explicit representation of business rules in the proposed method takes care of issue 2. The new contract specifies more precisely the obligations of each party concerning their behavior at the interface between the organizations. By reducing the room for interpretation of the contract, this leads to more realistic expectations and ultimately to increased trust (issue 4). The collaboration model clearly states who interacts with whom regarding what issue, which clarifies the communication structures (issue 3) and reduces the amount of "unnecessary" communication (issue 5). Business rules specify the behavior in exceptional situations, eliminating the need for ad hoc solutions. This reduces transaction costs (issue 6).

Related Research

Early research on electronic contracts utilized a limited definition of business processes as business rules (i.e., as pairs of conditions and associated actions) [32]. The workflows are not a part of these contract models but have to be developed in a separate step [41]. This means that the contract model and the workflow model are two completely different models, and considerable effort is required to derive the one from the other [46]. In the present approach, in contrast, the interorganizational workflow is directly negotiated, which makes the development of new models unnecessary.

More recent work, such as [55], uses a full-blown workflow language (e.g., that of Finite State Machines), but the formal workflow model is derived from a conventional written contract rather than negotiated. This raises considerable issues concerning disambiguation and makes it difficult to renegotiate contracts.

Negotiation is an important part of the contracting process, but the approaches mentioned so far do not address it. They focus on how to enact contracts, not on how we arrive at them. As this is of equal importance, however, negotiation has been made an integral part of the method proposed in this paper.

The discussion in this paper also touches on a number of other issues addressed by existing research, most especially the area of formalized or electronic contracts. Although these approaches have been studied extensively [6, 54, 80], they generally aim at transactional contracts in that they cover only a single business transaction such as a sale. The present approach targets frame contracts that cover a number of similar transactions, namely, a complete business process.

Another related issue is that of collaborative modeling [13, 25, 37, 38, 64]. The present 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 widely studied [e.g., 18, 43, 45, 72, 78]. Again the focus of these approaches is mainly on transactional contracts, so they cannot be used in the context of model negotiation.

Interorganizational workflows have also been studied [11, 75, 79]. They too are typically process-based but do not assume the necessity of negotiating these process models.

Conclusions

The method described in this paper pertains to the negotiation and enactment of contracts for business networks. Such contracts are interpreted as interaction models that govern the interactions between network members. The result is a detailed contract describing the workflow of the cooperation. An intermediary phase, simulation, is employed to parameterize and optimize the process. The final phase, contract enactment, consists of the development of a workflow model (collaboration model) to manage the cooperation at an operational level.

Contracts developed in this way are less ambiguous and thus facilitate implementation of procedures and enforcement of rules and conditions. As a result, they can reduce transaction costs, lessen the need for extraneous communication, and enhance the reliability of commitments. Ultimately this leads to increased process quality and improves the mutual trust among the participants in the cooperation.

The research performed so far shows the feasibility of the approach only with respect to a single case. The results need to be confirmed in the context of other cases. There is growing interest in more formalized contracts in the areas of e-commerce and virtual organizations [6, 54, 80]. The existing approaches, because they are so often technology-driven, need to be complemented with rigorous approaches that are strongly business oriented. Formalized contracts (e.g., e-contracts) are not only beneficial in electronic commerce or automated transactions. They can also make a substantial contribution to the support of interorganizational business processes and workflows [79]. The present research takes a step in this direction.

REFERENCES

1. Aalst, W.M.P. van den, and Hofstede, A.H.M. ten YAWL: Yet Another Workflow Language. *Information Systems, 30*, 4 (2005), 245–275.

2. Aalst, W.M.P. van den; Hofstede, A.H.M. ten; Kiepuszewski, B.; and Barros, A.P. Workflow patterns. *Distributed and Parallel Databases, 14*, (2003), 5–51.

3. Adelberg, B. NoDoSE—a tool for semi-automatically extracting structured and semistructured data from text documents. In <<**name (ed.)>,**> *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*. New York: ACM Press, 1998, pp. 283–294.

4. Alchian, A.A., and Demsetz, H. (1972). Production, information costs and economic organization. *American Economic Review, 62*, 5 (1972), 777–795.

5. AMICE. *CIMOSA: Open System Architecture for CIM,* 2d rev. and extended version. Berlin: Springer, 1993.

6. Artyshchev, S., and Weigand, H. Contract-based interoperability for e-business transactions. In D. Konstantas, J.-P. Bourrières, M. Léonard and N. Boudjlida (eds.), *Interoperability of Enterprise Software and Applications.* Berlin: Springer, 2005. **<<pp.?>>**

7. Austin, J.L. *How to Do Things with Words.* Oxford: Oxford University Press, 1962.

8. Baecker, R.M. (ed.). *Readings in Groupware and Computer-Supported Cooperative Work: Assisting Human-Human Collaboration.* San Francisco: Morgan Kaufmann, 1993.

9. Barker, R. *CASE*Method: Entity-Relationship Modelling.* Wokingham: Addison-Wesley, 1990.

10. Bennet, S.; McRobb, S.; and Farmer, R. *Object-Oriented Systems Analysis and Design.* Maidenhead, UK: McGraw-Hill, 1999.

11. Bernauer, M.; Kappel, G.; Kramler, G.; and Retschitzegger, W. (2003). Specification of interorganizational workflows: A comparison of approaches. In **<<name (ed.),>>** *Proceedings of the 7th World Multiconference on Systemics, Cybernetics and Informatics.* <<**place: publisher?>>** 2003, pp. 30–36.

12. Bernus, P., and Nemes, L. A framework to define a generic enterprise reference architecture and methodology. *Computer Integrated Manufacturing Systems, 9,* 3 (1996), 179–191.

13. Bommel, P. v.<<**von, van?>**>; Hoppenbrouwers, S.J.B.A.; Proper, H.A.E.; and Weide, T.P. van den Exploring modelling strategies in a meta-modelling context. In R. Meersman, Z. Tari, and P. Herrero (eds.), *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. Pt. II, Vol. 4278. Berlin: Springer, 2006, pp. 1128–1137.

14. Chandra, C.; Smirnov, A.V.; and Chilov, N. Business process reengineering of supply chain networks through simulation modeling and analysis. In **<<name (ed.),>>** *Proceedings of the Second International Conference on Simulation, Gaming, Training and Business Process Reengineering in Operations.* **<<place: publisher?>>**2000, pp. 345–349.

15. Clemons, E.K.; Reddi, S.P.; and Row, M.C. The impact of information technology on the organization of economic activity: The "Move to the Middle" hypothesis. *Journal of Management Information Systems, 10*, 2 (1993), 9–35.

16. Coase, R.H. The nature of the firm. *Economica, 4* **<<issue and/or season>>** (1937), 386–405.

17. Dean, D.L.; Orwig, R.E.; and Vogel, D.R. Facilitation methods for collaborative modeling tools. *Group Decision and Negotiation, 9* **<<issue and/or season>>** (2000), 109–127.

18. de Moor, A., and Weigand, H. Business negotiation support: Theory and practice. *International Negotiation, 9*, 1 (2004), 31–57.

19. Denning, P.J., and Medina-Mora, R. Completing the loops. *Interfaces, 25*, 3 (1995), 42–57.

20. Dietz, J.L.G. Understanding and modeling business processes with DEMO. In J. Akoka, M. Bouzeghoub, I. Comyn-Wattiau, and E. Métais (eds.), *Proceedings of the 18th International Conference on Conceptual Modeling*. Berlin: Springer, 1999, pp. 188–202.

21. Dietz, J.L.G., and Habing, N. The notion of business process revisited. In R. Meersman and Z. Tari (eds.), *Proceedings of the OTM Confederated International Conferences, CoopIS, DOA, and ODBASE*. Berlin: Springer, 2004, pp. 85–100.

22. Ericsson, K., and Simon, H. *Protocol Analysis: Verbal Reports as Data.* Boston: MIT Press, 1993.

23. Fielding, N.G., and Lee, R.M. *Computer Analysis and Qualitative Research.* London: Sage, 1998.

24. Fox, G. The challenge of convergence. In **<<name (ed.),>>** *Proceedings of the 28th Annual Hawaii International Conference on System Sciences.* Los Alamitos, CA: IEEE Computer Society, 1995, pp. 485–492.

25. Frederiks, P.J.M., and Weide, T.P. van den. Information modeling: The process and the required competencies of its participants. *Data and Knowledge Engineering, 58*, 1 (2006), 4–20.

26. Giaglis, G.M.; Doukidis, G.I.; and Paul, R.J. Simulation assessment of inter-organisational business process re-engineering initiatives. In **<<name (ed.),>>** *Proceedings of the First International Conference on Simulation, Gaming, Training and Business Process Reengineering in Operations.* <<**place: publisher?**>> 1996, **<<pp.?>>**.

27. Giaglis, G.M.; Paul, R.J.; and Doukidis, G.I. (1997). Simulation for intra- and inter-organisational business process modelling. *Informatica, 21*, 4 (1997), 613–620.

28. Giaglis, G.M.; Paul, R.J.; and Hlupic, V. Integrating simulation in organisational design studies. *International Journal of Information Management*, *19*, 3 (1999), 219–236.
29. Goldkuhl, G. Generic business frameworks and action modelling. In F. Dignum, J. Dietz, E. Verharen, and H. Weigand (eds.), *Communication Modeling: The Language/Action Perspective, Proceedings of the First International Workshop on Communication Modeling.* Berlin: Springer, 1996. **<<pp.?>>**
30. Goldkuhl, G., and Lind, M. Developing e-interactions—A framework for business capabilities and exchanges. In <<**name (ed.),>**> *Proceedings of the 12th European Conference on Information Systems.* <<**place: publisher>**>, 2004, **<<pp.?>>**
31. Goldkuhl, G., and Röstlinger, A. Joint elicitation of problems: An important aspect of change analysis. In D.E. Avison, J.E. Kendall, and J.I. DeGross (eds.), *Human, Organizational and Social Dimensions of Information Systems Development*. Amsterdam: North-Holland, pp. 107–125.
32. Grosof, B.N.; Labrou, Y.; and Chan, H.Y. A declarative approach to business rules in contracts: Courteous logic programs in XML. In M.P. Wellman (ed.), *Proceedings of the 1st ACM Conference on Electronic Commerce.* New York: ACM Press, 1999, pp. 68–77.
33. Gurbaxani, V., and Whang, S. The impact of information systems on organizations and markets. *Communications of the ACM, 34*, 1 (1991), 59–73.
34. Habermas, J. *The Theory of Communicative Action 1: Reason and the Rationalization of Society.* Boston: Beacon Press, 1984.
35. Hlupic, V., and Robinson, S. Business process modelling and analysis using discrete-event simulation. In D.J. Medeiros, E.F. Watson, M. Manivannan, and J. Carson (eds.), *Proceedings of the 1998 Winter Simulation Conference.* San Diego: Society for Computer Simulation, 1998. **<<pp.?>>**
36. Holland, C.P., and Lockett, A.G. Mixed mode network structures: The strategic use of electronic communication by organizations. *Organization Science, 8*, 5 (1997), 475–488.
37. Hoppenbrouwers, S.J.B.A.; Lindeman, L.; and Proper, H.A. Capturing modeling processes: Towards the MoDial modeling laboratory. In R. Meersman, Z. Tari, and P. Herrero (eds.), *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.* Pt. II, vol. 4278. Berlin: Springer, 2006, pp. 1242–1252.
38. Hoppenbrouwers, S.J.B.A.; Proper, H.A.; and Weide, T.P. van den Formal modelling as a grounded conversation. In G. Goldkuhl, M. Lind, and S. Haraldson (eds.), *Proceedings of the 10th International Working Conference on the Language Action Perspective on Communication Modelling*. Linköping and Borås **<<where?>>**: Linköpings Universitet and Högskolan i Borås, 2005, pp. 139–155
39. Introna, L.D., and Whitley, E.A. Against method: Exploring the limits of method. *Logistics Information Management*, *10*, 5 (1997), 235–245.

40. Jensen, M.C., and Meckling, W.H. Theory of the firm: Managerial behavior, agency costs and ownership structure. *Journal of Financial Economics, 3***<<issue and/or season>>** (1976), 305–360.
41. Karlapalem, K.; Dani, A.R.; and R.**<<name>>**, K.P. A frame work for modeling electronic contracts. In H.S. Kunii, S. Jajodia, and A. Sølvberg (eds.), *Proceedings of the 20th International Conference on Conceptual Modeling*. Berlin: Springer, 2001, pp. 193–207.
42. Keller, F., and Wendt, S. FMC: An approach towards architecture-centric system development. Paper presented at the 10th IEEE Symposium and Workshop on Engineering of Computer Based Systems, Pasadena, CA, 2003.
43. Kersten, G.E.; Strecker, S.E.; and Law, K.P. Protocols for electronic negotiation systems: Theoretical foundations and design issues. In K. Bauknecht, M. Bichler, and B. Pröll (eds.), *E-Commerce and Web Technologies: Proceedings of the 5th International Conference.* Berlin: Springer, 2004. **<<pp.?>>**
44. Klein, B.; Crawford, R.; and Alchian, A. Vertical integration, appropriable rents, and the competitive contracting process. *Journal of Law and Economics, 21* **<<issue and/or season>>** (1978), 297–326.
45. Köhne, F.; Schoop, M.; and Staskiewicz, D. A meta model for electronic negotiations: Comparison of existing approaches. In Y.H. Tan (ed.), *Proceedings of the 12th Research Symposium on Emerging Electronic Markets "Governance of Electronic Markets."* Amsterdam: Vrije Universiteit Amsterdam, 2005. **<<pp.?>>**
46. Krishna, P.R.; Karlapalem, K.; and Dani, A.R. From contracts to e-contracts: Modeling and enactment. *Information Technology and Management, 6* **<<issue and/or season>>** (2005), 363–387.
47. Lehtinen, E., and Lyytinen, K. An action based model of information systems. *Information Systems, 11*, 4 (1986), 299–317.
48. Liu, K.; Sun, L.; Barjis, J.; and Dietz, J.L.G. Modelling dynamic behaviour of business organisations—extension of DEMO from a semiotic perspective. *Knowledge-Based Systems, 16*, 2 (2003), 101–111.
49. Malone, T.W.; Yates, J.; and Benjamin, R.I. Electronic markets and electronic hierarchies. *Communications of the ACM, 30*, 6 (1987), 484–497.
50. Mann, W.; Matthiessen, C.; and Thompson, S. Rhetorical structure theory and text analysis. In W. Mann and S. Thompson (eds.), *Text Description: Diverse Analyses of a Fund Raising Text.* Amsterdam: John Benjamins, 1993. **<<pp.?>>**
51. Mann, W.C., and Thompson, S. Rhetorical structure theory: Toward a functional theory of text organization. *Text, 8*, 3 (1988), 243–281.
52. Medina-Mora, R.; Winograd, T.; Flores, R.; and Flores, F. The action workflow approach to workflow management technology. In J. Turner and R. Kraut (eds.), *Proceedings of the Conference on Computer-Supported Cooperative Work*. New York: ACM, 1992, pp. 281–288.
53. Menzel, C., and Mayer, R.J. The IDEF family of languages. In P. Bernus, K. Mertins, and G. Schmidt (eds.), *Handbook on Architectures for Information Systems*. Berlin: Springer, 1998, pp. 209–241.
54. Milosevic, Z.; Linington, P.F.; Gibson, S.; Kulkarni, S.; and Cole, J. Interorganisational collaborations supported by e-contracts. In W. Lamersdorf,

V. Tschammer, and S. Amarger (eds.), *Proceedings of the IFIP 18th World Computer Congress "Building the E-Service Society: E-Commerce, E-Business, and E-Government."* <<**place: publisher,**>> 2004**<<pp.?>>**
55. Molina-Jimenez, C.; Shrivastava, S.; Solaiman, E.; and Warne, J. Contract representation for run-time monitoring and enforcement. In **<<name (ed.),>>** *Proceedings of the IEEE International Conference on E-Commerce Technology*. Los Alamitos, CA: IEEE Press, 2003, pp. 103–110.
56. Morris, W.T. On the art of modeling. *Management Science, 13*, 12 (1967), B707–B717.
57. O'Donnell, M. RSTTool 2.4: A markup tool for rhetorical structure theory. In **<<name (ed.),>>** *Proceedings of the First International Conference on Natural Language Generation*. Morristown, NJ: Association for Computational Linguistics, 2000, pp. 253–256.
58. Object Management Group. UML 2.0 Superstructure specification. 2004. UML Resource Page, www.uml.org. <<**not cited in text**>>
59. Object Management Group. Unified Modeling Language: Infrastructure. UML Resource Page, 2006. <<**not cited in text**>>
60. Ould, M. *Business Process Management: A Rigorous Approach.* Tampa, FL: Meghan-Kiffer Press, 2005.
61. Paul, R.J., and Serrano, A. Simulation for business processes and information systems design. In S.E. Chick, P.J. Sánchez, D. Ferrin, and D.J. Morrice (eds.), *Proceedings of the 2003 Winter Simulation Conference*. Piscataway, NJ: Institute of Electrical and Electronics Engineers, 2003, pp. 1787–1796.
62. Paul, R.J., and Serrano, A. (2004). Collaborative information systems and business process design using simulation. In R.H. Sprague Jr. (ed.), *Proceedings of the 37th Annual Hawaii International Conference on System Sciences.* Los Alamitos, CA: IEEE Computer Society, 2004, **<<supply URL>>**
63. Paul, R.J.; Giaglis, G.M.; and Hlupic, V. Simulation of business processes. *American Behavioral Scientist, 42*, 10 (1999), 1551–1576.
64. Persson, A. Enterprise modelling in practice: Situational factors and their influence on adopting a participative approach. Stockholm University, Department of Computer and Systems Sciences, 2001.
65. Reijswoud, V.E. van <<**OK?**>>. The structure of business communication: Theory, model and application. Ph.D. dissertation, Delft University of Technology, 1996.
66. Rittgen, P. Deriving simulation models from business process models. *INFOCOMP Journal, 4*, 3 (2005), 23–31.
67. Rittgen, P. A language-mapping approach to action-oriented development of information systems. *European Journal of Information Systems, 15*, 1 (2006), 70–81.
68. Roboam, M.; Zanettin, M.; and Pun, L. GRAI-IDEF0-Merise (GIM): Integrated methodology to analyse and design manufacturing systems. *Computer-Integrated Manufacturing Systems, 2*, 2 (1989), 82–98.
69. Ross, S. The economic theory of agency: The principal's problem. *American Economic Review, 63*, 2 (1973), 134–139.
70. Rouached, M.; Perrin, O.; and Godart, C. A contract layered architecture for regulating cross-organisational business processes. In W.M. van den Aalst, B. Benatallah, F. Casati, and F. Curbera (eds.), *Proceedings of the 3rd*

International Conference on Business Process Management. Vol. LNCS 3649. Berlin: Springer, 2005, pp. 410–415.
71. Scheer, A.-W. *ARIS: Business Process Modeling*, 2d ed. Berlin: Springer, 1999.
72. Schoop, M., and Quix, C. DOC.COM: A framework for effective negotiation support in electronic marketplaces. *Computer Networks, 37*, 2 (2001), 153–170.
73. Schoop, M.; Jertila, A.; and List, T. Negoisst: A negotiation support system for electronic business-to-business negotiations in e-commerce. *Data and Knowledge Engineering, 47*, 3 (2003), 371–401.
74. Searle, J.R. *Speech Acts: An Essay in the Philosophy of Language.* London: Cambridge University Press, 1969.
75. Shen, M., and Liu, D.-R. Coordinating interorganizational workflows based on process-views. In **<<name (ed.),>>** *Proceedings of the 12th International Conference on Database and Expert Systems Applications.* Vol. 2113. Berlin: Springer, 2001, pp. 274–283.
76. Srinivasan, A., and Te´eni, D. Modeling as constrained problem solving: An empirical study of the data modeling process. *Management Science, 41*, 3 (1995), 419–434.
77. Stamper, R. The semiotic framework for information systems research. In H. Nissen, H. Klein, and R. Hirschheim (eds.), *Information Systems Research: Contemporary Approaches and Emergent Traditions*. Amsterdam: North-Holland, 1991, pp. 515–517.
78. Ströbel, M. Design of roles and protocols for electronic negotiations. *Electronic Commerce Research, 1*, 3 (2001), 335 –353.
79. Weigand, H., and van den Heuvel, W.J.A.M. Cross-organizational workflow integration using contracts. *Decision Support Systems, 33*, 3 (2002), 247–265.
80. Weigand, H., and Xu, L. Contracts in e-commerce. In R. Meersman, K. Aberer, and T.S. Dillon (eds.), *Proceedings of the Ninth IFIP TC2/WG2.6 Working Conference on Database Semantics "Semantic Issues in E-Commerce Systems."* **<<place, publisher>>,** 2003, **<<pp.>>.**
81. Weyland, J.H., and Engiles, M. Towards simulation-based business process management. In S.E. Chick, P.J. Sánchez, D. Ferrin, and D. J. Morrice (eds.), *Proceedings of the 2003 Winter Simulation Conference*. Piscataway, NJ: Institute of Electrical and Electronics Engineer, 2003, pp. 225–227.
82. Willemain, T.R. Insights on modeling from a dozen experts. *Operations Research, 42*, 2 (1994), 213–222.
83. Willemain, T.R. Model formulation: What experts think about and when. *Operations Research, 43*, 6 (1995), 916–932.
84. Williams, T.J. The Purdue enterprise reference architecture. *Computers in Industry, 24*, 2–3 (1994), 141–158.
85. Williamson, O.E. *The Economic Institutions of Capitalism.* New York: Free Press, 1985.
86. Williamson, O.E. *Markets and Hierarchies.* New York: Free Press, 1975.
87. Williamson, O.E. The modern corporation: Origins, evolution, attributes. *Journal of Economic Literature, 19* **<<issue and/or season>>** (1981), 1537–1568.

88. Wilson, R. The theory of syndicates. *Econometrica, 36* **<<issue and/or season>>**(1968), 119–132.
89. Winograd, T., and Flores, F. *Understanding Computers and Cognition: A New Foundation for Design.* Norwood, NJ: Ablex. 1986.

PETER RITTGEN (peter.rittgen@hb.se) has an M.Sc. in computer science and computational linguistics from University of Koblenz-Landau, Germany, and a Ph.D. in economics and business administration from Frankfurt University, Germany. Currently a senior lecturer in the School of Business and Informatics of the University College of Borås, Sweden, he has been doing research on business processes and the development of information systems since 1997 and has published many articles in these areas.