

The Language-Action Perspective and the Semantic Web – A Language-Action Approach to Electronic Contracts

Aida Jertila¹ and Mareike Schoop²

¹ Informatik V (Information Systems), Aachen University, 52056 Aachen, Germany

jertila@cs.rwth-aachen.de

² Information Systems I, University of Hohenheim, 70593 Stuttgart, Germany

schoop@uni-hohenheim.de

Abstract

An electronic business transaction consists of three phases, namely search, negotiation, and fulfilment. The negotiation phase is the most communication-intensive of the three. In order to enable electronic negotiations, the complex communicative exchanges need to be supported by means of information technology. The Language-Action Perspective can provide a suitable theoretical and conceptual basis. In addition to the communicative exchanges, document management also plays a vital role for e-negotiations. Semantic web ideas can be most useful for this part of a negotiation. We will present an integrated approach implemented in the negotiation support system Negoisst that combines LAP and Semantic Web and enables the support of highly dynamic complex electronic negotiations in a business-to-business environment.

Keywords: Electronic Negotiations, Electronic Contracts, Semantic Web, Dynamic Deontic Logic

1 Introduction

In an electronic business transaction, the negotiation phase is the second phase getting input from the search phase (i.e. the list of potential business partners with the goods in question) and, if successful, providing input for the fulfilment phase (i.e. through the business contract). Electronic negotiations are communication-intensive mostly inter-organisational exchanges and are restricted by at least one rule that affects the decision-making or communication process, if this rule is enforced by the electronic medium supporting the negotiation, and if this support covers the execution of at least one decision-making or communication task (Stroebel and Weinhardt, 2003).

Electronic negotiations consist of two main elements (Schoop et al., 2003). Negotiators communicate by exchanging *messages*. The contract as a specific type of *document* evolves during the negotiation. Accordingly, we can distinguish two levels of communication. The message exchange deals with the exchange of requests, (counter-)offers, arguments, discussions, explanations, threats, compliments etc. that form an integral part of a negotiation. Here, the negotiators communicate in a rich, dynamic and highly complex way. These communication steps lead to a more formal exchange concerning the contract versions. Here, the results of the communication steps are documented, especially the binding obligations. In this view, it is no longer relevant why a particular alternative has been chosen but which alternative has been chosen and what the consequences are. It is obvious that the message exchange and the document exchange constitute an electronic negotiation when taken together. Ideally, both are interrelated as done, for example, in the negotiation support system Negoisst that will be discussed in the present paper. Negoisst supports complex communicative exchanges through the exchange of semi-structured message combined with effective document management.

Negoisst has been introduced elsewhere especially in relation to Habermas' and Searle's theories (Habermas, 1981; Searle, 1969) and to LAP in general (Schoop et al., 2003; Schoop, 2005). Negoisst is a negotiation support system providing semi-structured message exchange enriched with semantic annotations and message types integrated with document management and decision support. The focus of the present paper is to discuss the communication aspects of electronic contracts. This is a topic that is usually overlooked in research on electronic contracting. We will argue that while the Language-Action Perspective is a valid basis for the rich communication exchanges, a Semantic Web approach is also required for the operational document management part (Schoop and Jertila, 2004). We will present an integrated approach drawing both on LAP and on Semantic Web. A formal representation of obligations by combining speech acts and dynamic deontic logic will be introduced. Finally, the implementation of contracts in Negoisst that is based on semantic web ideas will be presented.

Our work enables the dynamic establishment and negotiation of rules for contract execution. Furthermore, it is possible to reason about these rules and thus to enable effective monitoring services that can be (partly) automated. In contrast to other applications that only include general contractual rules that can neither be adapted to particular customers nor to the changing facts established during the dynamic execution phase, Negoisst offers new forms of electronic contracting and thus to conduct electronic business.

2 Formal Representation of Contractual Elements

Our aim is to provide an executable electronic contract that can be processed efficiently and effectively. While the negotiation leading to the contract will not be automated at all, the reasoning about it should be done automatically. Firstly, the contractual elements that are related to the communicative exchanges during the negotiation phase need to be represented in such a way that they are unambiguous as to avoid any misunderstandings about obligations and actions to be performed. Therefore, a purely textual representation of an electronic contract is insufficient. Rather, a formal representation is required that can specify what is meant and what consequences will follow. Furthermore, it has to take into account that derivations from the agreed contract often occur during the fulfilment phase. In that case, alternative routes actions have to be performed that represent the real versus the ideal world. Since deontic operators such as permission, obligation, and prohibition play an important role in this context, we have chosen dynamic deontic logic as the formal basis and developed a language called cooL for contract representation.

2.1 Dynamic Deontic Logic and cooL

Deontic logic is a logic that represents norms and normative behaviour. It includes operators for permission P, Obligation O and prohibition F (Aqvist, 1984). Dynamic Deontic logic (Meyer and Wieringa, 1994) is an extension of deontic logic that includes a dynamic operator to represent actions.

The formal language cooL (Schoop, 1998) is an extension of dynamic deontic logic and draws on the work of Dignum and Weigand (1995). cooL distinguishes acts into speech acts and (material or immaterial) actions. The language enables statements about speech acts concerning actions or propositions to be made by an author or a recipient; it enables to specify the resulting effects such as beliefs or obligations. cooL specifies some axioms that can be helpful for deducing obligations and is thus used as the formal basis for the contracting

processes. We will now briefly introduce the relevant axioms. An extended definition of the language can be found in (Schoop, 1998).

The separation of actions into speech acts and other actions can be well applied when dealing with contracts in electronic negotiations. Speech acts are the medium of negotiation exchange and usually occur in the form of messages in electronic negotiations. Material actions are the content of most messages and play an important part in the contract. However, a message can also concern an immaterial action such as a service or information. Hence, the formal language $_{\text{cooL}}$ can be used as a basis for representing the development of contracts. It is able to express even the status of obligations depending on the status of the contractual document (i.e. contract version or final contract).

Based on speech act theory, different message types are distinguished. The author of a message describes his or her intention using such message types. Furthermore, the negotiation protocol is based on the possible order of message types. The five message types that are used in Negoisst and will thus be considered in $_{\text{cooL}}$ are “offer” and “request” for initiating the negotiation, “counter-offer” for replying to and refining an incoming offer, and “reject” and “accept” for terminating the negotiation.

Next we present a simple example of the usage of $_{\text{cooL}}$ in an electronic business negotiation process taken from real-life project contexts.

A textile company sends a request to a yarn producer to deliver yarn until 12.10.2005. This speech act can be expressed in $_{\text{cooL}}$ as follows.

request_{ty} (deliver(yarn)<”date is 12.10.2005”)

This means that the request is sent by the textile company t to the yarn producer y with the propositional content of delivering some product (in this case yarn) before a certain fact becomes true, namely that the date is 12.10.2005.

Some time during the negotiation process, the yarn manufacturers accepts the request, expressed by:

acceptDIR_{yt} (deliver(yarn)<”12.10.2005”)

The following axiom of $_{\text{cooL}}$ can be used:

$\langle b_{\text{ar}}(\alpha) \rangle \text{true} \rightarrow [b_{\text{ar}}(\alpha); \text{acceptDIR}_{\text{ra}}(\alpha)] \text{Ora}(\alpha)$ for $b \in \text{DIR}$

It states that if an author is authorised to issue a directive speech act to a recipient (represented in $_{\text{cooL}}$ by $\langle b_{\text{ar}}(\alpha) \rangle \text{true}$), issues it and the recipient accepts it (represented in $_{\text{cooL}}$ by $b_{\text{ar}}(\alpha); \text{acceptDIR}_{\text{ra}}(\alpha)$), then this leads to an obligation for the recipient to carry out the action indicated.

Dates do not require separate formalisation. They will be expressed by propositions (in this case that the date is 12.10.2005). Together with the previous axiom, we can now deduce that the yarn producer is obliged to deliver the yarn before 12.10.2005:

O_{yt} (deliver(yarn)<”12.10.2005”)

In the domain of electronic commerce, business partners negotiate in the context of buying and selling goods (that can be, for example products and/or services). Therefore, two main roles can be distinguished in a bilateral negotiation, namely “supplier” and “customer”. The role of each partner can be deduced from the type of the first message in the negotiation. If a negotiator starts the negotiation with a message of type request, then Negoisst assigns the role of customer to the initiator and the role of supplier to the counterpart. If, on the other hand, the first message is an offer, then the sender will become the supplier and the receiver will become the customer.

Defining the correct role of each partner is a key issue in order to be able to specify which partner is responsible for which contract section / contract point. A precise and unambiguous assignment of duties to the correct partner is not simple and cannot be realised by considering the message type and the resulting contract version only since one contract version will usually include obligations for both partners. This means that a message of a particular type can include several different aspects. For example, the textile company from the previous example might have sent a message of type request that does not only include the delivery of yarn but also concerns the price. The two elements are often interrelated. An earlier delivery date by the yarn producer might result in a higher price to be paid by the textile company. Thus, a message of type request might in fact consist of different commitments for both partners. It must thus be made clear who is obliged to perform which action.

Therefore, a contract structure has to be developed that explicitly defines contract sections based on the partner who has to execute it. In other words, it is clear for each contract section who is responsible for the contractual elements included. For example, the section “delivery” has to be fulfilled by the user with the role “supplier”. There are also contractual elements that represent a mutual obligation for both business partners. An example is the quantity of the ordered product; the supplier has to deliver the mentioned goods in the quantity mentioned and the customer has to pay for the goods in the quantity mentioned.

The contract is divided into the following parts: supplier-specific, client-specific, and mutual obligations. We can benefit from cooL to deduce the correct obligations. Hence, after specifying the role of each partner (using the type of the first exchanged message), each message will be translated into one of the four following formulae of cooL represented in table 1 with “s” standing for supplier and “c” standing for customer.

Table 1: Formal Representation of Messages.

Sender is “Supplier”	Sender is “Customer”
$\text{offer}_{sc} ([\text{supplier section}(\text{contract version})])$	$\text{request}_{cs} ([\text{supplier section}(\text{contract version})])$
$\text{offer}_{sc} ([\text{mutual section}(\text{contract version})])$	$\text{request}_{cs} ([\text{mutual section}(\text{contract version})])$
$\text{request}_{sc} ([\text{mutual section}(\text{contract version})])$	$\text{offer}_{cs} ([\text{mutual section}(\text{contract version})])$
$\text{request}_{sc} ([\text{customer section}(\text{contract version})])$	$\text{offer}_{cs} ([\text{customer section}(\text{contract version})])$

After the negotiation is closed successfully by accepting the last contract version, the following obligations will be deduced:

- $O_{sc} ([\text{supplier section}(\text{final contract})])$
- $O_{sc} ([\text{mutual section}(\text{final contract})])$
- $O_{cs} ([\text{mutual section}(\text{final contract})])$
- $O_{cs} ([\text{customer section}(\text{final contract})])$

where $O_{sc} (\text{action})$ means that the supplier has the obligation towards the client to perform the action.

3 Implementation of Contracts in Negoisst

A business contract defines the rules and the process for the execution of a business transaction. The contract consists of a static part, which defines fixed contract items that do

not change during the execution of the contract, and a conditional part that defines contractual rules. These rules refer to events (or conditions) that might occur during execution of the contract and change other contract items. For example, a customer might receive a discount of 2% if the payment is done within two weeks after the receipt of the invoice. Another example for a contractual rule is that the product price is decreased if the delivery is late. It should thus be possible to specify rules for each individual contract rather than using only general rules which is the current practice in electronic negotiation systems. The negotiators themselves negotiate about rules and can thus define their common basis for a particular business transaction.

In addition to the contractual elements that have to be managed within an electronic contract, we state the following three goals that the electronic contract has to achieve:

- **Interpretability**

The contract has to be interpretable by both the human user and the machine. Thus, ambiguity has to be eliminated and semantic dependencies have to be explicitly defined. Furthermore, the complexity that is inherent in a negotiation and its result has to be mapped accurately into an electronic contract

- **Extensibility**

Since different business contexts have to be supported by the negotiation support system, the contract structure has to be flexible in a sense that new contract points and contract conditions can easily be added to the initial contract structure. This goal is achieved by introducing the ontology negotiation area as mentioned in (Schoop and Jertila 2004; Schoop, 2004). Ontology negotiations concern the common background of the negotiation partners while contract negotiations are performed based on a commonly agreed contractual structure.

- **Executability**

The possibility to execute the contract automatically is a main advantage of electronic contracts compared to traditional contracts. In particular, the monitoring of the fulfilment phase, the confirmation of actions, and, more importantly, the generation of new obligations or duties in the case of exceptions during the fulfilment phase are crucial issues to enhance a business transaction.

We have embedded the representation of a contract in the negotiation support system Negoisst in a complex (but extensible) framework. This framework includes contractual duties which are the basic elements of the contract. Duties are formulated within contract sections and represented by contract points. A contract point represents a fixed value that is linked to some property of the contract. A contract section is a set of contract points.

To represent duties accurately, all aspects have to be considered. The responsibility, the time, and the way of performing the duties have to be defined clearly and explicitly. This is especially important in an electronic environment, in which the lack of direct communication can lead to ambiguities and misunderstandings (Schoop et al., 2003). Furthermore the complexity of obligations has to be taken into account so that it is also possible to monitor them electronically. Complex obligations are composed of several other obligations. Such complex obligations can only be declared fulfilled if all sub-obligations have been fulfilled.

For this reason, we distinguish two types of duties, namely composed ones and simple ones. We can define a simple duty (or a contract point) as the smallest unit in the contract that is related to a concrete data value, e.g. “delivery date=12/10/2005”. A composite duty, also referred to as contract section, is a set of some other duties (which are either simple or composite) that all have to be fulfilled to declare the composite one as fulfilled. An example is the obligation “delivery”, which is composed of a “delivery deadline” and a “delivery kind”.

The functions to store and manipulate such a contract have to take into account this structure. Figure 1 shows how the electronic contract used within Negoisst is split into several parts, namely an ontology part and a knowledge base. In the ontology part, the metadata of the contract describes the structure of the contract, i.e. the set of contract points that may be defined. Note that this metadata structure may be different for each negotiation as the ontology can be extended and refined during the ontology negotiation phase (Schoop and Jertila, 2004). In addition, we also have instance data, which describe the contents of a specific contract. This includes the agreed points, which are instances of the metadata structures, and the contractual rules, which are stored as rules in the knowledge base. Details on the definition of the rules will be given in section 3.2.

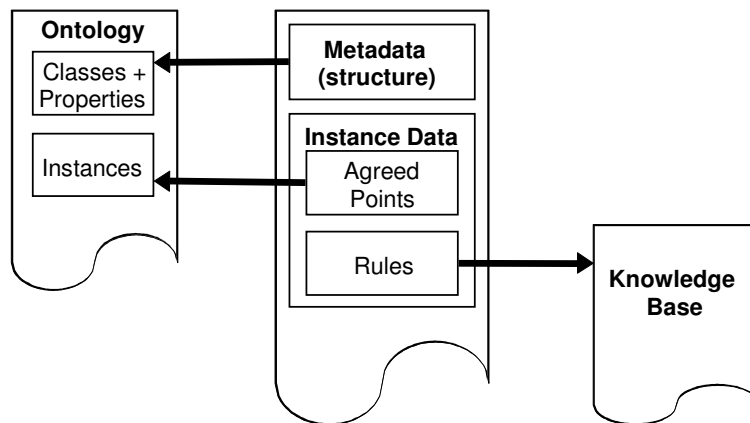


Figure 1: Different Components of an Electronic Contract

The ontology part of the contract is represented in OWL (Web Ontology Language). Contractual elements are defined as concepts in the negotiation ontology. Contractual elements are defined by relationships to other elements (e.g. composite contractual elements) and attributes (e.g. a delivery deadline has a date value as attribute). As mentioned before, we distinguish two kinds of contractual elements, namely contract points and contract sections. Both of them are defined as concepts in the ontology. Contract sections can be interrelated to other contract sections and contract points. Contract points will be defined mainly by attributes. For each type of contract point (or section), there is one specific class in the ontology (e.g. delivery or payment). The defined and agreed data values of the contract are represented as instances of these classes.

3.1 Semantic Structure

The contract structure adapted in Negoisst has to fulfil the criterion of interpretability as well as the criterion of extensibility. Therefore, some guidelines have to be defined which, firstly, guarantee that a resulting contract possesses some standards so that interpretation and executability is enabled, and which, secondly, define which parts of the contract can be

extended and how this can be done. Figure 2 shows a UML Diagram representing the semantic structure of the contract with cardinality restrictions.

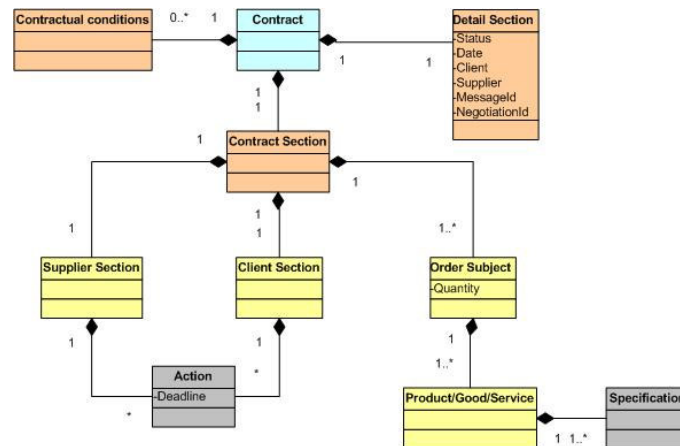


Figure 2: Semantic Structure of an Electronic Contract

Each electronic contract version in Negoisst is composed of three main parts (see also (Salle, 2002)):

3. **Detail Section (Informative part)** which is generated automatically by the system and which includes the unique identifier, the status of the contract (closed/open), the date of contract generation, the involved contract partners, and their roles (client/supplier). The definition of the role for each partner is a crucial issue since it is used for the assignment of obligations (cf. section 2).
4. **Contract Section (behavioural specification)** which includes all negotiable issues of the contract. We distinguish three parts, namely the client section, the supplier section, and the order subject. The client section and the supplier section can include several elements of the type “Action” associated with a “Deadline”. The element action is abstract and can be explicitly defined with attributes (e.g. delivery is an action in the supplier section with a deadline and a deliverykind). These parts explicitly represent the obligations expressed by the action, the actor, and the deadline. The OrderSubject includes information about the subject of negotiation. For example, the product and its specification can be defined in this section.
5. **Contractual Conditions (sanction)** which can define alternative behaviour for specific situation in the execution phase. Each contract can include several contractual conditions.

3.2 Definition of Rules

Since some irregularities occur frequently during the execution of a contract in the fulfilment phase, defining a concrete behaviour for each – or all important – exception(s) (such as late delivery or defective good) guarantees a smoother and more successful business transaction. Contractual rules can be used to represent the exception and the appropriate action to follow. They can be specified by logical rules that are composed of one or more conditions and one single conclusion. An expression such as “in case of late delivery and faulty packaging, the price will be 20 Euro”, is a rule that consists of two conditions and one conclusion. Hence, contractual conditions are an important integral part of a complex contract. An accurate definition and representation of contractual conditions in an electronic contract will provide a basis for monitoring and for deducing new obligations. Furthermore, the formal

representation can be used to perform satisfiability checks of the contract, e.g. if the contract contains contradictory elements.

Contractual rules are basic contractual elements and are the main enabler for reasoning mechanism concerning the fulfilment service. The rules will be composed by combining three different kinds of components; agreed values, actual values and predicates. Agreed values are obviously the values of the contract points as defined before. In the definition of a rule, we will just insert a pointer to the contract point and not the actual value. By doing so, we avoid inconsistencies between the rules and the agreed contract points, and we guarantee that the rules always reflect the latest state of the agreed contract points, since the values of the agreed points can evolve during the negotiation process. Actual values are the ones that will be recorded during the fulfilment phase, as the corresponding actions are executed. For example, the negotiators may have agreed on a delivery date “12/10/2005” (agreed value), but the delivery takes really place on “15/10/2005” (actual value). Obviously, the actual value of a contract point can only be used in a rule if it is possible to record it in the fulfilment phase, hence all assigned agreed point will be automatically marked to be monitored. Predicates are operators or functions that compare actual and agreed values. Such a predicate can, for example, decide whether the agreed value of the delivery date is earlier than the actual one. The conclusion of a rule is an action that generates new obligations or changes contract points, e.g. the price is updated to 20€ in case of late delivery.

It is possible that more than one rule can be applied in a particular situation. In such a case, other rules have to be defined to resolve this conflict. These additional rules are called priority rules and specify a prioritisation (i.e. a partial order) of the contractual rules to ensure unambiguity during the execution of the contract. This feature is provided by the language RuleML which is based on Courteous Logic (Grosz, 2001; Grosz and Poon, 2002). For example, let us consider two rules that specify the action that should be carried out in case of a late delivery. The first rule prescribes that the price should be updated to 20€ in case the delivery is more than 3 days overdue, and the second rule prescribes that the price should be updated to 18€ in case the delivery is more than 5 days overdue. Obviously, if the actual delivery happens to 6 days later than agreed, then the prerequisites of both rules hold, and the conflict has to be resolved by the negotiation support system. For this reason, Negoisst provides some functionality for resolving such kinds of conflicts and takes the decision defined by the priority rules.

The rules are implemented using the language RuleML (Boley, 2001; Boley et al., 2001; Berners-Lee et al. 2001;) since ontology languages do not provide the possibility to define rules. For the same reason, contractual rules are stored separately in a knowledge base, and not together with the contract sections and contract points. However, rules refer to contract points. Only contract points are used for the definition of rules, because they represent the actual values.

3.3 The Fulfilment Monitoring Service

The main functionality of the fulfilment monitoring service is to store and confirm the actual actions/values during the contract execution. These values can then be used to trigger the evaluation of contractual rules, and new obligations might be generated. The result will be documented in a fulfilment report. To guarantee the correct registration of the values, a Trusted Third Party (TTP) should monitor the contract fulfilment (Schoop and List 2001). Actual values have to be confirmed by the TTP before they can trigger the evaluation of a

contractual rule. However, if the business partners trust each other, the TTP is not necessary and the fulfilment service is used only by the business partners. Therefore, a TTP is not vital for our approach and we will not discuss the advantages and disadvantages of a TTP approach in this paper.

The monitoring service is a protected area that can only be used by authenticated users. We distinguish two groups of users, namely users of the negotiation system (i.e. the business partners) and the TTP. The fulfilment monitoring offers three core functionalities: (i) display an overview of the current situation of a business transaction in the fulfilment phase; (ii) insertion or modification of an actual value of a contract point that is being monitored; (iii) confirmation of the validity of this actual value (by the TTP or the business partner). The overview of the negotiation will only be available to the business partners and to the TTPs that have been authorised by them.

Once actual values have been confirmed, the corresponding rules can be evaluated and new duties can be generated. The generation of new duties should be documented by a statement in the fulfilment report. Figure 3 shows how the fulfilment monitoring service works. We can distinguish three main phases:

- **Storing and confirming the actual values of contract execution:** A pair of values is assigned for each contract point used in the negotiation phase, namely an agreed value and the actual value. The agreed value is the value that is contained within the final contract. The actual value is the value entered by the negotiators and confirmed by a TTP during the fulfilment phase.
- **Generating new duties:** Based on the actual values, new facts will be inserted in the knowledge base, i.e. the executable parts of the contract. If this fact corresponds to a condition part of a rule, this rule will be executed and a new agreed value is generated.
- **Updating the fulfilment report:** The fulfilment report contains the main changes to the contract that occur during its execution. Since the contract should not be changed after having been signed, the new agreed values generated by the fulfilment monitoring service will be saved in the fulfilment report.

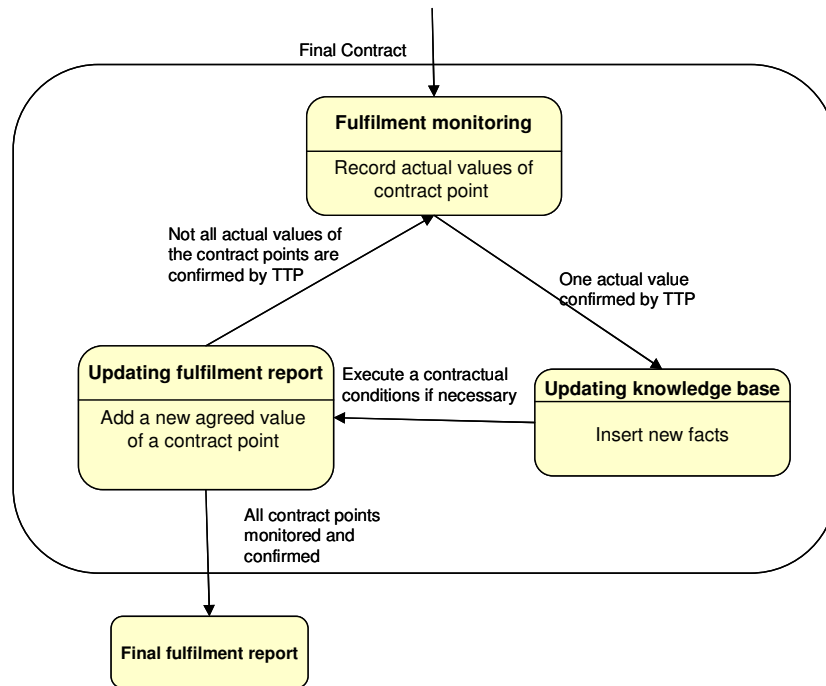


Figure 3: State Diagram of the Fulfilment Monitoring Service

4 Evaluative Scenario

A prototype for the fulfilment monitoring service has been developed as part of Negoisst. This section presents a sample session of the monitoring service. The scenario is based on our successful validation sessions with our industry partners in different projects.

The monitoring service is activated after a successful termination of the negotiation process, since it needs a valid contract as input. The negotiation itself in Negoisst is based on a LAP approach as discussed in (Schoop et al., 2003).

We assume the following scenario: a clothes shop receives an offer from a textile company to buy 300 trousers with some specific features for a unit price of 23 €. Both partners use the Negoisst system. The shop is interested in this offer and replies to the textile company. As a one-time delivery, it is important for the shop to specify a delivery date of “15/10/2004” and a contractual rule stating that “if the delivery is later than agreed, then the unit price will be 20 €”. In addition, a different material is not accepted, which is documented in another rule: “if the delivery is later than agreed and the material is not as described, we will only pay 18 €”. The definition of these rules in the rule editor of Negoisst is shown in figure 4. The user can formulate the rules based on the agreed and actual contract points listed in the boxes. The system also provides a set of predefined predicates that can be applied to the predefined data types.

Add new rule

Please compose the new rule by selecting the variables that should be compared in the "if" part, and by specifying the new value of a contract point in the "then" part.

If: is later than

and: is not the same as

add "and" clause

Then: is updated to/by

Negotiation view

All Negotiation

Figure 4: Editing Contractual Rules

After editing the rules, the clothes shop can define in which order the rules should be executed. This is necessary since both rules overlap in the conditional part and have an effect on the same contract point (namely price). For this purpose, the user can define an order of the rules in the message editor interface shown in figure 5. The right hand part of the figure shows the rules (and their order) that have been defined so far. The left hand part is a free text area in which the user can enter messages to be sent to the negotiation partner as described in section 1. In addition, the user can define the structure of the contract by creating contract points. This is shown in the middle part of figure 5. The definition of contract points is done by annotating a substring of the free text area with an element of the negotiation ontology (cf. Schoop et al., 2003). The clothes shop sends this message to the textile company which then accepts these conditions. This leads to the final contract shown in figure 6 and terminates the negotiation phase successfully.

The screenshot shows the 'Contractversion' text area with the following content: "We find your offer very interesting. But we want to guaranty some profit, so we will only pay 18 € if you deliver late and the the material of the Trousers is not Denim Jeans, but if you only deliver late we intend to pay 20 €. Denim jeans". Below the text is a 'Send' button.

The 'contract negotiation points:' section shows a tree structure:

- ContractPointsSection
 - Order_1
 - Delivery_1
 - DeliveryDeadline_1 (date = 15/10/2004 U)
 - DeliveryType_1
 - OrderSubject_1
 - Quantity_1 (amount = 300 M)
 - Unit_1
 - Trousers_1
 - Price_1 (amount = 23 M)
 - MonetaryUnit_1
 - material_1 (description = Denim jeans U)
 - size_1 (description = 36 M)
 - Payment_1

The 'Conditions' section lists two rules:

- Rule 1: If DeliveryDeadline_1 (date, actual) is later than DeliveryDeadline_1 (date, agreed) and material_1 (description, agreed) is not the same as material_1 (description, actual) then Price_1 (amount, agreed) is updated to/by 18
- Rule 2: If DeliveryDeadline_1 (date, actual) is later than DeliveryDeadline_1 (date, agreed) then Price_1 (amount, agreed) is updated to/by 20

Figure 5: Message Editor During the Negotiation Phase

Contract

§ 1 Details

The following is a **completed** version of a contract that was created on **13 Dec 2004 12:48:52 GMT** .
Between **Fashion trade** as a seller and **Dolce&Gabana** as a buyer the following contract is concluded:

§ 2 Agreed Points

The following orders will be delivered by **Fashion trade**:

§2.1 Order (Order_1)

§2.1.1 Delivery (Delivery_1)

The **DeliveryDeadline** (referred to as **DeliveryDeadline_1**) has been agreed to **15/10/2004** .

§2.1.2 Payment (Payment_1)

§2.1.3 OrderSubject (OrderSubject_1)

§2.1.3.1 Trousers (Trousers_1)

The **size** (referred to as **size_1**) has been agreed to **36** .
The **material** (referred to as **material_1**) has been agreed to **Denim jeans** .

§2.1.3.1.1 Price (Price_1)

The **amount** has been agreed to **23** .

§2.1.3.2 Quantity (Quantity_1)

The **amount** has been agreed to **300** .

§ 3 Conditions

1. If the **actual DeliveryDeadline_1** is **later than** the **agreed DeliveryDeadline_1** , and the **agreed material_1** is **not the same as** the **actual material_1** ,
2. If the **actual DeliveryDeadline_1** is **later than** the **agreed DeliveryDeadline_1** , then the **amount of the Price_1** is **updated to/by** 20.

Figure 6: Final Contract

Now the business partners have to fulfil their obligations. This phase will be monitored by a TTP. The business partners as well as the TTP have access to the fulfilment service, all of them can modify a value of an actual contract points but only the TTP can confirm such values. In our scenario, we assume that the textile company has delivered too late. This fact is confirmed by the TTP. The material has not yet been confirmed by the TTP (see figure 7).

Fulfilment Workspace

Mon Dec 13 15:14:22 CET 2004

List of monitored contract points

[Logout]

For each of the monitored contract points below click on the "modify" button if you want to enter a new actual value or click on the "confirm" button if you want to validate the already entered value. Once you have confirmed a monitored contract point you will not be able to access it anymore.

Contract Point	Type	Actual Value		
DeliveryDeadline_1	date	20/10/2004	<input type="button" value="modify"/>	confirmed
material_1	description	elastan	<input type="button" value="modify"/>	<input type="button" value="confirm"/>

[All Negotiations](#)

Figure 7: Monitoring Report

When the business partners access the Negoisst system to view the fulfilment report, they will find out that the price has been updated to 20€ according to the contractual rule as shown in figure 8. This value might be changed once more if the TTP confirms that the delivered material does not match the agreed material.

Fulfilment Report

§ 1 Details

In the following, updated values of the agreed contract points are listed. These result from the conditions of the contract and due to the actions that took place during the fulfilment phase. Only actions that have been confirmed by a Trusted Third Party of this Electronic Marketplace are being taken under consideration.

§ 2 Updated Agreed Points

1. The updated value of the amount of the Price_1 is 20.

Figure 8: Fulfilment Report

5 Conclusion

Electronic negotiations are the second phase in a business transaction and the most communication-intensive one of all phases. Electronic negotiations exploit the potential of information technology (IT) by offering rules and services that are only possible through IT. Therefore, electronic negotiations are not merely an electronic translation of traditional negotiations in business interactions.

In order to offer effective and efficient support of electronic negotiations, the communication perspective plays an important role. While the automation approaches follow highly structured processes, complex negotiations in business scenarios are highly dynamic. Therefore, system support for such negotiations must enable flexible structures that can be adapted to the current requirements.

The negotiation support system Negoisst enables and supports complex electronic negotiations by combining semi-structured message exchange with semantic enrichment, document management, and decision support (Schoop et al., 2003). In this paper, we have discussed the message exchange and the document management. Negotiators communicate via the exchange of semi-structured messages. Each message has a message type representing the illocutionary point (Searle, 1969). Furthermore, a semantic enrichment process enables the negotiators to specify what they mean by providing a semantics for parts of the natural language message content. The semantic enrichment process and the contractual elements are related to concepts of an ontology. The ontology specifies the background for both negotiation partners. Each message leads to a new contract version. Thus, the contract evolves during the negotiation and is also related to the ontology elements. The contract itself is a type of written communication. It needs to be formally modelled to avoid ambiguities and thus to

enable a clear basis for the fulfilment phase. The language `cooL` provides the formal basis for modelling. Its axioms are used to specify the system behaviour.

Since the negotiation partners do not always fulfil the contract as planned, it is important to specify rules that specify alternative behaviour if problems occur. These rules are formulated in the Rule Markup Language RuleML which is part of the Semantic Web world.

To summarise, while the Language-Action Perspective is the basis for the message exchange in Negoisst, Semantic Web ideas are mainly used for the contract management. Negoisst is based on an integrated approach of communication and document management and, therefore, LAP and Semantic Web are integrated in Negoisst. This integration proved to be very successful. Evaluation took place in a European context in the fashion industry and we could show that Negoisst not only can be applied to complex business exchanges but it enables complex electronic negotiations that were done only traditionally before due to a lack of appropriate system support.

To the best of our knowledge, Negoisst is the only negotiation support system that enables the dynamic negotiation about rules guiding the establishment of contracts and thus enabling the automatic execution of a contract and reasoning and monitoring facilities that can be adapted to each individual contract. By offering this type of metacapability, Negoisst provides the means for new forms of electronic contracting and for new ways of managing and conducting electronic business.

Acknowledgements

Our research has been supported by the European Commission for the research project “SEWASIE (Semantic Webs and Agents in Integrated Economies)” and by the German Science Foundation for the research project “Electronic Negotiation Support for Business-to-Business Electronic Commerce”. We gratefully acknowledge that support. We would also like to thank our project partners and colleagues for valuable discussions.

References

- Aqvist L (1984). Deontic Logic. Handbook of Philosophical Logic, Volume II, Chapter II.1, pp. 605-771. D. Reidel Publishing Company Dordrecht.
- Berners-Lee T, Hendler J and Lassila O (2001). The Semantic Web. Scientific American.
- Boley H (2001). The Rule Markup Language (2001): RDF-XML Data Model, XML Schema Hierarchy, and XSL Transformations, INAP 2001, Tokyo.
- Boley H, Tabet S, and Wagner G (2001) Design Rationale of RuleML: A Markup Language for Semantic Web Rules. In Proceedings of SWWS 01, Stanford.
- DIGNUM F and WEIGAND H (1995). Communication and Deontic Logic. (WIERINGA RJ and FEENSTRA R, Eds), Information systems, Correctness and Reusability, pp. 242-260, World Scientific Singapore.
- Grosf BN and Poon TC (2002). Representing Agent Contracts with Exceptions using XML Rules, Ontologies, and Process Descriptions. In Proceedings of the International Workshop on Rule Markup Languages for Business Rules on the Semantic Web.

- Grosz BN (2001). Representing E-Business Rules for the SemanticWeb: Situated Courteous Logic Programs in RuleML. In Proceedings of the International Workshop on Information Technologies and Systems (WITS).
- Gulliver PH (1997). Disputes and Negotiations: Across-Cultural Perspective. Academic Press, New York.
- Habermas J (1985). The Theory of Communicative Action. Beacon Press.
- Meyer JJC (1988). A Different Approach to Deontic Logic: Deontic Logic Viewed as a Variant of Dynamic Logic. Notre Dame Journal of Formal Logic, 29(1), 109-136.
- Meyer, JJC and Wieringa RJ (1993). Deontic Logic in Computer Science: Normative System Specification. John Wiley & Sons.
- Salé M (2002). „Electronic Contract Framework: An Overview“. American Association for Artificial Intelligence, 2002.
- Schoop M (2005). A Language-Action Approach to Electronic Negotiations. Systems, Signs and Actions, 1(1), to appear.
- Schoop M (2004). The Worlds of Negotiation. In Proceedings of the 9th International Working Conference on the Language-Action Perspective on Communication Modelling (LAP 2004), pp 179-196.
- Schoop M and Jertila A (2004). Electronic Commerce in the Semantic Web Era. In Coordination and Agent Technology in Value Networks, Multikonferenz Wirtschaftsinformatik (MKWI 04), Essen, March 2004, pp. 137-150.
- Schoop M, Jertila A and List T (2003). Negoisst - A Negotiation Support System for Business-to-Business Electronic Commerce. Data Knowledge and Engineering, 47(3), 371-401.
- Schoop M and List T (2001): To monitor or not to monitor- The role of Trusted Third Parties in Electronic Marketplaces. In Information Age Economy, 5th International Conf. Wirtschaftsinformatik, Augsburg, Germany, Physica Verlag, 605-618.
- Schoop M (1998). Towards effective multidisciplinary communication: a language-action approach to cooperative documentation systems. PhD thesis, The University of Manchester.
- Searle JR (1969). Speech Acts: An essay in the philosophy of language. Cambridge Univ Press.
- Stroebel M and Weinhardt C(2003). The Montreal Taxonomy for Electronic Negotiations. Group Decision and Negotiation, 12(2), 143-164.