H. Weigand, W-J. van den Heuvel and F. Dignum. Modelling Electronic Commerce Transactions; A layered approach. In G. Goldkuhl, editor, *Third International Workshop on Communication Modelling (LAP-98)*, pages 47-58, Stockholm, 1998.

Modelling Electronic Commerce Transactions

A layered approach

Hans Weigand [1], Willem-Jan van den Heuvel [1] and Frank Dignum [2]

[1]InfoLab
Tilburg University
P.O. Box 90153
5000 LE Tilburg
The Netherlands
weigand@kub.nl, wjheuvel@kub.nl

[2]Eindhoven University of
Technology
The Nehterlands
dignum@win.tue.nl

Abstract

Since electronic commerce transaction are characterized by their highly communicative character, the Language-Action-perspective proves to be very suitable to provide a deep-structure analysis since it concentrates on the essential aspects of an electronic commerce transaction rather than the form.

In this paper, we present a layered pattern definition language, based on various L/A-perspective approaches. The patterns are represented by means of an extension of the Formal Language for Business Communication. Furthermore, we demonstrate by means of a case study how the notion of patterns can be applied to L/A-based analysis of electronic commerce transactions, and how this meets requirements that Electronic Commerce modeling poses.

1. Introduction

During an electronic commerce transaction subjects interact electronically rather than by physical exchanges or direct physical contact. [kal96]. Depending on the parties that want to interact, we can identify four categories of electronic commerce: business-to-business, business-to-consumers, business-to-government and consumers-government. According to a recent market report, the annual turnover in EC in Western Europe will raise from 1 billion dollars in 1997 to 30 billion dollars in 2001.

At this moment the electronic commerce transactions, that are mainly positioned in the first two categories, usually take place via 'traditional' electronic data interchange (EDI) protocols, such as EDIFACT and ANSI X.12 standard. These protocols are oriented towards 'closed', long-term relationships with high start-up costs stemming from detailed trading partner negotiations. In a sense it can be said that the EDI relations extend the organisation into the partner organisation. Whereby detailed rules are defined about the transactions that can be triggered on both sides and their effect.

For small organisations that have many changing relations with their suppliers and customers this form of electronic commerce is not viable. For this situation Open EDI is developed. Open EDI is directed towards short-term trading relationships, and makes use of one world-wide standard for communication. This standard leads to lower transaction (start-up) costs since there are virtually no asset-specific investments needed. However, the problem with this standard is that it is inflexible and conservative.

A problem that applies to both categories is that many organisations only electrify the existing document flows when they want to start electronic commerce. The forms that are send through EDI are electronic versions of the forms that were normally sent on paper. Implementing electronic commerce this way might be safe, but probably looses out on new opportunities once the transactions are performed electronically. In order to start electronic commerce the organisation should rethink (redesign) their trade procedures, and see how they would be best implemented when the transactions are performed electronically. Electronic transactions have different features from paper ones, which might make some steps superfluous and create the need to introduce new steps.

In this paper, we propose the use of a layered model of patterns to model electronic commerce processes. This model is based on the Language/Action Perspective. Since the language/action perspective is focused on the communication between participants rather than the physical process it is very well suited to model the essential level of the process. We will show that this is crucial for modelling electronic commerce processes. The model is also formalised which makes it a suitable candidate to model Open EDI. In this respect it offers far more flexibility than the existing or proposed standards that usually standardise on a syntactic instead of a semantic level.

In a previous paper [heuvel97], we have discussed how the language/action perspective could be used to ensure the validity of electronic commerce transactions. As we will see, this theory to define the "static relationships in EC-domains" provides us with a solid foundation to catch more dynamical aspects of electronic commerce transactions.

The remainder of this article has been structured as follows. We will start in section 2 with a discussion of the special requirements that Electronic Commerce poses on the analysis of EC processes. In section 3, we proceed with a description of various methodologies that have been based on the L/A-paradigm. On the basis of these methodologies, we have designed L/A-based patterns. These patterns are organised in a layered architecture that enables the business analyst to reuse components of lower layers and to reduce complexity. We discuss these patterns in section 4. Thereafter, we present a case in section 5 taken from the domain of international trade. In the next section, we demonstrate how the layered model of patterns can be used to model the deep-structure, or essence of the transaction that can be identified in the case. Section 6 finishes this paper with some discussion, conclusions and future research.

2. Modelling Electronic Commerce processes

Although there is already an extensive body of theory about modeling information processes for companies ([hammer93], [schäl96]) not much is known about modeling processes that are used in electronic commerce. Of course one might use the techniques that are used in WorkFlow Management like Petri-nets or Data Flow Diagrams. However, we feel that electronic commerce processes differ in at least one aspect from other business processes. They cross the organisation boundaries. We will argue that this implies that most methods that are used to model traditional workflows are less suitable to model processes for electronic commerce.

There are two features of processes that cross organisation boundaries, which distinguish them from normal workflows. First the resources that are needed for different parts of the process cannot be assigned centrally as they reside in different organisations. Secondly, the organisations have at least a certain degree of autonomy. Depending on the type of relation between the organisations this autonomy might be bigger or smaller but will always be bigger than the autonomy of parts of one organisation. This autonomy has large consequences for the implementation of the procedures. This is clear from the fact that organisations have to decide for each process whether they will perform it themselves (within the organisation) or delegate it (to the market). The organisation is structured in a way that the core processes can be performed efficiently. Thus certain elements that influence the management of the processes are build into the organisation structure. E.g. an employer can order an employee to perform some task at a certain moment. The employee has to perform this task because he has some contract with the organisation that employs him to perform tasks. That is, the responsibilities of the employees to perform certain (types of) tasks are all fixed through the organisation structure. The organisation also arranges for standard information flows regarding this task.

When an organisation decides to have some part of a process to be performed by another organisation (through a market mechanism) they gain flexibility and might perform the process cheaper if the other organisation can work cheaper than when they do it themselves. However, the elements that were build into the organisation structure to manage the process have to be taken care of explicitly in this case. In traditional economic transactions this is done through the introduction of numerous checking mechanisms usually consisting of forms that have to be signed and send between the partners. We argue that the elements that are modeled explicitly through these forms are *responsibility*, *trust*, *commitment*, and *deontic* concepts such as *obligation*, *prohibition*, and *permission*.

In the case that the transaction processes are going to be performed electronically they can be redesigned. Some of the steps are no longer needed because the information is send through a different medium. E.g., electronic money transfers arrive almost instantly and can be checked automatically within seconds. In that case there is no time difference between the moment the partner states that he paid the money and the moment the money actually arrived. Therefore the shipping of the goods can be started the moment the message arrives and no separate steps are needed to ensure that the payment is actually made after the goods are shipped. However, the underlying (deontic) concepts do not change when a different medium is used.

Once it is decided to redesign the transaction process when it is performed electronically, the new process should of course also fulfil all the requirements regarding responsibilities, trust, commitment, etc. Therefore the modeling method that is used to model the processes for electronic commerce should support these concepts. Although the traditional techniques used for workflow modeling such as Petri-nets can be used to model these concepts they do not contain these concepts intrinsically. I.e., in a Petri-net I might model the concept of trust between partners as is done in [Bons97], but Petri-nets do not give me concepts that guide me in doing this. These concepts are only taken into account when we look at the *essential* level [dietz96] of the processes. On this level the processes are seen as a means to achieve some task of an organisation through the cooperation of several people. On this level we do not model the physical process but the conversations that manage that process. We will show in this paper that the language/action perspective offers the right tools to model this level and thus captures the concepts mentioned above. Therefore the modeling methods based on the Language/Action perspective will also be able to support the modeling and building of secure processes for electronic commerce.

3. Communication Modeling Methods

Before we describe our own model that is based on a layered model of patterns, we briefly describe three prominent, related L/A-modeling approaches that focus on business process modeling, the DEMO method, the Action WorkFlow Method and the BAT-method. We will show later on how our model has incorporated features of each of these models in the different layers of our own model. For a detailed discussion and comparison of these methods we refer to: [dietz96], [verharen97].

3.1 DEMO

The Dynamic Essential Modeling of Organizations Method, that has been developed by Dietz, focuses on business processes (or transactions) at the essential level. The essential level catches the archetypal form of an organisation in terms of (essential) transactions [dietz96a], [dietz96b], [dietz94]. A transaction is constituted of communicative as well as objective action between interacting subject that aim at establishing a new fact in the object world. A transaction can be decomposed in three phases: the order phase, the execution phase and the result phase. During the order phase, the subjects are engaged in an actagenic conversation. During the actagenic conversation an actor (for instance a customer) requests something from another actor (like a supplier), which the latter can reject or accept. This leads to a commitment, or obligation for the supplier to keep his/her promise. The result phase starts after the executor, e.g., the actor that has commitment himself to perform a certain action during the actagenic conversation, has created the desired state of affairs (during the execution phase). During the result phase a factagenic conversation takes place in which the executor declares that (s)he is finished, and followed up by an acceptation or rejection of the initiator. The factagenic conversation also leads to an obligation. This time (at least in the case of an acceptation by the initiator) to the customer to accept the result of the material action performed by the executor.

3.2 Action WorkFlow

The Action Workflow Approach has been introduced in the beginning of the 90s with an emphasis on workflow modeling [med92]. This approaches describes business processes as a loop consisting of four phases (see figure 1). The workflow loop starts with a proposal, a request from the customer (or initiator) or an offer from the performer (or executor). In the second phase, the customer and the performer come to an agreement. After the performer has executed the promised action, he states/declares that (s)he is finished to the customer. In the last phase, the satisfaction phase, the customer can declare to the performer that the transaction was (un)succesful.

The workflow loop is comparable to the transaction concept of DEMO, however, it only represents the communicative action and does not take the material action into account. The sequence in which transactions (the DEMO transaction counterpart) take place, is represented in the transaction process model. This model, that is not included in the conversation of action theory, consists of three levels, to be the success layer, the discussion-and-failure layer and the discourse layer [reijsw96].

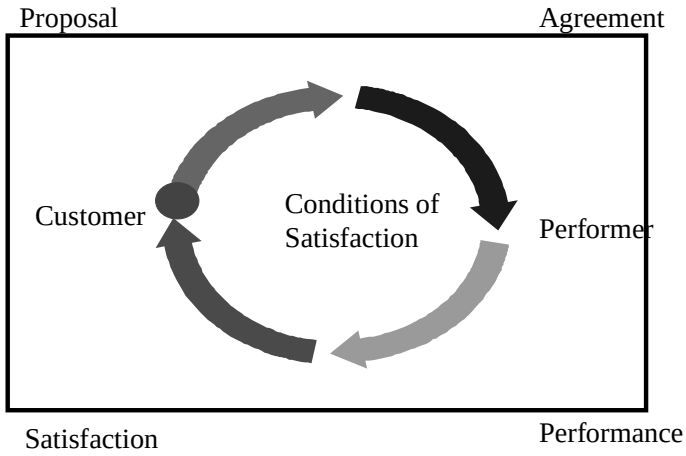


Figure 1: The Action Workflow loop

3.3 BAT

The third L/A-based business process modeling approach we discuss in this section is the Business Action Theory. The Business Action Theory is the successor of the Business as an Action game Theory. The BAT-

framework emphasises the *interaction* between the subjects during a business process. Furthermore, it provides in a more detailed framework, that includes the relations between various business actions.

The BAT framework resembles the workflow loop. In BAT, the first phase of the workflow loop, the proposal is refined in subsequently the business prerequisite phase, the exposure and contact phase and the contact establishment and proposal phase.

4. Layers and Patterns

Patterns play an increasingly important role during the analysis and design of information systems. The notion of patterns is based on the architectural patterns, as introduced by Alexander's who defines patterns as 'descriptions of communicating objects and classes that are customised to solve a general design problem in a particular context'. The roots of the pattern movement lay in several publications in the beginning of the 90s. However, the idea of patterns really broke through in 1995, after the book about design patterns [gamma94] written by the 'gang-of-four' has appeared. Since that time, numerous publications about patterns have appeared from design patterns, to analysis patterns [fowler97] and architectural patterns [busch96].

In [weigand98], we have proposed to apply the notion of patterns to L/A-model constructs. This has lead to one architectural pattern, a hierarchical layered pattern and one communication pattern for each level of abstraction. Since our patterns do not relate to particular problems, we used the term meta-analysis patterns in [weigand98], but for reasons of simplicity will use the sec term 'patterns' here. The patterns are used for two purposes in this context. First they can guide the modeling process through the use of a library of patterns that form the building blocks for each level. Secondly they improve the reusability of components. Often the same patterns can be used in different applications. Only the parameters will be different. This feature is even strengthened through the use of different layers of models. In this paper we will not explore this feature of patterns as it can only be visualised through the presentation of several cases, which would take to much space and would not add to the main points of this paper.

The patterns are represented as extensions of the Formal Language for Business Communication (FLBC) of Kimbrough [kim96]. We have chosen for this language because it not only focuses at the syntax but also on the semantics. FLBC represents a message as a sequence of speech acts, typically assertions and declarations that form the basis of potential reasoning procedures. FLBC uses the following elements to describe the speech act: the speaker and the hearer, the illocutionary force, the content and the context. FLBC-II that is defined in [kim97], distinguishes between three illocutionary points, to be an assertion, a request and a query. These three atomic speech acts are used to represent a variety of message types, such as appointments (assertions), staff action messages (directives) and read/review/comment messages (directives). The context in which the communication takes place, is represented by means of either the message-ID to which is responded, the time when the message was sent, the machine-ID from which the message was sent and/or the persons to which the message is cc-ed. The machine-ID can be interpreted as being a means to establish the identity of the sender. In this way, the agent is anchored to a 'real-world' person. In this paper, we will not focus on the (representation of the) content of the message, that is,. the proposition.

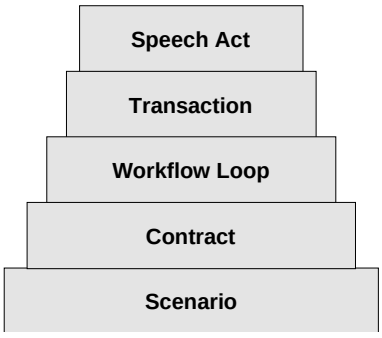


Figure 2: The Layered Pattern Architecture

In the following we will give a detailed discussion of the (layered) patterns as represented in figure 2.

4.1 Speech Acts

(Formal) representation languages such as the Formal Language for Business Communication (FLBC) and methods based on the Language/Action Perspective assume that the speech act is the most elementary unit within the communication between subjects.

The speech act theory is founded by Austin who developed the 'language as an action theory'. A speech act focuses on what people are 'doing in saying something' [austin62]. A speech act can be defined as an utterance that in it self is constituted of a performative act, like requesting and promising. According to Searle [searle69], speech acts are constituted of three parts: the propositional contents, the illocutionary context and the illocutionary force. The propositional content describes what the speech act is about. The illocutionary context represents the relevant background information of the speech act, like the speaker, addressee, time, location and circumstances. Finally, the illocutionary force indicates the intended effect of the communication. The main component of the illocutionary force is the illocutionary point that describes the causality and intention of the communication. Searle distinguishes between five different illocutionary points: assertives, directives, commissives, expressives and declaratives. This taxonomy defines what the speaker can do on the basis of an utterance, with a propositional content.

These speech acts can be, more or less explicitly, found back in the electronic commerce communication. In our representation of speech acts we include the following elements. The name of the speech act, like "accept_request". The sender and receiver of the speech act. For these elements usually only the type restrictions are indicated in the pattern, e.g., the sender of the speech act must be a person or a company.

We also include the illocutionary point which is restricted to assertions, accepts, commits and directives. Finally we include the propositional content in the speech act representation. We suggest that in a given domain, a small library of speech act patterns can quickly support 80 % of the actual messaging.

For the modeling of electronic commerce processes the most important point that is included in the speech acts is the illocutionary point. It indicates the intended effect of the speech act and therefore gives the purpose of this action. The purpose of the action is what is important on the essential level of modeling processes.

4.2 Transactions

A transaction is defined by us as the smallest possible sequence of actions (speech acts) that leads to a certain deontic state, in other words an obligation or an authorisation, or an accomplishment. These deontic consequences of (a sequence) of speech act play an important role during the representation of the electronic commerce transaction, because together they define the contract between the two parties. In general, it requires a "hand-shake" of at least two messages before a communicative action is realised. For example, the combination of the request of the customer to deliver a product, and the promise of the supplier to actually deliver it, constitutes a deontic effect, e.g. an obligation to the supplier to deliver the product, and an obligation of the customer to pay the agreed price.

According to Verharen [verharen97], a transaction can be represented by means of a set of communicating subjects, communicative actions, constraints on the sequence of these actions and the goal and exit states.

Particular kinds of transactions are the factagenic and the actagenic conversation in DEMO, each constituted of two speech acts (see 3.1). We take these two conversation types as archetypal transaction patterns, but there will be more, for example, for identification.

4.3 Workflow

At the next level, we identify a workflow that is called a transaction in DEMO or a business process in BAT. A workflow is a set of transactions aimed at some *goal*. Only at this level, the obligations, authorisations and accomplishments are modelled. They are modelled as states, which are connected through transactions. The reason for introducing a workflow level is to create patterns that are closed towards a certain goal. The archetypal workflow pattern is the workflow loop from the model of the basic conversation of action, as defined by Winograd and Flores (see Figure 1). This model expresses that actions are always executed *for* someone, so on request, and need evaluation to establish their accomplishment (cf. 3.1, 3.2). Because the loop is closed it automatically will fulfil some of the auditing requirements that require a feedback or check between the partners.

4.4 Contract

Both DEMO and Action Workflow give a rather asymmetric perspective of the business process. The analyst must either choose the viewpoint of the initiator or that of the executor of the transaction (in our case, the customer or the supplier). We follow Goldkuhl who claims that a business transaction must be interpreted as being an 'interchange process between a supplier and a customer' and that it 'involves the creation and sustainment of business relations' [gold96]. In this sense, many L/A-based methods model only half of the transaction.

The interchange process is what Taylor calls a symmetrical type of exchange (Taylor93,p. 211). In this type of exchange all actors involved in the conversation, have a common interest in a particular object. In case of electronic commerce, the actors are typically the consumer, supplier and one or more third party(-is), whereas the common object of interest is the product.

The reciprocal approach does not lead to two individual representations of the same business transaction, but

rather suggests two interrelated workflows. Figure 4 shows an example of a reciprocal transaction pattern. The customer requests a certain product. The supplier on the other hand, requests money for it in return. Both transaction patterns are coupled by means of an agreement on the terms of exchange. This agreement that described the mutual obligations and authorisations is often called a contract. Whenever the agreement is reached, the transaction takes place. Note that the parties may agree in advance on certain temporal constraints between the two workflows, for example, that payment is done after delivery. Such constraints have the effect of putting the risk on one side or another.

At the end of a contract the mutual obligations of the partners should be fulfilled. This resembles exactly the idea of a real world contract that governs the business relation between two partners. It governs the behaviour of the two parties until the end of the contract at which time there should not be any "spurious" obligation on either side. Both parties are "free" again.

4.5 Scenario

L/A-based methods focus on conversation patterns, like the basic conversation for action pattern [Wino86]. However, we take over the hypothesis of Taylor that the representation of the conversation, e.g. the inter-action, must be translated into a text to be understood. As has been stated by Taylor [taylor96], the context of a conversation is defined by 'identities of the speaker and hearer, physical and other incidental circumstances of time and place, the object of the conversational exchange, and the probable intentions of the speaker'.

In other words, in order to be interpreted the conversation has to be placed in a certain context. As has been argued in [heuvel97], an important element of the context of an electronic commerce transaction is the identification of both communicating actors, as well as their actions. By means of the identification speech act, the speaker tries to set up a long term relationship with the hearer.

Within this relationship particular agreements can be made about the effect of certain communicational actions. After the creation of this domain or world, the speaker and hearer can enter it by showing their identity.

The text structure (pattern) is composed of a begin, a development and an end. Within the context of an electronic commerce transaction, this can be translated into terms of: identification - the essential transaction - and ending the relationship (writing out). During the event, one or more reciprocal transactions can take place depending on the nature of the transaction. Note that the scenario typically exceeds the level of the contract and describes several contracts, for example, a contract between supplier and factory and between factory and retailer (supply chain). The scenario can also be used to describe that a contract is executed in the background of another long-term contract, which might supplement it or guide it in case of inconsistencies between several concurrent contracts. Thus the scenario also describes the interactions between several contracts that run concurrent between several parties.

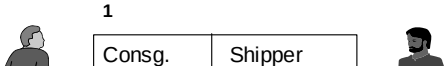
5. The Case

In this section, we apply the layered approach to a case of Electronic Commerce. The purpose of this case is both to illustrate the layered approach itself, and its applicability for modelling Electronic Commerce. The case that we present in this paper is based on the synopsis of the case presented by Bons in [BONS97], the so-called "open-account post-payment scenario".

In this case, the shipper has produced goods for the consignee, arranges the transportation (by sending a transport instruction to the sea-carrier) and transfers the goods to the carrier. The consignee has to pay a price for these goods and services in return. The agreement between the consignee and the shipper has been formalised in the C-terms sales contract (this indicates some terms of contract from an international standard), which assumes that the transfer of goods will be performed by a third-party agent, e.g. the sea-carrier.

In other words, the transportation is an outsourced activity. The C-term sales contract states that the obligation of the shipper is to properly hire the sea-carrier; so this party is not responsible for the successful delivery of the goods. The sea-carrier is dependent on the delivery of the goods by the shipper. The shipper has to pay freight in return for the services of the sea-carrier. The payment of the freight by the shipper as well as that of the consignee, are facilitated by another third party, a bank. So, there also exist relationships between the shipper and the bank (to pay freight to the shipper) and the consignee and the bank (to pay money for the goods and transportation). The payment task of the bank also can be perceived as an outsourced activity. The bank needs a receipt of the funds to be transferred from the principal (e.g., the shipper or the consignee) and a receipt of the payment order that indicates the receiving party. To conclude the structural aspects, we assume that the shipper does not trust the consignee and vice-versa. Besides that the sea-carrier does not trust the consignee nor the shipper. However, the shipper and the consignee trust the sea-carrier, and the bank is trusted by all subjects.

The procedural aspects of the case are as follows. The shipper produces the goods and subsequently arranges the transport of the goods by sending a transport instruction to the sea-carrier. Once the shipper has transferred the



goods to the sea-carrier he will instruct his bank to pay the freight to the sea-carrier by sending a receipt that contains the amount of money, and another receipt that prescribes the order.

Figure 3: Graphical representation of the contracts between the subjects

Finally, the shipper waits until the consignee has paid for the goods through the consignee's bank. The sea-carrier is triggered by receiving the instruction of the shipper. Thereafter he will wait with actually transporting the goods until he has received the goods and a payment of freight through the shipper's bank. Lastly the consignee is triggered by receiving the goods from the sea-carrier. Thereafter, he will transfer the payment through his bank to the shipper.

6. Elaboration of the case

In this section, we show how the layered model of patterns can be used to model the case. We have analysed the case in a top-down manner, from the high-level scenario to the speech acts. Because of the limited space, we will only present one example pattern of each level.

6.1 Scenario

The case describes one scenario, that of a delivery of goods between a consignee and a shipper facilitated by two third parties: a sea-carrier who takes care of the transportation of the goods ordered by the consignee, and, the bank that provides facilities to transfer money from the account of the consignee to the shipper, and from the shipper to the sea-carrier.

Below we have given a formal description of this scenario. We have identified four roles that the subjects can play: a customer, supplier, carrier and bank role.

The scenario starts with the identification of the parties involved in the electronic commerce domain (see [heuvel97]) by the domain administrator (here the chamber of commerce), and other domain knowledge such as the law that applies. Thereafter, it describes the various bilateral contracts that have to be set up between the subjects.

Note that the introduction of the C-terms contract between the consignee and the shipper at first sight results in an 'incomplete' contract "shipper/sea-carrier", since the shipper is only obliged to hire the sea-carrier and to pay the freight. So there seems to occur an actaganic transaction between shipper and sea-carrier, whereas the actagenic transaction - establishing the fact that the goods have been delivered - is performed between sea-carrier and consignee. The sea-carrier does not have to declare to the shipper that he has delivered the goods. It follows then that although there is a transaction between sea-carrier and consignee, there is no contractual relation between them, which is in fact quite intuitive. This is one way of modelling the case, in section 7 we will discuss an alternative approach.

The scenario ends with the termination of the relationships between the subjects. We have not worked out this part yet.

```
ScenarioType post-payment shipment;
```

```
(domain IC(subjects [person($consignee), person($shipper),
person($sea-carrier), person($bank)],  identification
["Chamber.of Commerce  Rotterdam"], law ["Dutch Trade Law"]);

contracts ([
shipper/consignee($shipper, $consignee)
shipper/sea-carrier($shipper, $sea-carrier),
shipper/bank($shipper, $bank),
consignee/bank($consignee, $bank)]);

transactions ([
deliver_goods($sea-carrier, $consignee, $goods, $date))],

termination
)
```

6.2 Contract

In this subsection, we have worked out one contract that has to be set up between the shipper and the sea-carrier (see figure). In the contracttype specification two roles are discerned: the customer and the supplier. The contract specifies the workflow loops that interact, and temporal conditions between these loops. For example: the request to transfer the goods from the shipper must take place before the request to transfer the payment.

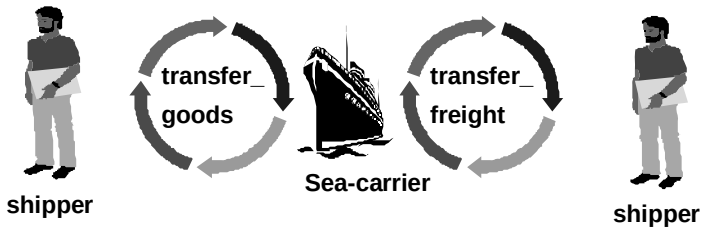


Figure 4: Contract between the Shipper and the Sea-Carrier

```
ContractType Shipper/Sea-Carrier (customer($shipper),
supplier($sea_carrier), product($transfer_goods), date($date) ==
      ([person($shipper), person($sea-carrier)], transfer_goods($shipper,
      $sea-carrier, $transfer_goods, $date), [person($sea-carrier),
      person($shipper)], transfer_freight($sea_carrier, $shipper,
      $transfer_goods, $date),
      transfer_goods.request_transfer_goods BEFORE
      transfer_funds.request_transfer_freight)
```

The contracttype Shipper/Sea-carrier consists of two workflow loops: “transfer goods” and “transfer freight”. Furthermore, the contracttype defines constraints between (speech) acts between both workflow loops: the request to transfer goods in the transfer goods workflow (see 6.3), has to be preceded by the request to transfer freight in the transfer_funds workflow loop.

Note that for illustration purposes we have added an additional constraint to the actual casus description: the sea-carrier can only ask to transfer the freight from the shipper, after the transaction in which the shipper has requested the sea-carrier to transfer the goods is finished successfully. In the default case, no constraints are specified and the two (transactions in) the two workflows can run truly in parallel.

6.3 Workflow Loop

The Workflow Loop Pattern represents the uni-directional agreement between two parties in terms of obligations, authorisations and accomplishments. The contract specified the bi-directional relationship between the shipper and the sea-carrier. Below we have presented the workflowtype definition of the agreement between the shipper and the sea-carrier from the perspective of the shipper. After the sea_carrier has requested a transfer_of_goods, the sea_carrier is obligated to transfer the goods.

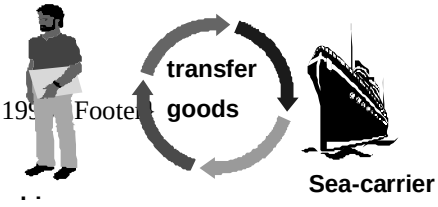


Figure 5: "Transfer goods" Workflow Loop

```
WflType transfer_goods (initiator($shipper), executor($sea_carrier),
            product($transfer_goods), date($date)) ==
      ([person($shipper), person($sea-carrier)],
      /* Obligation of the sea_carrier to transfer goods after request
      /* of the shipper
      S1: OBL($sea_carrier, transfer_goods)
      in request_transfer_goods($shipper, $sea-carrier, $price, $date)
      goal accept_transfer_goods ($sea-carrier, $shipper, $price)
      exit cancel (request_transfer_goods) --> Cancel_Request)
```

The "transfer_goods" workflow is triggered by the request to transfer_goods from the shipper to the sea_carrier. The workflow loop ends either if the goal is accomplished (transfer_goods) or if the shipper cancels the request. As one can see in the above, the workflow consists of two transactions: request_transfer_goods and accept_transfer_goods. The first transaction leads to a new deontic status where there is a new obligation; the second transaction also implies a change in the deontic state since the obligation is removed after the acceptance.

We could easily extend the example with deadlines and authorisations, e.g., the transfer of the goods should take place within one week after the request has been made. The transfer of the goods automatically leads to the authorisation for the sea-carrier to request funds.

6.4 Transaction Pattern

The transaction pattern describes the smallest, atomic unit of communication that leads to an obligation, authorisation or accomplishment. We take as example the transaction in which the shipper requests shipping of the goods from the sea-carrier. Also at this level, it is possible to specify temporal constraints between the elements (in this case, the speech acts) if necessary.

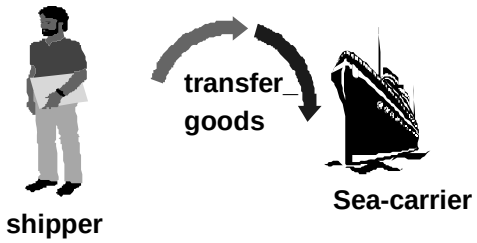


Figure 6: "Transfer goods" Transaction

```
TransType request_transfer_goods (speaker($shipper),
addressee($sea_carrier), product($goods), date($date) ==
      ([person(shipper), person(sea-carrier)],
      [request_transfer_goods($sh,$se,$date, msg1),
      accept_request_transfer_goods($se, $sh, $date, msg2),
      before(msg2,msg1))
```

6.5 Speech Act

The lowest level we identify in our framework is that of the speech act. The messagetype definition describes the propositional content "the request to transfer goods with the name $transfer_goods on date $date", the illocutionary force "accept".

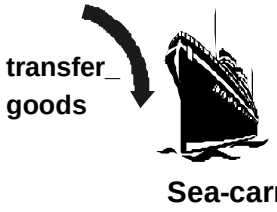


Figure 7: "Transfer goods" Speech Act

```
MsgType accept_request_transfer_goods (sender($sea_carrier),
receiver($shipper),product($transfer_goods), date($date) ==
      (person($sea_carrier), person($shipper), accept,
      request_transfer_goods($transfer_goods, $date))
```

7. Discussion and Conclusion

Although for reasons of space, we did not include the complete specification of the case, we have found that the formalism is expressive enough to render the case properly. Although the number of transactions and speech acts becomes high, one important advantage of our approach is that the case can be viewed at different levels of abstraction. For example, if we suppose that all transactions follow the typical actagenic or factagenic pattern, we don't have to define each of them anymore. And for a global overview, a representation of the contracts, as in figure 2, is already sufficient.

By using the L/A patterns of communication, the first bunch of control weaknesses in this case as discussed by Bons, is automatically resolved. For example, one control weakness is that the sea-carrier should confirm the transport instruction from the shipper using a transport confirmation document. In an actagenic transaction, it is taken for granted that the obligation is only effective when the executor has accepted the request by means of an accept or commit speech act.

The few remaining control weaknesses have to do with the lack of trust between parties. In particular, the sea-carrier and consignee do not trust each other. Therefore, a problem arises: should the goods be delivered before the consignee confirms the receipt or after. In the first case, the sea-carrier runs a risk, in the latter case the consignee. A possible solution is that transfer of goods and confirmation occur simultaneously. In our model of the case, the exchange of the goods takes place in a factagenic conversation between sea-carrier and consignee. This transaction only succeeds if two speech acts have been performed: a speech act from the sea-carrier "Hereby I have transferred to you the goods" and a speech act from the consignee" Hereby I accept the transfer". So the required simultaneity is already guaranteed by the fact that the transaction can not succeed with only one message. So the fact that we have units of larger granularity than a single message immediately solves the problem.

As we already hinted at above, there is an alternative way of looking at this issue, one in which we get more precise about the definition of "transfer goods". The transfer of goods is an activity that consists of several actions, in particular (relevant for this case) the shipping (putting the load on the ship), the delivery of the goods at the receiver's estate and the receiving of the goods. The distinction between delivering goods and receiving them is similar to the distinction between requesting and accepting a request: the underlying idea is that delivering goods is a social action that is about transfer of ownership (in an unspecified sense) rather than just movement in space. If we take this more fine-grained perspective, then the contract between shipper and sea-carrier is aimed at shipping only, and so is not deficient: the sea-carrier can confirm to the shipper that the goods have been shipped and the shipper can accept this (or not). At the time of delivery, a contractual interaction occurs between sea-carrier and consignee in which the sea-carrier request the consignee to accept the goods and the consignee requests the sea-carrier to deliver the goods. One could think of this concretely as the telephone call that occurs when the goods have arrived in the country of the consignee and an appointment is made when and where the goods will be handed over. Then in the result phase, the sea-carrier must confirm that the goods have been received and the consignee must confirm that the goods have been delivered. If the parties do not trust each other, constraints must be specified to the effect that the confirmations occur simultaneously.

The contract between consignee and sea-carrier is a bit different from the other ones in the sense that both parties are obliged beforehand to enter the interaction. The consignee is obliged to accept the goods (unless the delivery does not conform to the original order) and the sea-carrier is obliged to deliver the goods (as a consequence of his accepting the shipping order). This also implies that both parties are *authorised* to request for an appointment

for handing over the goods. So we still maintain the intuition that the interaction between sea-carrier and consignee is a bit different from the other interactions, even if we represent if formally by means of a contract. Note that this refinement (but in fact, the same holds in our original model) highlights the obligation of the consignee to accept the goods: in fact, this obligation is not automatic, but is quite natural as a condition in the contract between consignee and shipper. The Dynamic Deontic Logic that we use allows the specification of such "secondary" obligations.

The final control weakness discussed by Bons has to do with the involvement of a trusted third party (the Inspection Agency) to witness for some fact in the case that parties do not trust each other completely. We could work this out a special kind of factagenic conversation with three instead of two participants in which the TTP asserts the fact and the other two parties accept this. Alternatively, it could be modelled as a separate contract between e.g. shipper and Inspection Agence that starts to run when the shipper has to accept a certain statement of the sea-carrier and awaits the confirmation of the Inspection Agency before it does so.

The case study has shown that our approach is suitable for modelling Electronic Commerce transactions. While abstracting from the form (e.g. physical or electronic documents), it highlights what is essential: the obligations of the parties and the establishment of facts. The workflow loops and contracts also make immediately clear what is the function of each message in the whole process. We see this as an important advantage over for instance the Documentary Petri Nets used by Bons in which one can easily forget a speech act without noticing the gap. The gap will become clear when the integrity rules are applied, but in our approach they are already highlighted during the modelling itself.

Although the modelling approach that we have used is suitable and also very helpful for designing secure protocols, we must keep in mind that the relationship with the material world is not taken into account yet. If we want to ensure this relationship, we must also model the physical object streams and the acts of witnessing that mediate between the material and social world. We intend to work out this part of the audit check later.

Finally, we want to come back to the use of design patterns. The design of a procedure from scratch can be quite complicated, even if different levels of abstraction are used. However, it is not necessary to reinvent the wheel every time again. Patterns of transactions, contracts, scenarios can be stored in a component library and taken in toto when a new interaction is set up. In a current ESPRIT project (MEMO), we are developing such a component library as part of a Trusted Third Party service.

8. References

[austin62] Austin, J., "How to do things with words", Clarendon Press, 1962

[bons97] Bons, R., "Designing Trustworthy Trade Procedures", Phd Thesis, Rotterdam School of Management, 1997

[busch97] Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., and Stal, M., "Pattern-Oriented Software Architecture", John Wiley & Sons, 1996

[dietz96] Dietz, J.L.G., Rijst, N.B.J. van der, and Stollman, F.L.H., "The specification and implementation of a DEMO supporting CASE-tool, In: Workshop on Communication Modeling – The Language/Action Perspective, eds.: Dignum, F., Dietz, J., Verharen, E., and Weigand, H., Computer Science Reports, Eindhoven University of Technology, http://www.win.tue.nl/win/cs

[dietz96a] Dietz, J.L.G., "Introductie tot DEMO: Van Informatietechnologie naar organisatietechnologie", Samson Bedrijfsinformatie, Alphen aan den Rijn/Zaventem, 1996

[dietz96b] Dietz, J.L.G., "Dynamic Enterprise Modeling Basics", in: Dynamic Enterprise Modeling: A Paradigm Shift in Software Implementation", Kluwer Bedrijfsinformatie, Deventer, 1996

[fowler97] Fowler, M., "Analysis Patterns: Reusable Object Models", Reading, MA, Addison-Wesley, 1997

[gamma94] Gamma, E.R., Helm, R., Johnson, R., and Vlissides, "Design Patterns: Elements of Reusable Object-Oriented Software", Reading, MA, Addison-Wesley, 1995

[goldk96] Goldkuhl, G., "Generic business frameworks and action modelling", In: Workshop on Communication Modeling, eds.: Verharen, E., Rijst, N. van der and Dietz, J., Springer, Electronic Workshops in Computing, htp://www.spinger.co.uk/WiC/Workshops/CM96.hmtl

[hammer93] Hammer, M., Champy, J.A., ”Reengineering the Cooperation: A Manifesto for Business Revolution”, Nicholas Brealy, London, 1993

[heuvel97] Heuvel, W.J. van den, Weigand, H., ”Ensuring the Vailidity of Electronic Commerce Communication”, In: Workshop on Communication Modeling – The Language/Action Perspective, eds.: Dignum, F., Dietz, J., Verharen, E., and Weigand, H., Computer Science Reports, Eindhoven University of Technology, http://www.win.tue.nl/win/cs

[kal96] Kalakota, R., and Whinston, A.B., “Frontiers of Electronic Commerce”, Reading, MA, Addison-Wesley, 1996

[kim96] Kimbrough, S. and Lee, R., ”On formal aspects of electronic commerce: examples of research issues and challenges”, In: Proceedings HICSS’96, IEEE Computer Society Press, 1996

[kim97] Kimbrough, S. and Moore, S.A., ”On automated message processing in electronic commerce and work support systems: Speech Act”, ACM Transactions on Information Systems (TOIS), 1997

[med92] Medina-Mora, R., Winograd, T., Flores, R., and, Flores, F., “The Action Workflow Approach to Workflow Management Technology”, in: Proceedings of 4th Conference On Computer Supported Cooperative Work (CSCW’92), ACM Press, 1992

[reijs96] Reijswoud, V.E. van, “The structure of Business Communication: Theory, Model and Application”, PhD Theses, Delft University of Technology, Delft, 1996

[rosch96] Roscheisen, M., ”A communication agreement framework for access/action control”. Technical Report, Stanford University, 1996

[schäl96] Schäl, T., ”Workflow Management for Process Organisations”, LNCS 1096, Springer Verlag, Berlin, 1996

[searle69] Searle, J., ”An essay in the philosophy of language”, Cambridge University Press, 1969

[taylor93] Taylor, J.R., ”Rethinking the Theory of Organizational Communication”, Ablex Publishing Company, 1993

[verharen97] Verharen, E.M., “A Language-Action Perspective on the Design of Cooperative Agents”, Phd Thesis, Tilburg University, 1997

[weig98] Weigand, H. and Heuvel, W.J. van den, ”Meta-Patterns for Electronic Commerce Transaction based on FLBC”, in: Proceedings of Hawaii International Conference on Systems Sciences, 1998

[weigand98a] Weigand, H. and Heuvel, W.J., ”Trust in Electronic Commerce – a Language/Action Perspective”, submitted to the Workshop on Deception, Fraud and Trust in Agent Societies, Minneapolis/St Paul, 1998

[wino86] Winograd, T., ”A language/action perspective on the design of cooperative work”, In: Greif, I., editor, Computer Supported Cooperative Work: A Book of Readings, Morgan Kaufmann, 1986