An Approach to the Construction of Flexible B2B E-Contracting Processes

Samuil Angelov, Paul Grefen
{sangelov, grefen}@cs.utwente.nl

Computer Science Dept.
University of Twente
The Netherlands

Abstract

In business-to-business relations, contracts serve both as a protection mechanism of trading parties, as well as a prescription document for activities to be executed by them. The processes of contract establishment and its enactment are often expensive and time-consuming. E-contracting aims at improving the efficiency and effectiveness of the contracting process. Additionally, it offers new opportunities to the contracting parties, e.g. micro-contracting. For the design of an information system for support of e-contracting, a clear description of the e-contracting processes is required. In this report, we introduce a process model for flexible business-to-business e-contracting. To separate concerns, we distinguish structured function and communication perspectives of e-contracting processes supplemented with consistency rules. Using the defined model we propose an approach for elaboration of concrete e-contracting process specifications and their implementation in real-life business scenarios. The approach is a basis for complete functional e-contracting process specifications, with coherent communication between contracting parties and coupling between the internal and external activities of a company.

Table of Contents

1 Introduction

Business process modelling aims at improving the efficiency and effectiveness of business processes executed in a company. Software applications with different levels of complexity are provided for the coordination and automation of intra-organizational processes. Until recently, due to the level of development of information technology, business process modelling and supporting software applications were limited to the scope of one company, i.e., the intra-organizational processes. After the rapid development of information technology, the possibility for the support of cross-organizational business processes emerged [Gre00], [Alo99], [Cro00]. Many research institutes and standardization efforts, e.g. ebXML [ebX01a] and RosettaNet [Ros01], work on the problem of modelling cross-organizational processes and realization of supporting information systems.

Service outsourcing improves the efficiency and effectiveness of companies. In the service outsourcing paradigm, a service consumer outsources part of its non-core business processes to a service supplier. The service consumer and service supplier form a virtual enterprise that appears as a single entity to third parties. Contracts contain the specification of the outsourced service and are a basis for collaboration between the parties. Standard paper contracting is often slow and requires involvement of human actors in all contracting phases. Electronic contracting is faster and cheaper than standard paper contracting and offers new opportunities to the partners, e.g., micro-contracting [Gre02b]. For the support of dynamic service outsourcing, efficient and flexible e-contracting is required.

Business-to-business e-contracting comprises a collection of coherent intra- and cross-organizational activities. As the level of information technology allowed, attempts to realize information systems for the support of contracting processes were made [diC02], [Cro00], [SeC99], [Gri98], [Dan98], [Ora01]. These efforts did not deliver a complete and context-independent solution. The reason for this is the complexity of the contracting process. This complexity is caused by the variety of contexts in which e-contracting is to be performed, which leads to a variety of performed activities and their order of execution. However, implementation of information technology for the support of business-to-business electronic contracting requires a precise description of the activities that are to be performed by the participating companies. In this report, we present an approach to achieve flexible and precise business-to-business e-contracting process specifications.

We describe a model of the business-to-business e-contracting process. The model consists of two perspectives, i.e., function and communication perspectives. We model the cross-organizational activities separately from the complete set of e-contracting activities, in order to achieve coherence of communication between parties. By using process decomposition and inheritance in the function view and process inheritance in the communication view, we aim to achieve a structured and complete e-contracting process description. The two perspectives do not suffice for composing e-contracting processes, as they do not specify sequence, mutual exclusion, etc. of activities. For this reason, we define a set of consistency rules for the description of the order and existence relations between the e-contracting activities.

We use the function and communication perspectives, in combination with the consistency rules to define an approach for the flexible, context-specific construction of concrete e-contracting process specifications. The model can be used for analysis of existing e-contracting process specifications as well. Requirements that are not satisfied or inconsistencies in existing specifications can be discovered. A software architecture for realization of e-contracting systems still does not exist. The elaboration of this software architecture demands precise description of the e-contracting process. In this respect, the model proposed in this report is the basis for obtaining requirements for an e-contracting system. In the course of preparation of this report, we examined leading industrial solutions in the field of e-contracting [diC02], [ebX01a], [Ora01], [Pon02], [Ros01], [SAP01]. Though these solutions do not deliver a complete solution for e-contracting, they indicate requirements for the e-contracting process. For example, monitoring and control support during contract enactment is an issue that is still not supported to its real potential, but is addressed in [diC02] and [Ora01]. For this reason, requirements on e-contracting systems obtained from our model will reflect the requirements on e-contracting systems of the business world as well.

This report is organised as follows. In Section 2, we discuss the proposed model. In Section 3, we take a closer look at the function perspective of our model. Section 4 provides details on the communication perspective. For the lowest level of decomposition of the function perspective, we use the communication perspective. For this reason, we elaborate this level of the function perspective in Section 5, following the description of the communication perspective. In Section 6, we describe the consistency rules used to define relations between activities (Appendix A lists the minimum set of consistency rules applicable to all e-contracting process specifications). A description of the process of construction and use in practice of the e-contracting process specifications is presented in Section 7. To illustrate the application of our approach, in Section 8, we describe the construction of a fragment of a concrete e-contracting process specification based on an imaginary business scenario. We provide the complete process specification in Appendix B. In Section 9, related work in this domain is discussed. The report ends with conclusions.

2 E-contracting process model

In the first part of this section, we provide a general description of our e-contracting process model. We briefly present the perspectives in the model and the relations between them. Next, we position the e-contracting process model in existing business modelling and requirements modelling frameworks.

2.1 E-contracting process model description

To provide a complete functional model of the e-contracting process with consistent communications between parties, we separate concerns by elaborating two perspectives of the process.

An e-contracting process contains various activities. Some activities can be vital for specific business domains and completely irrelevant for other domains. Even activities relevant for a domain can become unnecessary in a specific context. This indicates that special attention must be paid to the completeness of the e-contracting process model that we shall define in this report. To achieve completeness, we elaborate a function perspective of e-contracting activities. In this perspective, e-contracting activities are decomposed at several levels of abstraction. The function perspective provides a complete picture of the e-contracting activities to a certain level of decomposition. Its hierarchical presentation allows further decomposition for the support of specific context requirements. Specific issues of a business domain (such as the insurance domain) can be addressed in this way.

E-contracting is a blend of intra- and cross-organizational activities. An e-contracting model should guarantee consistency of cross-organizational activities as well as coherence between the cross- and intra-organizational activities. To achieve this, in addition to the function perspective we elaborate a communication perspective. The function perspective aims at complete description of the activities in the e-contracting process to a certain level of detail. For this reason, we use a decomposition technique at the three levels of abstraction. In the communication perspective, we want to describe the different communication activities that can exist in an e-contracting process. For this reason, the communication perspective is a specialization tree of one root communication activity. The communication hierarchy is coupled with the internal processes that relate to the communication activities. This allows modelling of coherent cross- and intra- organizational process.

The activity leaves in the communication perspective are a subset of the activity leaves in the function perspective. The communication perspective aims at facilitating the process of defining consistent communication activities in the function perspective. We use it in the construction of the last level of the function perspective where concrete communication activities are identified (the function perspective and its levels are described in detail in Section 3 and Section 5). The role of the communication perspective is not limited to the construction of the function perspective. As we show in Section 8, any subsequent specialization of leaves in the function perspective might uncover new communication activities. In these situations, the communication perspective is used by a process designer for the specialization of the leaf activities to communication activities. For this reason, the communication perspective is an integral part of the proposed model and approach. In combination with the consistency rules (explained in Section 6), the modeller can achieve specification of consistent communication leaf activities.

To model the relations between the e-contracting activities, we define a set of consistency rules. These rules impose constraints on the construction and execution of concrete e-contracting processes, preventing process designers from the usage of incorrect activities or omission of activities from the function perspective, and preventing the execution of activities in incorrect situations. A list of predefined rules that serve this purpose is provided in Appendix A. A process designer is free to apply additional rules that are needed for the specific business situation. The consistency rules can be applied at the different levels of abstraction of the function perspective. For some of the rule types, rule inheritance is applicable. In this way, rules can be defined in a top-down fashion. This simplifies the problem with the increasing complexity for defining consistency rules in the lower levels of specialization caused by the increasing number of activities. Process designers can use rule inheritance in situations where further specialization of the function perspective is required. They can inherit rules defined in the higher level of abstraction and define only additional rules.

The two perspectives, in combination with the consistency rules, provide a tool for a business party to construct its own flexible e-contracting process specifications. The model allows coherent communications between parties to be achieved.

In this section, we have explained briefly the role of the two perspectives and the consistency rules in our e-contracting process model. In Sections 3, 4, 5 and 6, we elaborate each of the three elements of the model. We start with the function perspective, as it represents the process decomposition and is fundamental for the e-contracting process model. In the next subsection, we pay attention to the position of our model with respect to existing business-modelling and requirements modelling frameworks.

2.2 Relation to existing frameworks

E-contracting can be viewed as a business process that is performed by two or more business partners at the same time. In this respect, we can position our model in commonly accepted frameworks for business modelling. Requirements modelling captures requirements on information systems and the environment analysis is a source for system requirements. In this report, we present a model of the e-contracting process, which is used in the e-contracting system environment analysis. For this reason, we can position our model in frameworks for requirements modelling as well.

In [Eri00], four views on a business are depicted, i.e. business vision, business process, business structure, and business behaviour view. The functional perspective in our model is a decomposition of the e-contracting process to a certain level of abstraction. Thus, it is part of the process view described in [Eri00]. The function perspective does not provide the complete picture of the process view. It represents only activities related to e-contracting. In our model, we do not discuss the resources involved in the different activities, as this is not of importance for the proposed approach. The communication perspective and the consistency rules that we describe impose constraints on the e-contracting activities. For this reason, they can be regarded as part of the behaviour view described in [Eri00]. The business vision view describes the goal structure of a company. Primary business processes allow business goals to be achieved. E-contracting is an important secondary business process that supports the execution of the primary business processes. That is why our model does not contribute directly to the business vision view. The business structure view describes the resources in a business or the structure of the created products. This is out of the scope of our goals in this report.

In [Oll88], a framework for requirements models is described. Three views are identified, i.e., process, behavioural and data view. Using the same way of reasoning, we can position our function perspective in the process view, and the communication perspective together with the consistency rules in the behavioural view. As we have already mentioned the data view is not in the scope of this paper. Analogous comparison can be done for other frameworks as well, e.g., [Roe96].

3 Function perspective

In this section, the structure of the function perspective of the e-contracting process model is described. The perspective is presented as a combination of a subtyping hierarchy and composition associations. We distinguish three levels of abstraction in the hierarchy, i.e., phase, abstract activity, and concrete activity levels. We start with a description of the first level of abstraction and with the overall structure of the function perspective. For reasons that are explained at the end of this section and in

Sections 4, we describe the specializations of abstract activities and the decomposition of these specializations to concrete activities (third level of abstraction) in Section 5.

3.1 Phase level

An e-contracting process consists of a number of e-contracting phases. These phases are executed successively in time. As in [Gis00], we distinguish four phases: information, pre-contractual, contracting and enactment phases (see Figure 1 – phase level). In the information phase, general preparations are made, information is provided (for a request or offer of services) and possible partners are identified. In the pre-contracting phase, activities that aim at determining if negotiations with the other party should be initiated are performed. In the contracting phase, the contract is negotiated and established. During the enactment phase, the contract is executed, accompanying activities are performed, and finally the contract execution is evaluated. For a successful e-contracting process, at least the last two phases must take place [Ang02] (this requirement can be defined using the consistency rules described in Section 6).

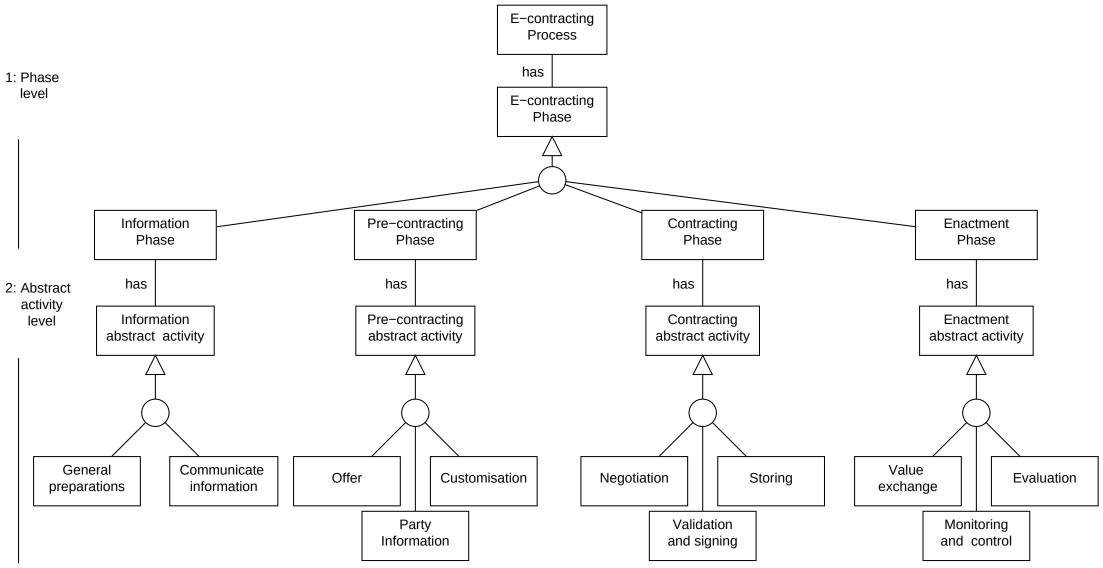


Figure 1 – Function perspective (levels one and two)

3.2 Abstract activity level

Each phase contains different abstract activities. For example, the *pre-contracting phase* is decomposed to *pre-contracting abstract activities*. Abstract activities form the second level of abstraction. Abstract activities are collections of concrete activities. A collection of concrete activities is a logical unit that delivers value to the parties. Abstract activities are specialized according to the concrete activities performed in them (process aspect) and the data associated with these concrete activities (data aspect). For example, the *pre-contracting abstract activity* is specialized to *offer, party information*, and *customisation* abstract activities (see Figure 1). In the *offer* abstract activity, activities that support the exchange of an offer are performed. Analogously, we can reason for the rest of the specializations of abstract activities. Concrete activities form the third level of abstraction of the function perspective. The level of detail in the third level of specialization allows communication activities, i.e., activities that send or receive messages among companies to be distinguished from internal activities. To specialize the communication activities in the third level of the function perspective, we use the communication perspective (see Section 4). For this reason, we explain the abstract activities for each phase, their decomposition into concrete activities, and the different specializations of the concrete activities in Section 5.

4 Communication perspective

In Section 3, we have described three levels of abstraction of the e-contracting activities. At the concrete activity level, activities can be distinguished as intra- or cross-organizational. The specializations of abstract activities to cross-organizational activities (i.e., communication activities) have to be consistent in order to guarantee coherence of the communication activities of parties and thus to guarantee successful exchange of information between parties. Additionally, coherence between the communication activities and the internal activities associated to them is required. To facilitate the specialization of abstract activities in the function view to concrete cross-organizational activities, a communication perspective is required. The communication perspective helps parties to identify the communication activities that will be performed by them. Further on, it standardizes the specifications of the communication activities. Finally, it allows parties to link their communication and internal activities. In this section, we describe the communication perspective of the e-contracting model.

4.1 Structure of communication perspective

Communication activities are specialized from a root *communication activity*. We specialize the communication activity into *request*, *respond* and *inform* activities (see Figure 3). A *request* activity is an activity that is performed when a party requires information from another party. A *respond* activity is an answer to a request event. The request event is generated by an internal activity or by an external *request* activity. An *inform* activity is an activity that delivers a single information message from one to the other party. It is initiated by an internally generated event. The inform activity can play two roles. It can be an activity that acts as closing activity of a request-response conversation (a response can generate a counter response), e.g. an acceptance or rejection message. Also, it can be a single notification message to the other party, with no direct relation to previous communication activities.

This set of communication activity types covers all aspects of a communication process[1]. The defined communication activity tree, with this level of specialization,

[1] Well-known frameworks, e.g., [WSD01] and [Put01], identify the same communication activity types.

suffices to represent the cross-organizational perspective. It defines the general communication activities. All concrete communication activities can be specialized from the *request*, *respond* and *inform* activities.

To achieve coherence between the intra- and cross-organizational activities, we add to this perspective the internal activities that are directly connected to the communication activities. Such an activity is, for example, *response processing* (i.e. when a party receives a response from another party). By adding intra-organizational activities in this view, communication activities and internal activities associated to them are linked to each other. In Figure 3, internal activities are positioned below the communication hierarchy. Internal activities cause an intermediate level of specialization of communication activities to be defined. We specialize the communication activities according to their relation with the internal activities. A request activity is specialized into requests that expect a single response and requests that expect multiple responses. The internal activity, according to which they are specialized, is the response processing activity. For example, the request for offer expects only one response, while a request for information, published in a market place, expects many responses from different parties. Analogously, the *respond* activity is specialized to a required and optional response. Only the response to a request for information (published in a market place) is optional. For all other response activities, the requesting party requires response. The *inform* activity is specialized to three subactivities, i.e., *accept, reject*, and *notify* activities. They send respectively an acceptance, rejection or notification message.

Using the defined communication perspective, we specialize the decompositions of the communication abstract activities in the function perspective (explained in Section 5). In this way, all cross-organizational activities that are identified as concrete activities in the function perspective are leaf activities in the communication perspective (as we have already mentioned in Section 2). For completeness, we show in Figure 3 all concrete communication activities that are identified later on in Section 5.

4.2 Relations between communication activities

An abstract communication activity from the function perspective is decomposed and is further on specialized into activities that are at the same time specializations of the already defined general communication activities in Section 4.1 (i.e., specializations of the requesting, responding, informing, and processing activities). However, this is not sufficient to achieve coherence in the communication between the contracting parties. To show explicitly the possible patterns for the execution of communication activities, we must define the relations and constraints between them. In this section, we define the relations and the cardinality constraints that exist between the requesting, responding, informing, and processing activities. In order not to overload Figure 3, we show the relations and cardinality constraints between communication activities separately in Figure 2.

A *request* activity is associated with one or more responses (see Figure 2). A *respond* activity can be triggered by a request or by an internal event (thus zero or one requests). Also, the *respond* activity is associated with exactly one *response processing* activity. The *response processing* activity can lead to zero or one (counter) responses, or zero or one *inform* activities.

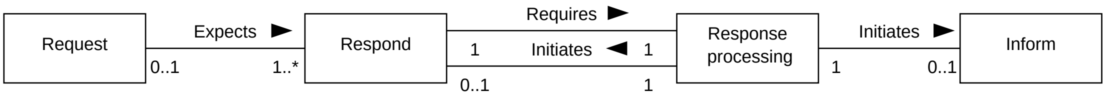


Figure 2 – Relations between communication activities

Until now, we have shown how the associations between communication activities are defined and we have applied cardinality constraints over them. In Section 6, we define a set of consistency rules that can be applied over the e-contracting activities to define temporal and existence constraints. We use these rules to define additional constraints over the communication activities.

4.3 Task inheritance in the communication perspective

To facilitate the construction of concrete communication activities, we exploit the specialization aspect of the constructed hierarchy. Process specialization techniques are described in more detail in [Bus99]. In the communication perspective, we use the fact that communication activities consist of similar tasks. For example, every communication activity consists of at least two tasks, i.e., *elaborate message* and *send message*. These two tasks belong to the root *communication activity*. In the communication hierarchy, subactivities inherit the tasks of their super-activities. The complete set of tasks of a subactivity is constituted from the inherited tasks and the added tasks that are specific for the subactivity. Thus, the *request*, *respond* and *inform* activities inherit *elaborate message* and *send message* tasks from the root *communication activity*. For the *request* activity, we add a *wait* state, as each request waits for a response and a triggering event for this state (*response received* or *timeout*). The inheritance of tasks from super-activities to subactivities in the communication perspective facilitates the construction of concrete specifications of leaf communication activities. The complete set of tasks executed in an activity results from the automatically inherited tasks from its super-activity and the identified tasks that are specific only for this activity. We illustrate this with an example in Section 8.

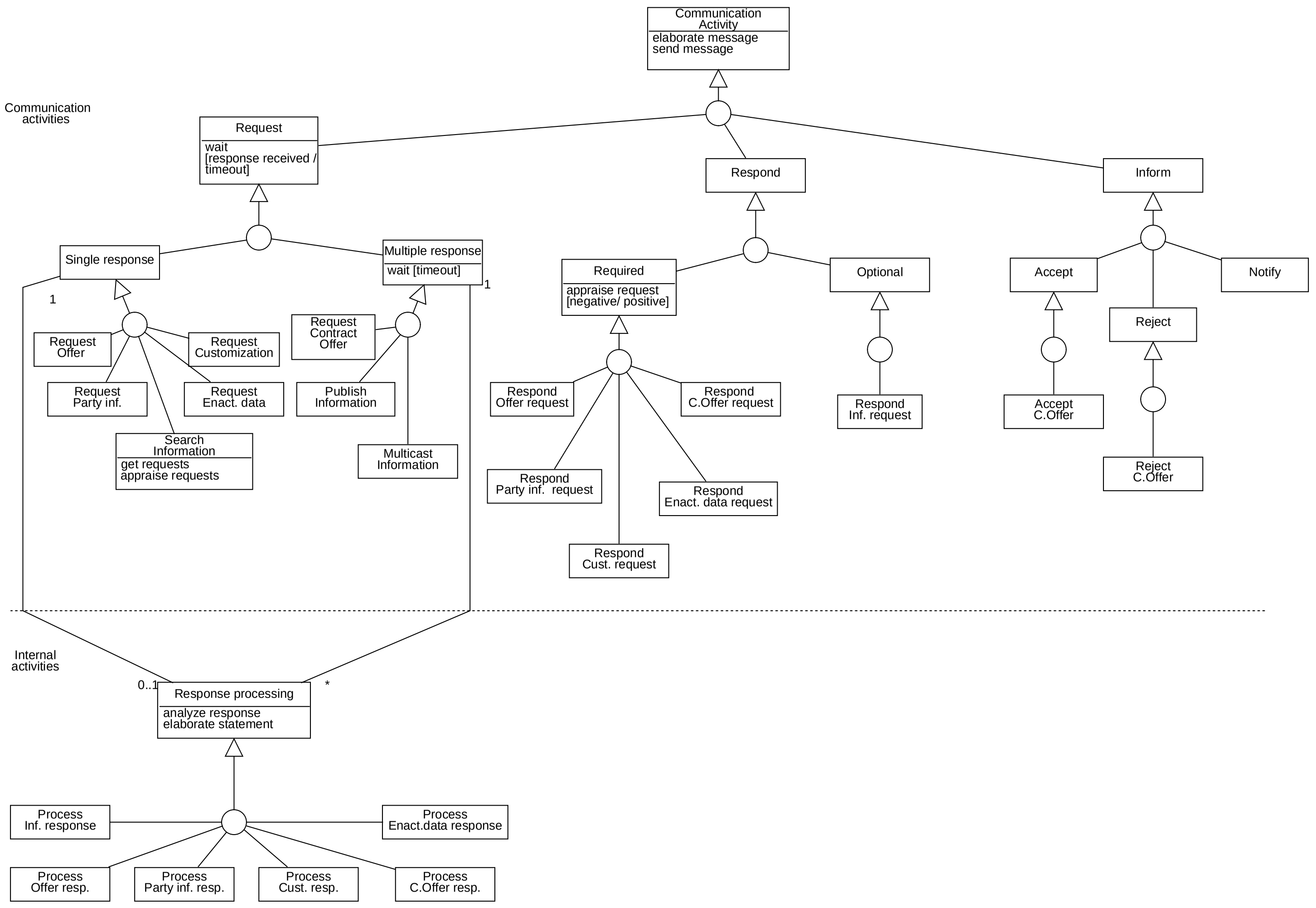


Figure 3 - Communication perspective

5 Concrete activity level of the function perspective

In Section 4, we have described the communication perspective. As we have already explained, we use the communication perspective to construct the specialization of concrete communication activities in the function perspective (see Section 3.2). In this section, we explain the different specializations of abstract activities and their decomposition and further specialization to concrete activities. At the end, we discuss further decomposition and specialization of concrete activities to lower levels of abstraction.

5.1 Abstract and concrete activities for the information phase

The information phase has two abstract activities, i.e., the general preparations and communicate information activities (see Figure 4). Next, we describe each of them and their decomposition to concrete activities.

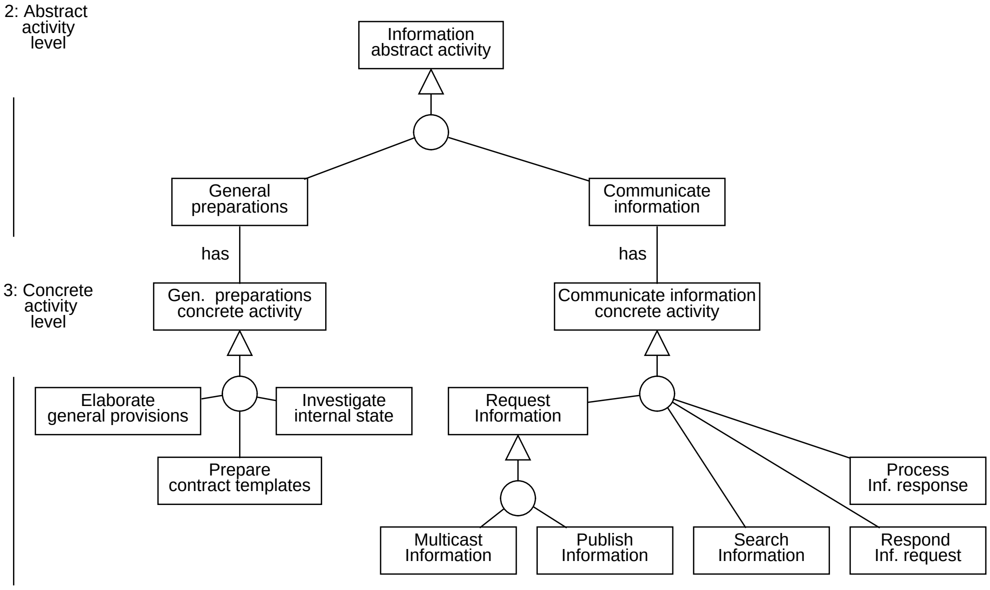

2: Abstract activity level
Information abstract activity
General preparations
Communicate information
has
has
3: Concrete activity level
Gen. preparations concrete activity
Communicate information concrete activity
Elaborate general provisions
Investigate internal state
Prepare contract templates
Request Information
Process Inf. response
Multicast Information
Publish Information
Search Information
Respond Inf. request

Figure 4 – Abstract and concrete activities (information phase)

General preparations: Parties prepare for the e-contracting process. This includes *elaboration of general provisions*, *preparation of contract templates*, *investigation of the internal state* of a company, etc. Internal business processes (i.e., processes that do not involve communication between parties) can significantly vary among companies. The set of concrete activities that we provide is not complete and is used only as an illustration of possible concrete activities.

Communicate information: In this stage, parties exchange information, searching for suitable business partners. For a party, the successful result of this stage is a collection of companies that match its searching criteria. The *communicate information concrete activity* is specialized to five concrete activities. A party sends a request for information. The request for information can be either an advertisement from a supplier looking for buyers, or a request for a product (service) from a consumer. We consider advertising as a request for information because responses from interested companies can follow each advertisement (see the request-response pattern explained in Section 4). The request can be sent directly to other parties, i.e., *multicast information* (e.g. if previous trade relations existed between these parties in the past) or through an intermediary market place, i.e., *publish information*. Both concrete activities expect for each request arbitrary number of responses. Another party might receive directly a request or might initially *search information* in a market place. Depending on the evaluation of the requests, the company might *respond* to requests. The requesting company *processes* and evaluates every received response.

5.2 Abstract and concrete activities for the pre-contracting phase

The pre-contracting phase has three abstract activities, i.e., *offer, party information* and *customisation* (see Figure 5).

Offer: In order to get provisional contract information, consumers can request offers from suppliers. An offer is a document that provides information on the operational aspects of the exchange, and is accompanied by the general provisions of the company. Consumers make a *request* for an offer. Suppliers might *respond* to the request with an offer. The response is *processed* and evaluated by the consumer.

Party information: More information about a company might be required at any stage after the matching companies are identified. In this activity, parties collect additional information about other companies. The information might be evaluation on previous contract activities, official information on the company state, digital certificates, etc. This information can be collected directly from the company or through a Trusted Third Party (TTP) [Kee00]. TTPs that can provide such information are for example the national chambers of commerce. A company sends a *request for party information*. A company or a TTP *responds* to the request. The requesting company *processes* the response.

Customisation: After an offer is exchanged between companies, the consumer might request customisation of the offer. A *request* for customisation is sent to the supplier. The supplier *responds* to the request and the consumer *processes* the response. This initial negotiation is intended only to help in determining whether the two parties should start contract negotiations.

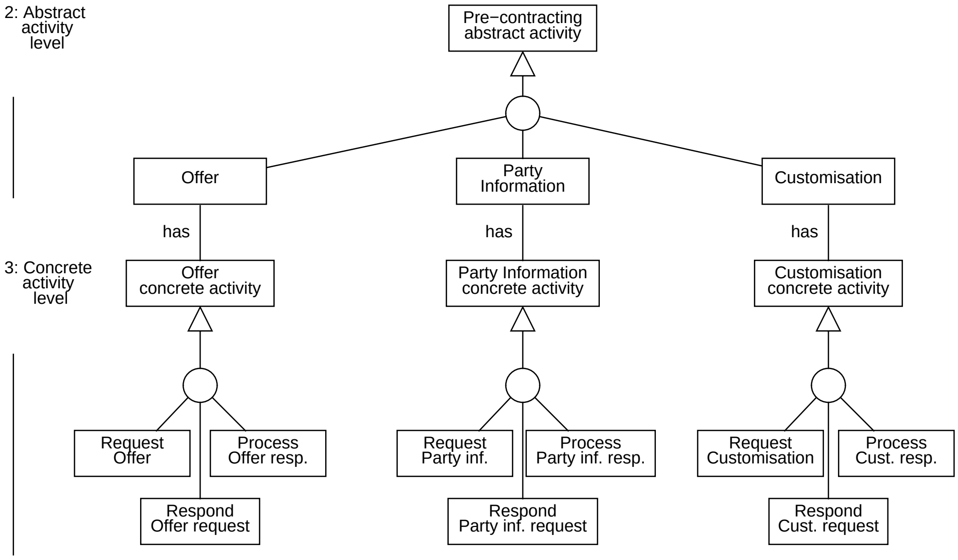


Figure 5 – Abstract and concrete activities (pre-contracting phase)

5.3 Abstract and concrete activities for the contracting phase

In the contracting phase, the major abstract activity is *negotiation* (see Figure 6). If negotiations on a contract lead to agreement, a contract is *validated and signed*. Finally, the contract is *stored.*

Negotiation: The process of negotiation aims at establishing a mutually agreed contract. One of the parties *requests* a contract offer from the other. In most situations, the requesting party will be the consumer. In some cases, however, the requesting party might be the supplier, e.g., when the consumer is a large company that dictates the contracting process. The requested party *responds to the contract offer request*. The response is *processed* and evaluated. The response can be *accepted, rejected*, or might lead to a counter offer response (which is again a *respond contract offer request* activity). If a counter offer is produced, the other party processes the counter offer response and accepts it, rejects it or a produces a new counter offer. These activities are repeated until a contract offer is accepted or rejected, indicating the end of the negotiation process [Gis00], [Sch00].

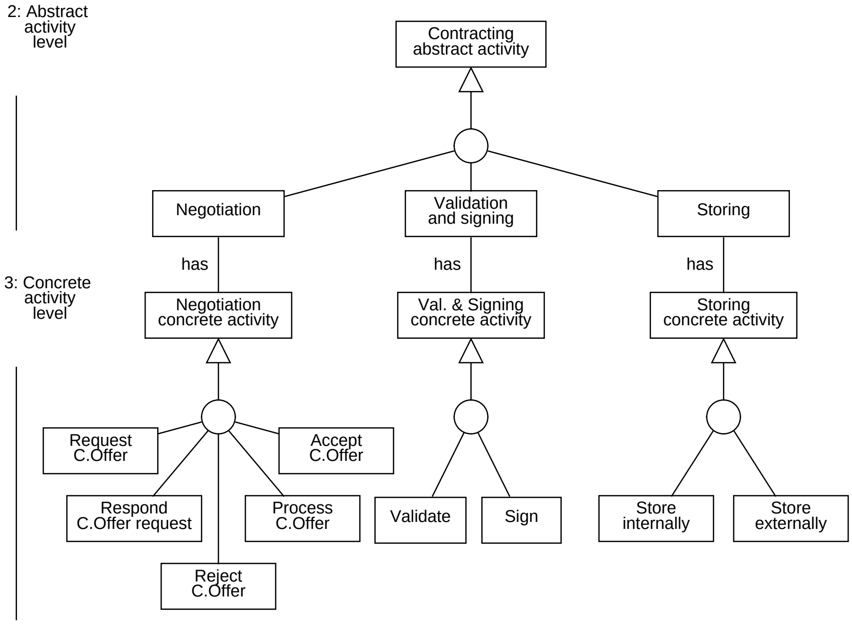


Figure 6 – Abstract and concrete activities (contracting phase)

Validation and signing: When the contract is agreed, the parties *validate* the contract and *sign* it. Its signing includes exchange between the parties of the signed contract. It must be noted that due to security issues, each document that is exchanged will be digitally signed. This digital signature serves two security requirements, i.e., user authentication and document integrity [Tur00]. We do not pay specific attention to this type of signing in the function perspective. During the exchange of contract offers and final contracts it must be indicated that these documents are legally binding [ebX01b]. The creation of a digital signature of a document and the indication that the document is legally binding, we consider as a *sign* activity. The *sign* activity is performed when a legally binding offer or the final contract are exchanged. A possible solution for the realization of the *sign* activity is provided in ebXML.

Storing: In order to keep the signed contract, parties might store it *internally* in the company, *externally* (at a Trusted Third Party) or both.

5.4 Abstract and concrete activities for the enactment phase

In the enactment phase, we identify the *exchange value*, *monitoring and control*, and *evaluation* abstract activities (see Figure 7). The first two activities are tightly coupled as monitoring and control is performed on the exchange of the agreed value and on the conditions that accompany it.

Exchange of value: The supplier performs one or many *service delivery activities*. The consumer *delivers* the promised *reward* in consecutive (one or more) activities. The exchange of values can be preceded or accompanied by the exchange of data that is required for the performance of these activities. Thus, parties can *request*, *respond to a request* and *process data* that is needed for the contract enactment.

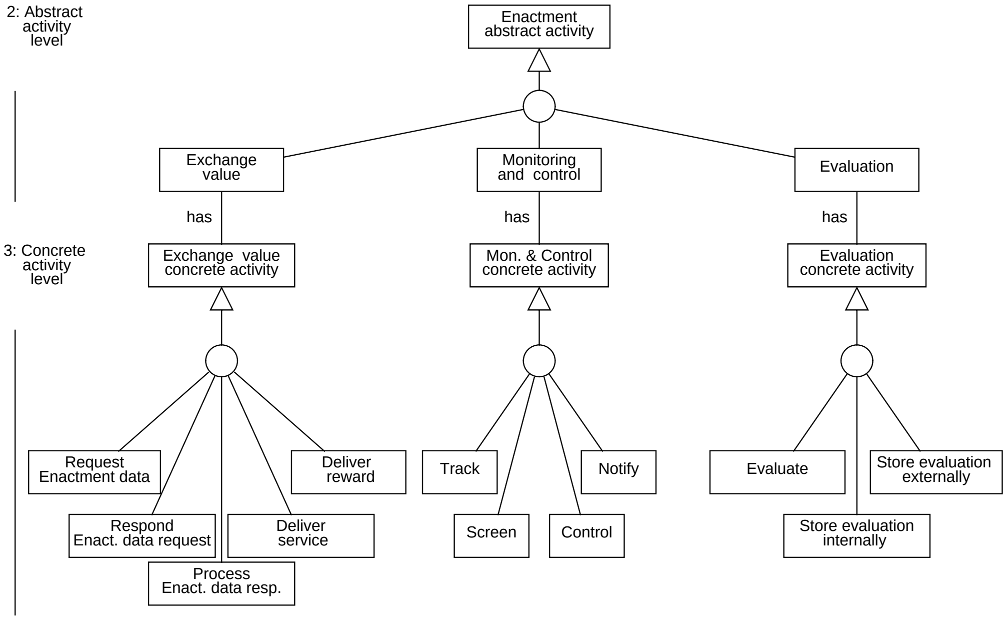

2: Abstract activity level
Enactment abstract activity
Exchange value
Monitoring and control
Evaluation
has
has
has
3: Concrete activity level
Exchange value concrete activity
Mon. & Control concrete activity
Evaluation concrete activity
Request Enactment data
Deliver reward
Respond Enact. data request
Deliver service
Process Enact. data resp.
Track
Notify
Screen
Control
Evaluate
Store evaluation externally
Store evaluation internally

Figure 7 – Abstract and concrete activities (enactment phase)

Monitoring and control: The exchange of values might not always follow the negotiated and agreed process [Coo00] and the contract might not foresee in its content how exceptional situations that appear during enactment can be handled. Also, collaboration or synchronization between parties might be required for the proper contract enactment. To be informed at any moment for the status quo of the contract enactment and to be in a position to control the contract enactment, parties can *monitor and control* the enactment process. As the contract content can be generally separated in two parts, i.e., the description of the exchange of values and the accompanying provisions [Ang01b], we differentiate two different types of monitoring, i.e., contract *tracking*, which monitors the execution of the promised activities and contract *screening*, which monitors the contract provisions [Das97]. At any moment the parties can *notify* each other for successfully completed activities, deviations from the agreed path of execution, etc. In case that some changes to the current execution plan or additional specifications are needed, the parties could impose *control* over the planned contract enactment and change the execution path.

Evaluation: At the end of the contract enactment, the parties can evaluate the contract and its enactment. This evaluation serves to improve future contract activities of the companies, and when being externally shared, to give information to other companies about the contracting potentials of the party. TTPs can perform the collection and maintenance of externally shared evaluations.

5.5 Further decomposition of leaf activities

The activities identified at the concrete activity level are leaves in the function perspective. This level of decomposition is sufficient for modelling standard e-contracting processes. Depending on the business situation, specific activities that are performed during e-contracting might be required. These specific activities are obtained through further decomposition and specialization of leaf activities from the function perspective. In this way, the defined activity tree can be decomposed to new levels of detail, e.g., domain level, company level, service level, etc. It must be noted that only leaf activities from the function perspective can be further decomposed and specialized. This follows from the fact that the function perspective represents complete description of the e-contracting activities at different levels of detail. As a result, each new decomposition and specialization gives a lower level of abstraction. The communication perspective, on the other hand, defines the different types of communication activities. Hence, all communication activities that are identified in the function perspective are specializations of the different types of communication activities in the communication perspective. In Section 8, we provide an example for the decomposition and specialization of the *control* concrete activity driven by the company needs in a concrete business scenario.

6 Consistency rules

In Sections 3, 4, and 5, we have presented the two perspectives of the e-contracting process. The communication perspective describes associations and cardinality constraints between communication activities (see Section 4.2). However, these constraints are applied only for the construction of concrete process specifications. To express temporal and existence constraints over the execution of activities a set of rules is required. In this section, we define a set of consistency rules that are used to model the activity constraints. A textual notation is used for their definition. There are also graphical notations to represent rules for temporal precedence of activities that can be used as well, e.g. [Jac97].

6.1 Description

To investigate the rules that we can apply as constrains to activities, we define a classification tree of the consistency rules (see Figure 8). We distinguish two classes of rules, i.e., applied to a single activity (unary) and applied to two activities (binary). From the unary rules, we consider only the existence consistency rule REQUIRED.

Other unary rules (e.g. FIRST, LAST) are not of importance for our model. A binary rule can be an order, a parallelism, or an existence constraint. A SEQUENCE relation is defined to express required sequence of activities. In order to express parallel execution of activities, the relation DURING is defined. The binary existence constraint EXCLUDE shows mutually exclusion of activities. The IMPLY relation defines execution dependencies between activities.

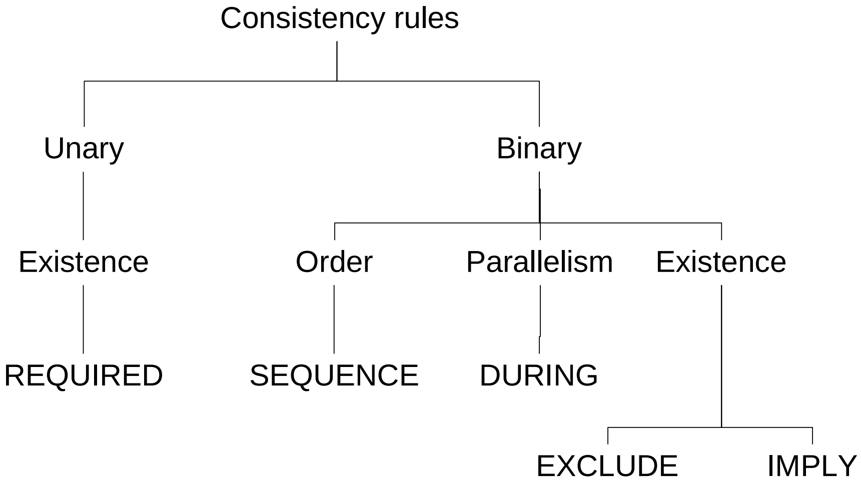


Figure 8 - Consistency rules

Next, we describe in detail each of the consistency rules identified as leaves in the classification tree.

REQUIRED (A_1): The REQUIRED operator indicates that the activity given as an argument must be executed for the successful enactment of an e-contracting process, e.g., REQUIRED (Exchange value).

SEQUENCE (A_1,A_2): The SEQUENCE relation expresses that the execution of A_1 has to precede that of A_2. This relation, however, does not require the execution of the activities. For example, the relation SEQUENCE (Offer, Negotiation) shows that the *Offer* activity hat to be executed before the *Negotiation* activity. The SEQUENCE operator can be extended to define the sequence of more than two activities, e.g., SEQUENCE (A_1, A_2,…A_n). We use extensions of binary rules to n-ary rules only to reduce excessive writing in the examples that follow.

IMPLY (A_2,A_1): The IMPLY (A_2,A_1) relation expresses the requirement that if activity A_2 is executed then activity A_1 is executed as well. Or in other words A_2 implies A_1. An example would be IMPLY (Process offer response, Respond offer request), stating that if a *Process offer response* activity is executed then this implies the execution of the *Respond offer request* activity.

DURING (A_1,A_2): The DURING relation requires activity A_1 to be executed during the execution of A_2. For example, DURING (Service delivery, Monitoring and control) states that during the performance of a *Service delivery* activity, the *Monitoring and control* activity has to be performed in parallel. This operator does not require a simultaneous start or end of A_1 and A_2. The DURING operator can be extended for more than two activities. Thus DURING (A_1, (A_2,…A_n)) states that during the execution of activity A_1, activities A_2,…A_n have to be executed as well.

EXCLUDE (A_1,A_2): This relation shows that both activities are mutually exclusive, e.g., EXCLUDE (Accept contract offer, Reject contract offer). If two groups of activities are mutually exclusive, then we can write EXCLUDE ((A_1, A_2,…A_n), (B_1, B_2,…B_m)). This relation states that each activity A_i excludes the execution of B_1, B_2,…B_m and vice versa each activity B_i excludes the execution of A_1, A_2,…A_n.

This set can be extended with other rule types, if additional constraints must be imposed.

In this section, we have defined a set of consistency rules that are used for definition of constraints on e-contracting processes. In Appendix A, we provide a list of rules (including general rules over leaf communication activities), which is the minimum set of constraints imposed over the activities from the function perspective. This means that these constraints are valid for all e-contracting process specification. Parties can add more constraints over the specified activities, if this is required by the specific situation.

Another approach that can be taken is to use patterns for composition of e-contracting processes [Tut02], [Cas00]. A library of patterns can be defined, specifying different possible patterns for e-contracting. The use of patterns has one major drawback. Patterns can provide a solution (i.e., an e-contracting process specification) for a specific context. However, e-contracting can take place in many different contexts [Ang01b], [Ang02]. This will require a large number of patterns to be defined. To partially overcome this problem, patterns can be defined on different levels of abstraction of the function perspective. Thus, if a party chooses a pattern on the phase level that includes exchange of offers, at a lower level of abstraction, it will choose from a set of "offer exchange patterns". In order to be useful, the defined patterns have to reach the level of concrete e-contracting activities. Otherwise, they will be too general and will not provide coherent communication between parties (as communication activities are distinguished at the concrete activity level).

6.2 Rule inheritance

The hierarchical representation of the function perspective allows rule inheritance for some of the operators to be applied. The definition of rule inheritance is as follows. Let A and B are two activities that are decomposed and further on specialized to subactivities A'_1, A'_2, … A'_n and B'_1, B'_2, …, B'_m respectively (see Figure 9). If a rule defined over activities A and B is in force for each pair of subactivities A'_i and B'_j then we have rule inheritance for this operator. In the case of unary operators, we require that a rule defined over an activity A is in force for each of its subactivities. Next, we show that rule inheritance can be applied for the SEQUENCE, DURING and EXCLUDE operators and cannot be applied for the REQUIRED and IMPLY operators.

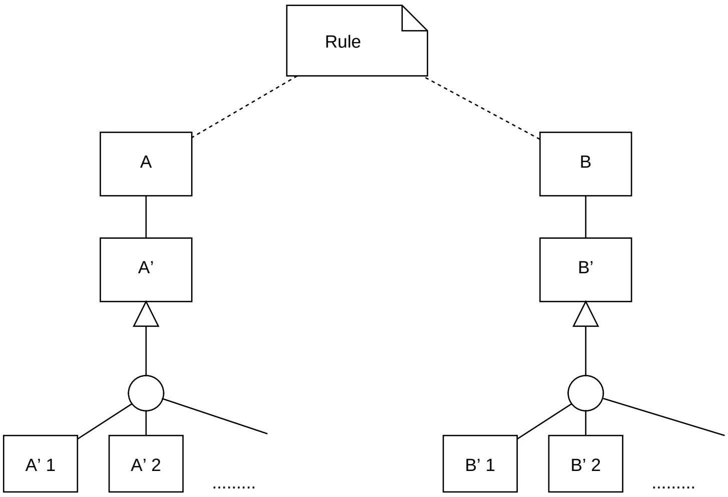


Figure 9 - Rule inheritance

A consistency rule of the type SEQUENCE (A_1, A_2) states that if A_1 is executed, it should precede the execution of A_2. Let B_1 and B_2 are specializations of A_1 and C_1, C_2 are specializations of A_2. Apparently, if B_1 or B_2 is executed, it should precede the execution of C_1 and C_2. Subsequently, we can directly write down SEQUENCE (B_1, C_1), SEQUENCE (B_2, C_1), SEQUENCE (B_1, C_2), and SEQUENCE (B_2, C_2). For example, defining SEQUENCE (Contracting phase, Enactment phase) expresses the order of execution of these two phases. The subactivities of the two phases inherit the defined rule. Thus subactivities of the Contracting phase are executed before subactivities of the Enactment phase. For the DURING and EXCLUDE operators, the line of reasoning is analogous.

It can be easily shown that REQUIRED and IMPLY operators cannot be inherited in this way. For example, REQUIRED (Contracting phase) is a constraint, as no contract can exist without executing the Contracting phase. Signing is a subactivity of the contracting phase and is a required activity. Storing is, however, another subactivity of this phase but is not required in every situation. In this way, we have shown that inheritance of REQUIRED consistency rules is not possible. Analogously, the rule IMPLY (Validation and signing, Negotiation) can be used as an example proving that IMPLY rules cannot be inherited throughout the function perspective.

It might be necessary that leaf activities from the function perspective are further specialized if required by the circumstances (see Section 8 for an example of specialization of concrete activities). Rule inheritance can facilitate the definition of consistency rules over the newly defined leaf activities. However, it can only give part of the required rules for the new activities. The other constraints have to be defined manually, in the specific situation.

7 E-contracting process specification construction and usage

The functional and communication perspectives, in combination with the defined consistency rules, do not define a unique e-contracting process. Business-to-business e-contracting processes vary in the performed activities depending on the business context [Ang01b]. The defined model gives flexibility to construct various concrete e-contracting processes. Consistency rules defined in Appendix A set constraints applicable to all e-contracting situations. Apart from these constraints, a company is free to define and execute the activities according to its own preferences. In this section, we explain the process of the elaboration of business-to-business e-contracting process specifications and their role and usage in business relations.

7.1 Process specification construction

To construct an e-contract specification, a party starts with identification of the activity leaves in the function perspective that will be included in its process definition. To achieve this, the party uses both the function and communication perspectives. The phase and abstract activity levels of the function perspective are used to guide parties to the required branches of the concrete activity level. The communication perspective is used to facilitate the selection of concrete communication activities. The relations defined over the communication activities are used for the selection of the proper set of activities (see Section 4.2). As the identification of leaf activities is a time consuming process, process templates or previously defined specifications might facilitate it. Also, this step might require additional decomposition of some of the leaf activities in the function perspective. When such decomposition reveals cross-organizational activities the communication perspective is used again (see Section 8). Companies make use of a library (see Appendix A) of predefined consistency rules to set constraint on the identified activities. If required by the situation, companies can set additional rules. Then, events that trigger the identified leaf activities have to be specified. These events can be internal, external or both. Internal and external events are explained in more detail in [Chi02]. Finally, process specifications are elaborated for the identified activities. This set of steps for the construction of concrete e-contracting process specifications will require an adequate tool support. We envision a software module for the construction of contracting process specifications that will facilitate the identification of the leaf activities, and will check process specifications for valid communication activities and for conformance with the predefined consistency rules. This module will be part of the design module of an e-contracting management system. It should be able to interface with the e-contracting management system (in order to export the defined specifications) and with the design module of the internal process management system implemented at the party's side (e.g., with a WFMS).

In Section 8, the model is used to construct a fragment of a concrete e-contracting process specification (the complete process specification is provided in Appendix B). We use an imaginary example to show how a company uses both perspectives to identify the leaf activities that are to be included in its e-contracting process specification, how consistency rules are applied and how concrete specifications are defined.

7.2 Process specification usage

Initially, both the consumer and the supplier make publicly available an external view of the defined process specification [Gre01], [Gre02a]. This view hides or generalizes internal activities. It contains communication activities, as well as activities that might be of value for the other parties.

When executing the information phase, a company (a consumer or a supplier) might find an advertisement from a party that matches the requirements of the company. The supported activities and the applied constraints of the two companies are then compared, preferably in an automated manner. The result of this comparison will be a statement whether the two companies can engage in a common e-contracting process (the common e-contracting process starts from the pre-contracting phase or if the pre-contracting phase is skipped directly from the contracting phase). However, it will be often the case that the two specifications do not match completely. This requires changes to one or both of the specifications to be made.

There are two possible situations when changes in the identified leaf activities have to be made. First possibility is when the two parties use standard concrete activities, but in the specification of one of the companies (or in both specifications), activities exist that are not present in the specification of the other party. The second possible situation is when one of the companies (or both of them) has additionally specialised some of the concrete activities from the function perspective. In these two situations, additional efforts are needed to align the two specifications. The alignment is to be done preferably in an automated manner. In both situations, different strategies can be applied for the alignment of the common activities. For example, a company might accept the additionally defined activities by the other party (union alignment); the other company might remove the additionally defined activities (intersection alignment), etc. These strategies will depend on the specific situation. The process of alignment is out of the scope of this report. Further work in this direction is necessary.

When the two process specifications are aligned, companies can start executing their process specifications.

8 Example concrete e-contracting process specification

This section is organised as follows. First, we explain a simple business scenario, which we use to illustrate the approach. Next, using the function and communication perspectives, we describe the process of identification of the leaf activities. To the identified activities, we apply consistency rules. Finally, we describe the construction of concrete process specifications.

Activity diagrams [OMG01] are a common technique for process modelling, used in leading standardization efforts on cross-organizational processes, e.g., ebXML [ebX01a] and RosettaNet [Ros01]. As e-contracting is a blend of intra- and cross-organizational processes, we use activity diagrams as a modelling technique for the construction of concrete process specifications (also advocated in [Gre02a]). In this example, we concentrate on the construction of cross-organizational activities, as they show the application of both perspectives of the model. The complete process specification is given in Appendix B.

8.1 Business scenario

We consider a business scenario, in which a service consumer is looking for a language translation service from a service providing company. The consumer searches for suitable advertisements and finds a collection of translating companies. Next, the consumer asks the selected companies for their offers. Using the information from the collected offers and information collected additionally about the companies, one or more of them are selected to start negotiation with. The negotiation process leads to an establishment of a contract with one of them. At this point, the contract enactment can begin. During the enactment phase, the service provider performs the translation work. The consumer can monitor the work of the translating company and if necessary to request changes in its work.

For reasons of simplicity, we concentrate on the pre-contractual and enactment phases and in the following example, we provide concrete process specification for the *offer* and *control* activities. The complete e-contracting process specification requires identification of leaf activities, definition of consistency rules and elaboration of concrete process specifications for all identified activities (see Appendix B). In this example, we show how these steps are performed only for the *offer* and *control* activities. We choose the *offer* and *control* activities, as they suffice to illustrate our approach. The *offer* abstract activity is a typical communication activity. With this activity, we aim to illustrate the case of identification of leaf activities of an abstract communication activity. The *control* leaf activity is used to illustrate how leaf activities can be further specialized in specific situations.

In the sequel of this example, we take the side of the service consumer and describe the approach from its perspective. Analogous work must be performed at the service supplier side. In Appendix B, we provide complete process specifications for both service consumer and service provider sides.

8.2 Identification of leaf activities

Identification of the activities that will be included in the process specification requires usage of the function and communication perspectives.

Leaf activities for the offer abstract activity

First, the company identifies subactivities of the pre-contractual phase in the function perspective (see Figure 1) and finds the *offer, party information,* and *customisation* abstract activities. The consumer wants to collect offers from many providers, as it will allow it to compare them and to choose the best offer. Therefore, the offer activity is chosen to be included in the process specification. Next step is to select the activity leaves of the offer activity.

The offer activity is a communication activity, as it involves the exchange of an offer between the parties. For this reason, both the function and communication perspectives (see Figure 5 and Figure 3) are used to identify the leaf activities of the *offer* abstract activity. In the function perspective, the offer abstract activity is decomposed to three concrete activities, i.e., *request offer, respond offer request* and *process offer response* activities. Additionally, the communication pattern in the communication perspective (see Figure 2) defines the possible combinations of the communication activities.

To use the two perspectives, first the consumer has to define its requirements. The offer activity will start with a request for offer from the consumer, in order to get the specifications of the provided translation service by the supplier. As each offer request is directed to one translating company, a single response by the supplier is required in return. When received, the response (i.e., the offer) has to be processed by the consumer. This shows that all three activities (*request offer*, *respond offer request*, and *process offer response)* are required for the consumer's process specification. The defined communication pattern states that a request can be associated with one or more respond activities and a respond with a response processing activity. Hence, the required three activities by the consumer are in conformance with the communication pattern. As a result, three activity leaves are identified (see Figure 10), i.e., *request offer, respond offer request* and *process offer response* activities. Figure 10 represents the entire subtree of the offer activity in Figure 5. However, in a different scenario the consumer might skip the *request offer* activity if it prefers to receive offers directly by suppliers.

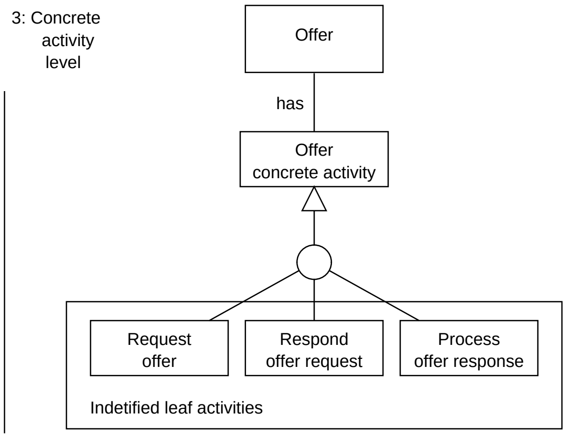


Figure 10 – Leaf activities of offer abstract activity

Leaf activities for the control concrete activity

After a successful negotiation and contract establishment, the consumer would like to monitor the execution of the translation service closely and to exercise control over the execution. For example, the consumer might be dissatisfied with the style of the translated document. In this case, the consumer will request correction from the service supplier to be performed.

Starting from the enactment phase, and following the monitoring and control specialization branch, the company identifies the control concrete activity. However for the specific case, the consumer would like to further decompose this activity, as it does not express the way the control will be imposed. In this example, the consumer will impose control through requests for correction to the service supplier and hence the control activity becomes a communication activity. Using the communication perspective the consumer can easily specialize it into leaf activities. The communication perspective allows request with multiple responses and single messages to be defined (Figure 2). The consumer uses the communication pattern as a guide for the identification of the communication activities. Examining its requirements for the control process, the consumer identifies the following steps.

- The first step is to send a *request* for correction from the consumer.
- The consumer expects at most one *response* from the service supplier. The service supplier provides a response to the request with the required changes fully or partially applied.
- The consumer *processes* the response.
- If the consumer is satisfied with the response, an *acceptance* of the provided correction will be sent. Otherwise a new *request* for correction will be sent.

In this way, four leaf activities for the control activity are identified (see Figure 11), i.e., *request correction*, *respond correction request*, *process correction response* and *accept correction*. The *request correction, respond correction request, process correction response,* and *accept correction* activities are specializations of the *single response, required response, response processing*, and the *accept* activities respectively from the communication perspective (see Section 5.5 for a discussion on new specialization of communication activity types). In a way analogous to the previous example, the consumer verifies that the four identified leaf activities are in conformance with the communication pattern.

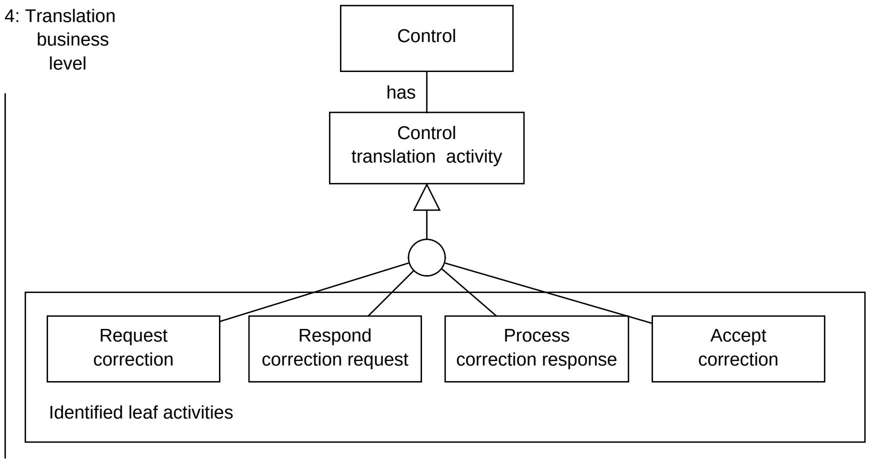


Figure 11 - Specialization of control activity

This example shows that the function and communication perspectives facilitate easy identification of the participating activities in the concrete e-contracting process. The communication perspective is used in two ways. In the first example, it helps to determine, which of already defined leaf activities of the offer abstract activity can be used. In the second example, it is used to facilitate the specializations of the control concrete activity.

8.3 Consistency rules

In the rest of the example, we consider that a company has already identified all leaf activities that constitute its e-contracting process (see Appendix B for the complete list). Next, consistency rules must be applied on the selected leaf activities.

For the standard leaf activities, the consumer can use predefined consistency rules (see Appendix A). This is the case for the *offer* activity. The consumer can retrieve:

IMPLY (Offer, Elaborate general provisions)
IMPLY (Customisation, Offer)
SEQUENCE (Elaborate general provisions, Offer, Customisation)
SEQUENCE (Request offer, Respond offer request)
IMPLY (Process offer response, Respond offer request)
SEQUENCE (Respond offer request, Process offer response)

As the offer activity is important for the consumer, he additionally defines REQUIRED (Offer), which applied to offer concrete activities leads to:

REQUIRED (Request offer)
REQUIRED (Respond offer request)
REQUIRED (Process offer response)

The consumer has to define consistency rules for the control activities by itself. Rule inheritance explained in Section 6.2 facilitates this process. As this is a communication activity, the following rules are predefined (see Appendix A on defining consistency rules over communication activities):

SEQUENCE (Request correction, Respond correction request, Process correction response)
IMPLY (Process correction response, Respond correction request)
IMPLY (Accept correction, Process correction response)
SEQUENCE (Process correction response, Accept correction)

Using the list of consistency rules, the consumer has the following rules:

DURING (Exchange value, Monitoring and control)
IMPLY (Control, Track)
SEQUENCE (Track, Control)
IMPLY (Control, Screen)
SEQUENCE (Screen, Control)

The DURING and SEQUENCE operators are directly inherited.
Additionally, the consumer has to define:

IMPLY (Request correction, Track)
IMPLY (Request correction, Screen)

Before specifying concrete processes, a start event that triggers the activity and if necessary an end event for it must be defined. In our example, an event internal to the company *offer needed* will trigger the *request offer* activity. This activity ends when a *response is received* or a *timeout occurs*. The event *offer received* at the consumer side will trigger the *process offer response* activity. Analogously, start and end events are identified for the *control* activities. Next, activity diagrams can be constructed.

8.4 Concrete process specification

In this paragraph, we show how the concrete process specifications for the offer and control activities are constructed. From the perspective of the service consumer, it is necessary to specify the *request offer* and the *process offer response* activities. These activities will be performed at the service consumer side and the contracting management system will require specification of the tasks to be performed. However, for the service consumer it is sufficient to identify the *respond offer request* activity without further specifying it. It has to be identified in order to define a correct e-contracting process specification. This activity will be executed at the service supplier side and has to be specified by the supplier.

First, we construct the *request offer* specification. We use the communication perspective (see Figure 3) to inherit tasks to the *request offer* activity from super-activities (see Section 4.3). From the root *communication activity*, the *request offer* activity inherits *elaborate offer request* and *send offer request* tasks. From the *request* activity (a specialization of the root activity) the *request offer* activity inherits the *wait* state. Finally, the *offer request* contains three basic activities, i.e., *elaborate offer request, send offer request*, and *wait* state (see Figure 12).

Analogously, using the communication perspective, the *process offer response* activity diagram is constructed. The response is received by the requesting party and is analysed. Based on the analyses, a statement on the received offer is elaborated. The *analyse response* and *elaborate statement* tasks are inherited from the *response processing* activity in the communication perspective.

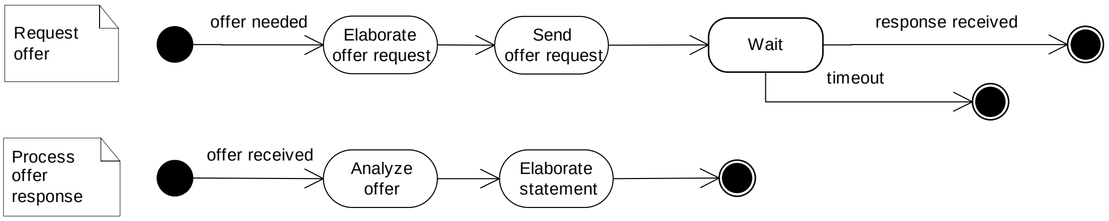


Figure 12 – Concrete process specification of the Offer activity

Figure 13 shows the process specification for the control activity. As the way of construction is analogous, we do not discuss it.

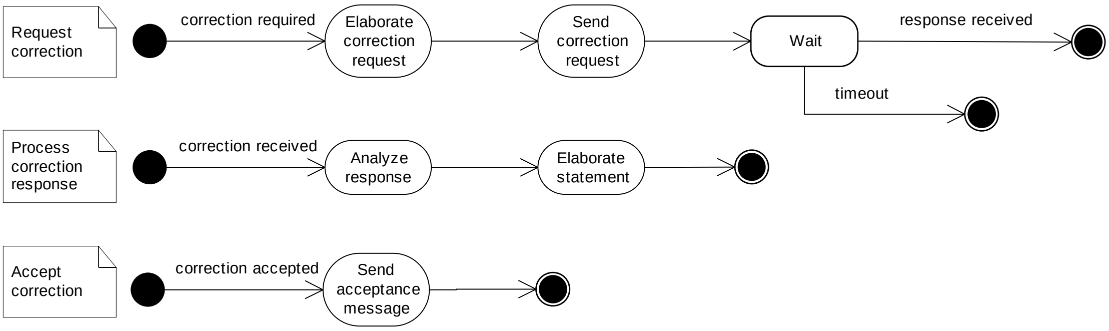


Figure 13 – Concrete process specification of the Control activity

The specification of the Offer and the Control activities is only a fragment of the complete concrete e-contracting process specification for the service consumer provided in Appendix B. The Offer specification is part of the pre-contracting phase specification and the Control specification is part of the enactment phase specification. This fragment illustrates the usage of both perspectives and the consistency rules in two different cases, i.e., for the specification of concrete level activities and activities from a lower level of abstraction not defined in the function perspective. The specification of the rest of the leaf activities identified by the service consumer is analogous and we do not further discuss it.

The described specification of concrete activities is based solely on the function and communication perspectives. The consistency rules that are defined over these concrete specifications set the basis for defining and executing correct e-contracting specifications. The example demonstrates that the two perspectives and the consistency rules are a powerful tool that helps for complete and coherent e-contracting process specifications.

9 Related work

To position our approach, we relate it to other developments in this field. First, we relate our approach to "Secure Electronic Contracts" research project that was carried out at the University of St. Gallen [SeC99]. In this project, fundamental requirements on secure electronic contracts were delivered. In this report, we examine the identified requirements on the e-contracting processes. Next, we describe the CrossFlow research project carried out by several partners among which the Twente University, IBM, and GMD. In CrossFlow, compared to the SeCo project, e-contracting processes are considered at a lower level of detail. Other research projects in which the e-contracting process is briefly discussed are [Goo00], [Gri97]. As they do not deliver any substantial results on the e-contracting process modelling, we do not discuss them. A number of efforts exist for the standardization of cross-organizational activities [Ang01a]. We relate our approach to RosettaNet and ebXML, which are established standardization efforts. Finally, we pay attention to a specific proprietary development. SAP AG is a leading company in this domain providing an e-business platform that supports the internal business processes and the collaboration between companies. For this reason, we describe the SAP business-modelling tool called Solution Composer.

9.1 Secure Electronic Contracts project

The project "Secure Electronic Contracts" (SeCo) at St. Gallen University researched the requirements for secure electronic contracting and its possible applications. In [Gis00], three views on the e-contracting process are depicted. The view on e-contracting activities is equivalent to our phase level of abstraction in the function view. However, the authors describe only briefly the phases and the processes that constitute them. In their paper, a document and a legal perspective are aligned to the activities view. The document view shows the documents delivered in the course of each phase. This is not enough to give a clear vision over the communication activities performed by the parties, through which these documents are delivered. In the legal perspective, legal issues are discussed that are not directly related to our approach. The three views are described at a high level of abstraction and only sketch the general characteristics of the e-contracting process. In our work, we describe in greater details the contracting process, reaching level of concrete contracting activities. As already explained, to achieve coherence of communication activities between parties, we distinguish in our model a separate communication view.

9.2 CrossFlow project

CrossFlow [Cro00] is an ESPRIT project looking into the support for cross-organizational workflow management in virtual enterprises. It was successfully completed in the end of 2000. In the CrossFlow project, workflow service outsourcing within a service consumer/supplier paradigm is considered. Contracting processes are not investigated in detail. Described is one pattern for service contract establishment [Cro99]. It consists of the following phases: making initial contact, exchanging information, negotiating, reaching an agreement, signing a contract, contract establishment. The fixed contracting process with this level of details suffices for the goals of the project. As we have shown, for flexible e-contracting between different parties, a more advanced approach is required.

In the CrossFlow project, the electronic contract contains a description of the enactment process to a level required for collaboration between the parties. Five parts in the contract content are distinguished [Koe00]. Two of these parts are of interest for us in this report. The process description part provides a workflow specification of the traded process. This can be regarded as a context dependent specialization of the deliver service concrete activity in our functional perspective. The enactment part specifies additional functionalities required for the collaboration between the parties. Such functionalities can be monitoring and control of the enactment process, adaptation of the supplier’s workflow by the consumer, etc. These activities are part of the monitoring and control abstract activity in the function perspective. The specification of part of the enactment phase in the contract content, illustrates that concrete activities of the enactment phase will be often additionally specialized to serve for best collaboration between parties in particular business situations.

9.3 RosettaNet

RosettaNet is a standardization effort aiming at a description of the cross-organizational business processes. In RosettaNet, a three level hierarchy of the activities of the parties is used. This hierarchy is built by decomposing the domain of e-business supply chain activities. The e-business supply chain domain is divided into *clusters*. Each cluster is divided into *segments*. In each segment exist one or more Partner Interface Processes (PIPs). PIPs are the leaves of the decomposition tree. A PIP specifies the business roles for a given business process, the business activities between these roles, and the type, content and sequence of the business documents exchanged by the partners. Some of the PIPs are still to be defined. These three levels of decomposition of the e-business supply chain activities have a role, similar to the three levels of abstraction in our function perspective, i.e., achieving completeness by gradually reducing abstraction. PIPs are the third level of decomposition and the activities that they define are specified through activity diagrams. The level of abstraction of PIPs is comparable to the level of concrete activities in our function perspective. In RosettaNet, two communication patterns exist, i.e., request-response pairs of activities and notification activities. A request-response pair is positioned in one PIP. As PIPs are already concrete specifications, this gives only one communication possibility for parties. In our communication pattern, we apply only constraint on the activities and give maximum flexibility for communication to the contracting parties. Also, in contrast to RosettaNet, we allow a request followed by several responses to be defined (e.g. in the negotiation activity).

Additionally, in PIPs, collaboration and sequence diagrams are used to specify the business roles identified for this activity and the business documents exchanged by the partners. As already mentioned, the exchanged documents between parties in an e-contracting process are not of importance for our approach and we do not discuss them.

In RosettaNet, in contrast to our approach, no attention is paid to the internal business processes and their relation to the cross-organizational processes. As a result, only the e-contracting communication activities can be extracted from this standard. In this report, we look towards description of the complete e-contracting process.

RosettaNet provides a highly restrictive model. RosettaNet requires parties to comply with the defined specifications in order to guarantee coherence of the executed process on both sides. There are no formal rules that constrain the order of the execution of the PIPs and the relations between them. In some of the PIP specifications, relations to other PIPs are briefly mentioned in a free text format.

9.4 ebXML

ebXML is a standardization effort that provides a framework for doing business through electronic means [ebX01a]. In ebXML, the Business Process Specification Schema (BPSS) provides a framework for the specification of business processes. For the modelling of business processes the UN/CEFACT Modelling Methodology (UMM) is used. UMM supports modelling of business processes in general and does not provide a specific support for modelling of concrete e-contracting specifications. In ebXML, no specific attention is paid to activities that are internal for companies.

In ebXML, a Collaboration Protocol Profile (CPP) defines the capabilities and requirements of a party in a possible e-commerce process. The CPP embeds or references the Business Process Specification of the party. When two parties want to establish business relations, they compare their CPPs and the intersection of the CPPs that is mutually agreed on defines the Collaboration Protocol Agreement (CPA) [ebX01a]. However, it is not clear how this intersection is performed. In BPSS, the Business Transaction Choreography defines the order in which the transactions are executed. The rules used to define the order of activity execution are based on the activity diagrams control flow concepts, i.e. transition guards, start state, completion state, etc.

In BPSS, business collaborations may contain a set of business activities. A business activity can be a business transaction or business collaboration. Transactions are the atomic units that cannot be further decomposed. This decomposition of the business collaborations, similar to our function perspective, aims at providing a way for reducing the complexity of process specifications. However, the use of activity diagram control flow concepts implies strictly defined business collaborations. This significantly increases the complexity for e-contracting processes specifications as usually the number of possible combinations between e-contracting activities is very high (see discussion in Section 6.1). In our approach, we use consistency rules to provide flexible e-contracting processes specifications. The communication pattern that has to guarantee coherent communication between parties in the BPSS is simpler than in our approach. The only possible communication in a business transaction is through request-response pairs of activities. Responses in these pairs are optional. Thus, a request without a response in BPSS is equivalent to an inform activity in our communication pattern. A series of request-response pairs that simulate a two-way conversation between parties is equivalent to a request followed by arbitrary number of responses respectively. Though the communication pattern in BPSS is simpler in its definition, the requirement each response activity to be preceded by a request activity, can complicate conversations between parties unnecessary (e.g. during negotiations between parties).

In our work, we concentrate solely on the e-contracting processes. This allows us to define a more clear approach for construction and use of e-contracting process specifications. With our method coupling between the internal and external business processes can be achieved. Consistency rules allow flexible and consistent e-contracting process specification to be defined.

9.5 SAP Solution Composer

An example of a proprietary solution for business modelling is a composer tool provided by SAP [SAP01]. This tool is intended for construction of process specifications. It does not prescribe a modelling method. In addition to the modelling tool, SAP provides predefined industry-specific and cross-industry business process models. These models can be used as templates for the creation of concrete business processes. However, they are not flexible and imply specific choices. For example, in the Request for Quotation (RfQ) model, the consumer publishes a RfQ, evaluates the received responses and sends notification to the supplier with the best offer. This predefined model does not allow a request for customisation or for more information to be sent to one or more of the suppliers in order to obtain a personalized quotation or information that is not given in the offer.

Contract negotiation and establishment is not part of the processes described in the models, which makes the set of predefined models incomplete. A taxonomy is used to structure the list of predefined business models. The criteria for the classification of the processes in this taxonomy are ambiguous and do not allow easy identification of the models. Relations between different process specifications are not elaborated.

10 Conclusions and future work

In this report, we have described an e-contracting process model. The model is based on two perspectives, i.e., function and communication. These perspectives, in combination with a set of consistency rules provide for tools to guard completeness and consistency of e-contracting processes. Based on the two perspectives and the set of consistency rules, we define an approach for specification of concrete e-contracting process. Our model can be used for the analysis of existing e-contracting process specifications as well.

To best illustrate the proposed approach, we have applied it to an imaginary business scenario. In the context of this example, we have shown how the model is used in building concrete e-contracting process specification. For reasons of brevity, we have concentrated on several sample activities.

This work is a foundation for further research in the area of the e-contracting processes. Additional work has to be done in the directions of data aspects in e-contracting processes, extensions of the consistency rules, and alignment of the process specifications of contracting parties. Also, a tool for support of the design of e-contracting process specifications has to be elaborated.

An information system is required for the support of e-contracting processes. However, a detailed reference architecture of an information system that will serve this goal does not yet exist. This research work is part of the e-contracting system environment analysis. A data model is required in addition to the process model. More specifically, the e-contract content and its representation are to be researched. The process and data models together allow collecting requirements for an information system that supports an e-contracting process in its four phases. This preliminary work is a step towards the construction of detailed e-contracting reference architecture, which is our research goal.

Acknowledgements

We thank Roel Wieringa for his remarks and recommendations during the discussions on the work presented in this report.

References

[Alo99] Alonso G, Fiedler U, Hagen C, Lazcano A, Schuldt H, Weiler N. WISE: business to business e-commerce. Procs. 9th International Workshop on Research Issues on Data Engineering: Information Technology for Virtual Enterprises. Australia, 1999.

[Ang01a] Angelov S, Grefen P. B2B eContract handling - a survey of projects, papers and standards. Twente University: CTIT Technical Report 01-21. 2001. http://www.ctit.utwente.nl/publications/Tr01/.

[Ang01b] Angelov S, Grefen P. A framework for the analysis of B2B electronic contracting support. Procs. 4th Edispuut Conference - Multidisciplinary Perspectives on Electronic Commerce. The Netherlands, 2001.

[Ang02] Angelov S, Grefen P. A conceptual framework for B2B electronic contracting. Procs. 3rd IFIP Working Conference on Infrastructures for Virtual Enterprises. Portugal, 2002.

[Bus99] Bussler C. Workflow class inheritance and dynamic workflow class binding. Procs. of the Workshop Software Architectures for Business Process Management at the 11th Conference on Advanced Information Systems Engineering CAISE’99. Germany, 1999.

[Cas00] Casati F, Castano S, Fugini M. G, Mirbel I, Pernici B. Using patterns to design rules in workflows. Transaction on Software Engineering Journal. Vol. 26, No 5. 2000.

[Chi02] Chiu D.K.W, Cheung S.C, Till S. A three layer architecture for e-contract enforcement in an e-service environment. Technical Report HKUST-CS 19-02. 2002.

[Coo00] Coombs A. Business relationship vs. legal contract? White paper. Echopass Corporation. 2000.

[Cro99] CrossFlow Consortium. Architecture Description. CrossFlow deliverable: D3.a, La Gaude, 1999.

[Cro00] CrossFlow project. 2000. http://www.crossflow.org/.

[Dan98] Dan A, Dias D, Nguyen T, Sachs M, Shaikh H, King R, Duri S. The Coyote project: framework for multi-party e-commerce. Procs. of the 7th Delos Workshop on Electronic Commerce. Greece, 1998.

[Das97] Daskalopulu A, Sergot M. J. The representation of legal contracts. AI and Society, 11 (1 & 2). 1997.

[diC02] diCarta. Enterprise contract management. White paper. diCarta Inc. 2002.

[ebX01a] ebXML. Technical architecture specification v1.0.4. ebXML. 2001. http://www.ebxml.org/specdrafts/ebXML_TA_v1.0.4.pdf.

[ebX01b] ebXML. ebXML Business process specification schema, v1.01 6. ebXML. 2001. http://www.ebxml.org/specs/ebBPSS.pdf.

[Eri00] Eriksson H.-E, Penker M. Business modeling with UML: business patterns at work. John Wiley & Sons. 2000.

[Gis00] Gisler M, Stanoevska-Slabeva K, Greunz M. Legal aspects of electronic contracts. Procs. of the CAISE’00 Workshop on Infrastructure for Dynamic Business-to-Business Service Outsourcing (IDSO'00). Sweden, 2000.

[Goo00] Goodchild A, Herring C, Milosevic Z. Business contracts for B2B. Procs. of the CAISE00 Workshop on Infrastructure for Dynamic Business-to-Business Service Outsourcing (IDSO'00). Sweden, 2000.

[Gre00] Grefen P, Aberer K, Hoffner Y, Ludwig H. CrossFlow: Cross-organizational workflow management in dynamic virtual enterprises. International Journal of Computer Systems Science & Engineering, Vol. 15, No. 5, pp. 277-290. 2000.

[Gre01] Grefen P, Angelov S. A three-level process framework for contract-based dynamic service outsourcing. Procs. 2nd International Colloquium on Petri Net Technologies for Modelling Communication Based Systems. Germany, 2001.

[Gre02a] Grefen P, Ludwig H, Angelov S. A framework for e-services: a three-level approach towards process and data management. Twente University: CTIT Technical Report 02-07, 2002. http://www.ctit.utwente.nl/publications/Tr02/.

[Gre02b] Grefen P, Angelov S. On τ-, μ-, π-, and ε-Contracting. Procs. Web Services, e-Business, and the Semantic Web. Canada, 2002.

[Gri97] Griffel F, Tu T, Münke M, Merz M, Lamersdorf W, Mira da Silva M. Electronic contract negotiation as an application niche for mobile agents. Procs. 1st International Workshop on Enterprise Distributed Object Computing. USA, 1997.

[Gri98] Griffel F, Boger M, Weinreich H, Lamersdorf W, Merz M. Electronic contracting with COSMOS - how to establish, negotiate and execute electronic contracts on the internet, Procs. from 2nd Int. Enterprise Distributed Object Computing Workshop (EDOC '98). USA, 1998.

[Jac97] Jackson M, Twaddle G. Business process implementation building workflow systems. Addison-Wesley. 1997.

[Kee00] Keen P, Balance C, Chan S, Schrump S. Electronic commerce relationships: trust by design. Prentice Hall PTR. 2000.

[Koe00] Koetsier M, Grefen P, Vonk J. Contracts for cross-organizational workflow management. Procs. 1st International Conference on Electronic Commerce and Web Technologies. UK, 2000.

[Oll88] Olle T, Hagelstein J, Macdonald I, Rolland C, Sol H, van Assche F, Verrijn-Stuart A. Information systems methodologies: a framework for understanding. Addison-Wesley. 1988.

[Ora01] Oracle. Enterprise Contract Management: The Roadmap to Success. An Oracle white paper. 2001

[OMG01] OMG. Unified modeling language specification. v1.4. OMG. 2001.

[Pon02] Ponton Consulting. OCS Manual, vers. 0.6. 2002.
http://www.ponton-consulting.de/ocs/

[Put01] Putman J. Architecting with RM-ODP. Prentice Hall. 2001

[Roe96] Wieringa R. Requirements engineering: frameworks for understanding. John Wiley. 1996.

[Ros01] RosettaNet. RosettaNet implementation framework: core specification (RNIF 02). RosettaNet. 2001. http://www.rosettanet.org.

[SAP01] SAP. SAP solution composer, v1.5.2. SAP AG. 2001.
http://www.sap.com/solutionmaps/.

[Sch00] Schoop M, Quix C. Towards effective negotiation support in electronic marketplaces. Procs. of 10th Annual Workshop on Information Technologies & Systems, WITS 2000. Australia, 2000.

[SeC99] Secure Electronic Contracts (SeCo). SPP ICS Project N° 5003-54544. 1999.

[Tur00] Turban E, Lee J, King D, Chung H. Electronic commerce - a managerial perspective. Prentice-Hall. 2000.

[Tut02] Tut M T, Edmond D. The use of patterns in service composition. Procs. Web Services, e-Business, and the Semantic Web. Canada, 2002.

[WSD01] Web Services Description Language (WSDL) 1.1. 2001.
http://www.w3.org/TR/wsdl

Appendix A

List of consistency rules for e-contracting activities

1. General constraints

SEQUENCE (Information phase, Pre-contracting phase, Contracting phase, Enactment phase)

Communication pattern constraints

Let A_0, B_0, C_0 are communication activities and A is a request, B is a respond, and C is a processing of the response activity. Then the following constraints are applied:

SEQUENCE (A_0, B_0, C_0)
IMPLY (C_0, B_0)

When communication continues after a response processing activity (with a new response or inform activity) the following constraints are additionally applied[2]:

EXCLUDE (B_i, D_i)
IMPLY (B_i, C_{i-1}) OR IMLPY (D_i, C_{i-1})
SEQUENCE (C_{i-1}, B_i)
SEQUENCE (C_{i-1}, D_i)

2. Information phase

SEQUENCE (General preparations, Communicate information)

a. General preparations

REQUIRED (Elaborate general provisions)

b. Communicate information

IMPLY (Respond information request, Multicast information) OR IMPLY (Respond information request, Search information)
SEQUENCE (Multicast information, Respond information request)
SEQUENCE (Search information, Respond information request)
IMPLY (Process information response, Respond information request)
SEQUENCE (Respond information request, Process information response)

3. Pre-contracting phase

IMPLY (Offer, Elaborate general provisions)
IMPLY (Customisation, Offer)
SEQUENCE (Elaborate general provisions, Offer, Customisation)

a. Offer

SEQUENCE (Request offer, Respond offer request, Process offer response)
IMPLY (Process offer response, Respond offer request)

b. Party information

SEQUENCE (Request party information, Respond party information request, Process party information response)
IMPLY (Process party information response, Respond party information request)

c. Customisation

SEQUENCE (Request customisation, Respond customisation request, Process customisation response)
IMPLY (Process customisation response, Respond customisation request)

4. Contracting phase

[2] The index indicates that the activities belong to the ith continuation ($i \geq 1$).

IMPLY (Validation and signing, Negotiation)
IMPLY (Storing, Validation and signing)
SEQUENCE (Negotiation, Validation and signing, Storing)
REQUIRED (Negotiation)

a. Negotiation
SEQUENCE (Request contract offer, Respond contract offer request, Process contract offer response)
IMPLY (Process contract offer response, Respond contract offer request)
EXCLUDE (Accept contract offer$_i$, Reject contract offer$_i$)
EXCLUDE (Respond contract offer request$_i$, Accept contract offer$_i$)
EXCLUDE (Respond contract offer request$_i$, Reject contract offer$_i$)
IMPLY (Accept contract offer$_i$, Process contract offer response$_i$) OR IMPLY (Reject contract offer$_i$, Process contract offer response$_i$) OR IMPLY (Respond contract offer request$_i$, Process contract offer response$_i$)
SEQUENCE (Process contract offer response, Accept contract offer)
SEQUENCE (Process contract offer response, Reject contract offer)
SEQUENCE (Process contract offer response, Respond contract offer request)
IMPLY (Respond contract offer request, Sign)
SEQUENCE (Sign, Respond contract offer request)
REQUIRED (Accept contract offer)

b. Validation and signing
SEQUENCE (Validate, Sign)
REQUIRED (Sign)

c. Storing
REQUIRED (Store internally)

5. Enactment phase
IMPLY (Enactment phase, Sign)
SEQUENCE (Sign, Enactment phase)
DURING (Exchange value, Monitoring and control)
SEQUENCE (Monitoring and control, Evaluation)
IMPLY (Evaluation, Exchange value)
SEQUENCE (Exchange value, Evaluation)

a. Exchange value
SEQUENCE (Request enactment data, Respond enactment data request, Process enactment data response)
IMPLY (Process enactment data response, Respond enactment data request)
REQUIRED (Deliver service)
REQUIRED (Deliver reward)

b. Monitoring and control
IMPLY (Track, Control)
SEQUENCE (Control, Track)
IMPLY (Track, Notify)
SEQUENCE (Notify, Track)
IMPLY (Screen, Control)
SEQUENCE (Control, Screen)
IMPLY (Screen, Notify)
SEQUENCE (Notify, Screen)

c. Evaluation
IMPLY (Store evaluation internally, Evaluate)
SEQUENCE (Evaluate, Store evaluation internally)
IMPLY (Store evaluation externally, Evaluate)
SEQUENCE (Evaluate, Store evaluation externally)

Appendix B

Complete process specification of the language translation business scenario

In Table 1, the identified concrete activities by the service consumer and the service supplier are given in the first and second column respectively (see Section 8.2 for discussion on identification of leaf activities). In the third column, the alignment of the activities identified by the service consumer and the service supplier is presented (see Section 7.2 for discussion on alignment of process specifications).

Analogously, in Table 2, the applied consistency rules by the service consumer and service supplier are listed in the first and the second column respectively. The third column shows the consistency rules that are applied over the common e-contracting specification.

Finally, concrete specifications for the service consumer and service supplier are presented. Each activity is triggered by an event. Events are produced by other e-contracting activities or by other business processes (see Section 7.1). A concrete specification of an activity is executed when a triggering event for this activity occurs and the consistency rules applied over this activity are not violated. In order not to overload the activity diagrams, the triggering events for each activity are not shown in the activity specifications.

1. Identified concrete activities

Table 1 – Identified concrete activities

Concrete activities identified by the service consumer	**Concrete activities identified by the service supplier**	**Concrete activities agreed by both parties to be included in the common e-contracting process specification**
Information phase		
General preparations concrete activities Elaborate general provisions *Information concrete activities* Search information Respond information request	*General preparations concrete activities* Elaborate general provisions Prepare contract templates *Information concrete activities* Publish information (c) Respond information request Process information response Notify on availability	*(This phase does not have to be agreed on, see Section 6.1 for explanations)*

Pre-contracting phase		
Offer concrete activities Request offer Respond offer request (s) Process offer response *Party information concrete activities* Request party information Respond party information request (c) (s) Process party information response	*Offer concrete activities* Request offer (c) Respond offer request *Party information concrete activities* Request party information Respond party information request (c) (s) Process party information response *Customisation concrete activities* Request customisation (c) Respond customisation request	*Offer concrete activities* Request offer (c) Respond offer request (s) *Party information concrete activities* Request party information(c)(s) Respond party information request (c) (s)
Contracting phase		
Negotiation concrete activities Request contract offer Respond contract offer request (s) Process contract offer Accept contract offer Reject contract offer *Validation and signing concrete activities* Validate Sign *Storing concrete activities* Store internally	*Negotiation concrete activities* Request contract offer (c) (s) Respond contract offer request (c) (s) Process contract offer (c) (s) Accept contract offer (c) (s) Reject contract offer (c) (s) *Validation and signing concrete activities* Validate Sign *Storing concrete activities* Store internally	*Negotiation concrete activities* Request contract offer (c) Respond contract offer request (s) Accept contract offer (c) Reject contract offer (c) *Validation and signing concrete activities* Sign (c) (s)

Enactment phase		
Exchange value concrete activities	*Exchange value concrete activities*	*Exchange value concrete activities*
Request enactment data (s)	Request enactment data	Request enactment data (s)
Respond enactment data request	Respond enactment data request (c)	Respond enactment data request (c)
Deliver service (s)	Deliver service	Deliver service (s)
Deliver reward	Deliver reward (c)	Deliver reward (c)
Monitoring and control concrete activities	*Monitoring and control concrete activities*	*Monitoring and control concrete activities*
Track	Track	
Screen	Screen	
(control specialization)	Control	
Request correction		Request correction (c)
Respond correction request (s)		Respond correction request (s)
Process correction response		
Accept correction		Accept correction (c)
Notify	Notify	Notify (c) (s)
Evaluation concrete activities	*Evaluation concrete activities*	
Evaluate	Evaluate	
Store evaluation internally	Store evaluation internally	
Store evaluation externally	Store evaluation externally	

Legend:

(c) or (s) An identified activity that is expected to be supported by the other party (consumer or supplier).

(c) (s) An identified activity that is supported by the party and is expected to be supported by the other party as well.

Note:

The consumer has specialized the control activity according to his preferences. In order to achieve an agreement on one common e-contracting specification, the supplier has to accept this specialization and add the *respond correction request* activity. The rest of the identified activities by the parties are compliant. This allows a successful e-contracting process to be executed.

2. Consistency rules

Table 2 - Consistency rules

Service consumer consistency rules	**Service supplier consistency rules**	**Aligned consistency rules**
General rules SEQUENCE (Information phase, Pre-contracting phase, Contracting phase, Enactment phase)		
Information phase		
SEQUENCE (General preparations, Communicate information) REQUIRED (Elaborate general provisions) $_{C}$ REQUIRED (Search information) $_{C}$ IMPLY (Respond information request, Search information) SEQUENCE (Search information, Respond information request)	SEQUENCE (General preparations, Communicate information) REQUIRED (Elaborate general provisions) $_{S}$ REQUIRED (Publish information) $_{S}$ IMPLY (Process information response, Respond information request) SEQUENCE (Respond information request, Process information response)	*(This phase does not have to be agreed on, see Section 6.1 for explanations)*
Pre-contracting phase		
IMPLY (Offer, Elaborate general provisions) SEQUENCE (Elaborate general provisions, Offer) SEQUENCE (Request offer, Respond offer request, Process offer response) IMPLY (Process offer response, Respond offer request) REQUIRED (Request offer) $_{C}$ REQUIRED (Respond offer request) $_{S}$, REQUIRED (Process offer response) $_{C}$ SEQUENCE (Request party information, Respond party information request, Process party information response) IMPLY (Process party information response, Respond party information request)	IMPLY (Offer, Elaborate general provisions) IMPLY (Customisation, Offer) SEQUENCE (Elaborate general provisions, Offer, Customisation) SEQUENCE (Request offer, Respond offer request, Process offer response) IMPLY (Process offer response, Respond offer request) REQUIRED (Respond offer request) $_{S}$, REQUIRED (Process offer response) $_{C}$ SEQUENCE (Request party information, Respond party information request, Process party information response) IMPLY (Process party information response, Respond party information request) SEQUENCE (Request customisation, Respond customisation request, Process	SEQUENCE (Request offer, Respond offer request, Process offer response) IMPLY (Process offer response, Respond offer request) REQUIRED (Request offer) $_{C}$ REQUIRED (Respond offer request) $_{S}$, REQUIRED (Process offer response) $_{C}$ SEQUENCE (Request party information, Respond party information request, Process party information response) IMPLY (Process party information response, Respond party information request)

	customisation response) IMPLY (Process customisation response, Respond customisation request)	
	Contracting phase	
IMPLY (Validation and signing, Negotiation) IMPLY (Storing, Validation and signing) SEQUENCE (Negotiation, Validation and signing, Storing) REQUIRED (Negotiation) SEQUENCE (Request contract offer, Respond contract offer request, Process contract offer response) EXCLUDE (Accept contract offer, Reject contract offer) EXCLUDE (Respond contract offer request, Accept contract offer) EXCLUDE (Respond contract offer request, Reject contract offer) IMPLY (Accept contract offer, Process contract offer response) OR IMPLY (Reject contract offer, Process contract offer response) OR IMPLY (Respond contract offer request, Process contract offer response) REQUIRED (Respond contract offer request)$_{S}$ REQUIRED (Accept contract offer) IMPLY (Respond contract offer request, Sign) SEQUENCE (Sign, Respond contract offer request) SEQUENCE (Validate, Sign) REQUIRED (Sign) REQUIRED (Validate)$_{C}$ REQUIRED (Store internally)$_{C}$	IMPLY (Validation and signing, Negotiation) IMPLY (Storing, Validation and signing) SEQUENCE (Negotiation, Validation and signing, Storing) REQUIRED (Negotiation) SEQUENCE (Request contract offer, Respond contract offer request, Process contract offer response) EXCLUDE (Accept contract offer, Reject contract offer) EXCLUDE (Respond contract offer request, Accept contract offer) EXCLUDE (Respond contract offer request, Reject contract offer) IMPLY (Accept contract offer, Process contract offer response) OR IMPLY (Reject contract offer, Process contract offer response) OR IMPLY (Respond contract offer request, Process contract offer response) REQUIRED (Accept contract offer) IMPLY (Respond contract offer request, Sign) SEQUENCE (Sign, Respond contract offer request) SEQUENCE (Validate, Sign) REQUIRED (Sign) REQUIRED (Validate)$_{S}$ REQUIRED (Store internally)$_{S}$	IMPLY (Validation and signing, Negotiation) IMPLY (Storing, Validation and signing) SEQUENCE (Negotiation, Validation and signing, Storing) REQUIRED (Negotiation) SEQUENCE (Request contract offer, Respond contract offer request, Process contract offer response) EXCLUDE (Accept contract offer, Reject contract offer) EXCLUDE (Respond contract offer request, Accept contract offer) EXCLUDE (Respond contract offer request, Reject contract offer) IMPLY (Accept contract offer, Process contract offer response) OR IMPLY (Reject contract offer, Process contract offer response) OR IMPLY (Respond contract offer request, Process contract offer response) REQUIRED (Respond contract offer request)$_{S}$ REQUIRED (Accept contract offer)
	Enactment phase	
IMPLY (Enactment phase, Sign) SEQUENCE (Sign, Enactment phase) DURING (Exchange value,	IMPLY (Enactment phase, Sign) SEQUENCE (Sign, Enactment phase) DURING (Exchange value, Monitoring and control)	IMPLY (Enactment phase, Sign) SEQUENCE (Sign, Enactment phase) DURING (Exchange value, Monitoring and control)

Monitoring and control) SEQUENCE (Monitoring and control, Evaluation) IMPLY (Evaluation, Exchange value) SEQUENCE (Exchange value, Evaluation)	SEQUENCE (Monitoring and control, Evaluation) IMPLY (Evaluation, Exchange value) SEQUENCE (Exchange value, Evaluation)	SEQUENCE (Monitoring and control, Evaluation) IMPLY (Evaluation, Exchange value) SEQUENCE (Exchange value, Evaluation)
SEQUENCE (Request enactment data, Respond enactment data request, Process enactment data response) IMPLY (Process enactment data response, Respond enactment data request) REQUIRED (Respond enactment data request)$_C$ REQUIRED (Process enactment data response)$_S$ REQUIRED (Deliver service) $_S$ REQUIRED (Deliver reward) $_C$	SEQUENCE (Request enactment data, Respond enactment data request, Process enactment data response) IMPLY (Process enactment data response, Respond enactment data request) REQUIRED (Respond enactment data request)$_C$ REQUIRED (Process enactment data response)$_S$ REQUIRED (Deliver service) $_S$ REQUIRED (Deliver reward) $_C$	SEQUENCE (Request enactment data, Respond enactment data request, Process enactment data response) IMPLY (Process enactment data response, Respond enactment data request) REQUIRED (Respond enactment data request)$_C$ REQUIRED (Process enactment data response)$_S$ REQUIRED (Deliver service) $_S$ REQUIRED (Deliver reward) $_C$
IMPLY (Track, Notify) SEQUENCE (Notify, Track) IMPLY (Screen, Control) SEQUENCE (Control, Screen) IMPLY (Request correction, Track) SEQUENCE (Track, Request correction) IMPLY (Request correction, Screen) SEQUENCE (Screen, Request correction) SEQUENCE (Request correction, Respond correction request, Process correction response) IMPLY (Process correction response, Respond correction request) IMPLY (Accept correction, Process correction response) SEQUENCE (Process correction response, Accept correction)	IMPLY (Track, Control) SEQUENCE (Control, Track) IMPLY (Track, Notify) SEQUENCE (Notify, Track) IMPLY (Screen, Control) SEQUENCE (Control, Screen) IMPLY (Screen, Notify) SEQUENCE (Notify, Screen)	SEQUENCE (Request correction, Respond correction request, Process correction response) IMPLY (Process correction response, Respond correction request)
IMPLY (Store evaluation internally, Evaluate) SEQUENCE (Evaluate, Store evaluation internally) REQUIRED (Store evaluation internally) $_C$		

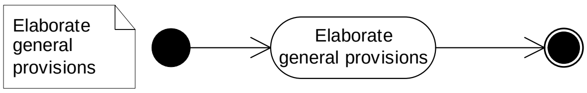
Legend:

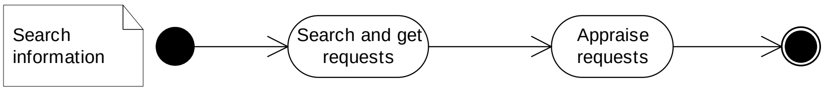
REQUIRED (A)$_C$	The index $_C$ indicates that activity A is required for the consumer.
REQUIRED (A)$_S$	The index $_S$ indicates that activity A is required for the supplier.

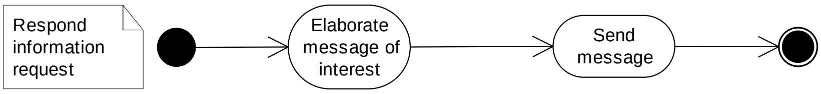
3. Concrete activities specifications

3.1 Concrete specifications for the service consumer

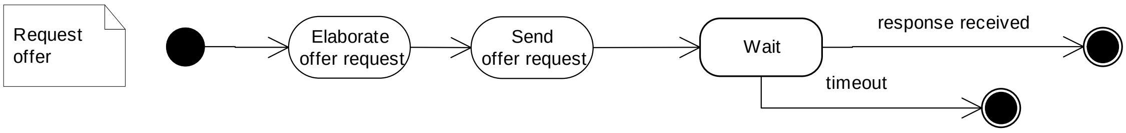
Information phase

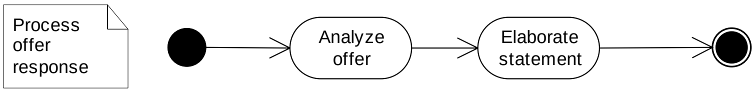


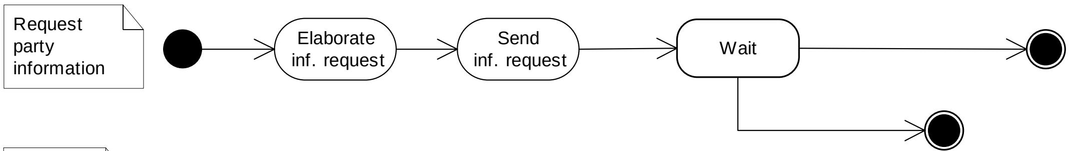


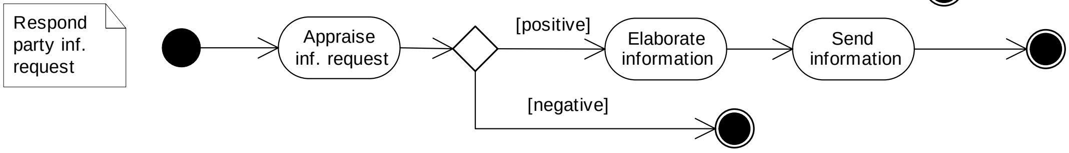


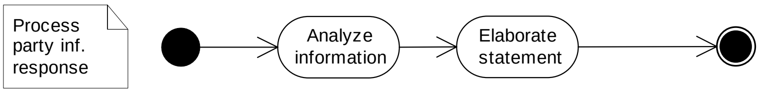
Pre-contracting phase



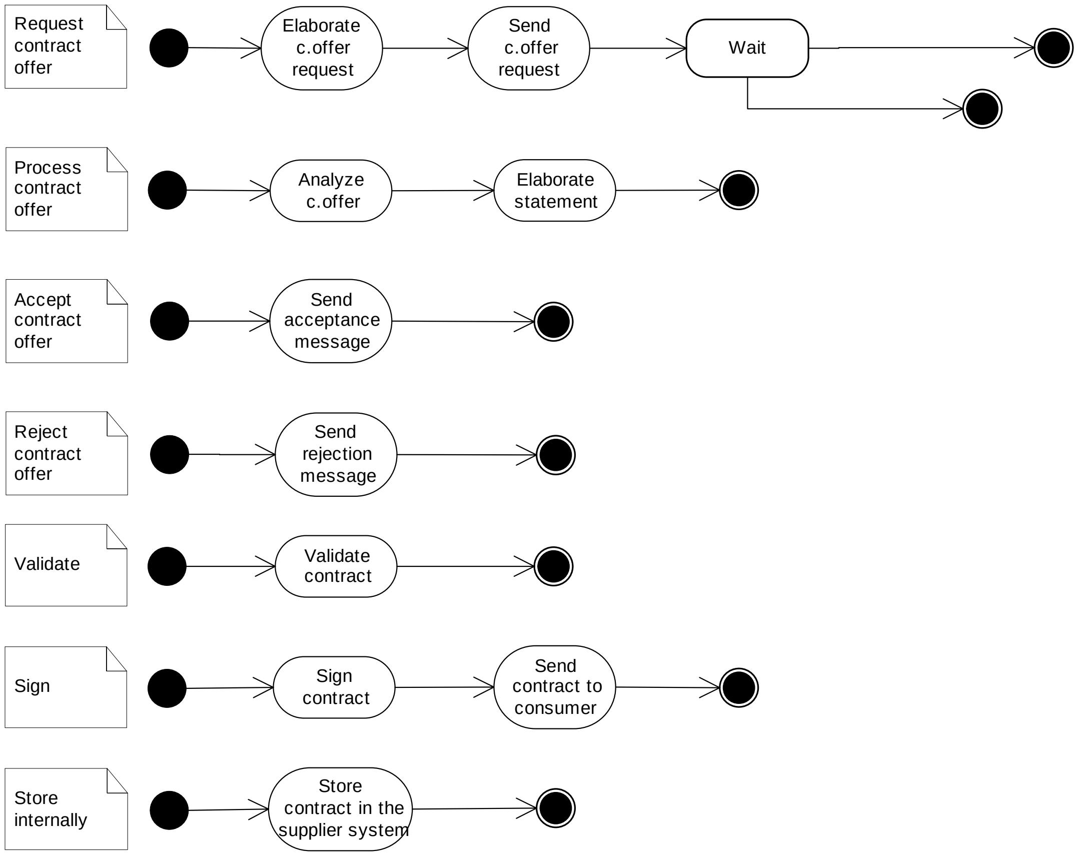




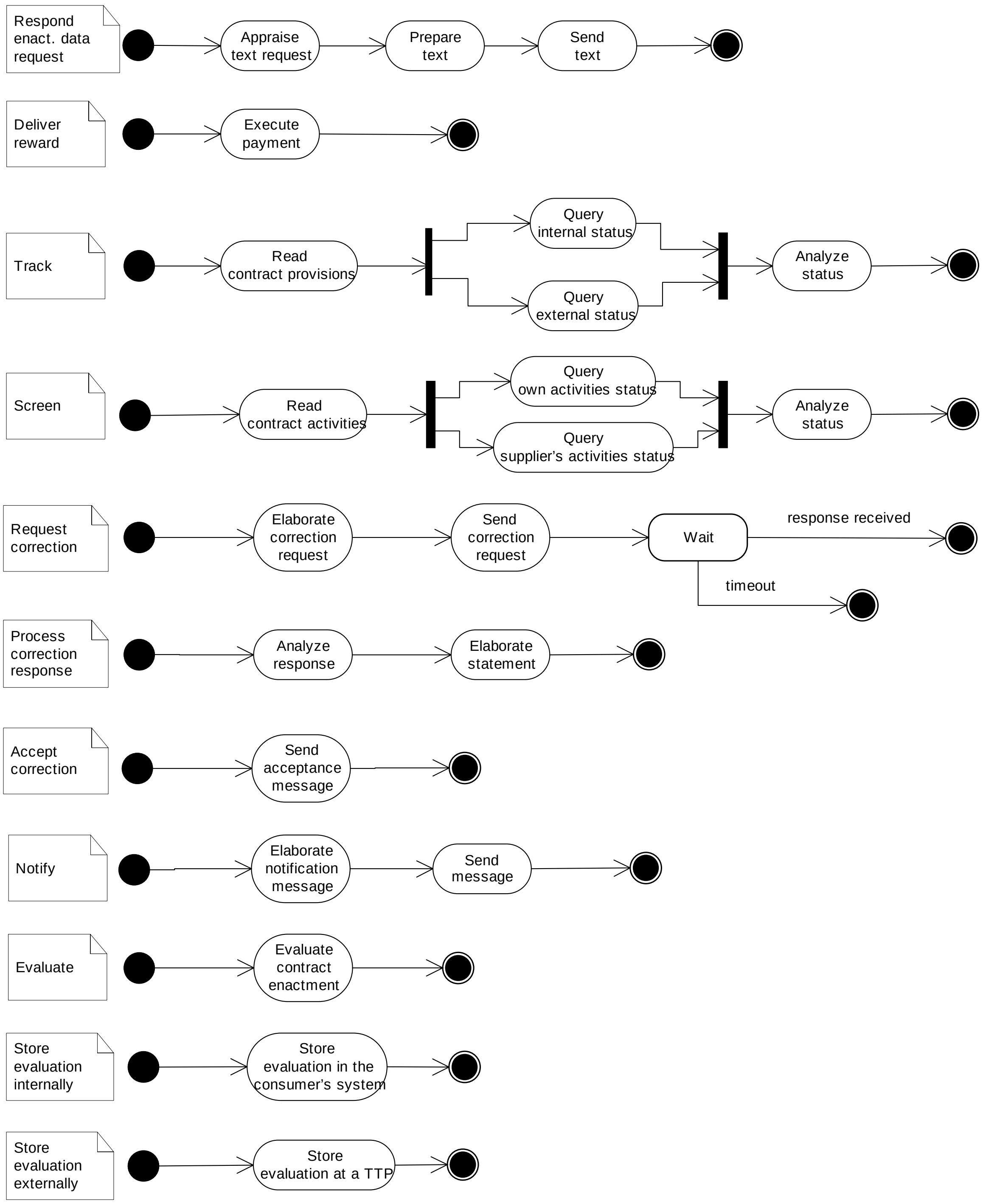




Contracting phase

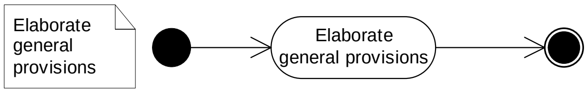


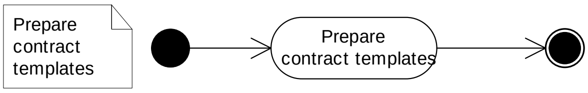
Enactment phase

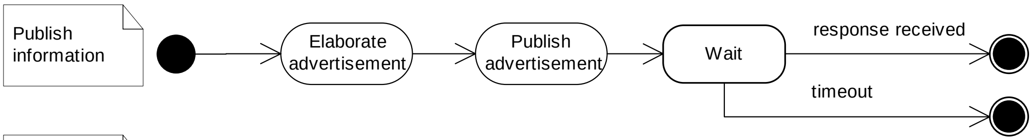


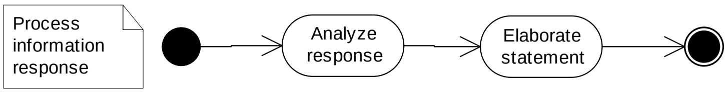
3.2 Concrete specifications for the service supplier

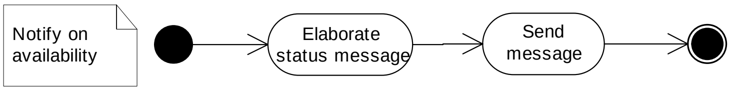
Information phase



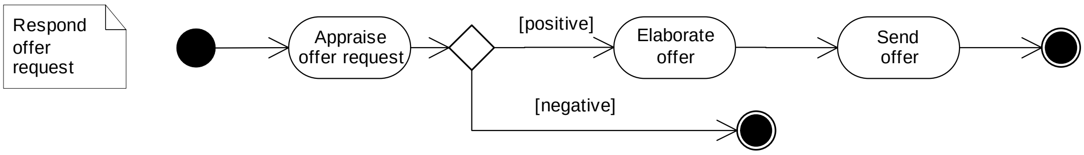


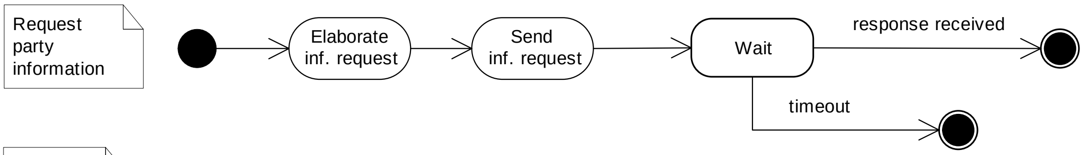


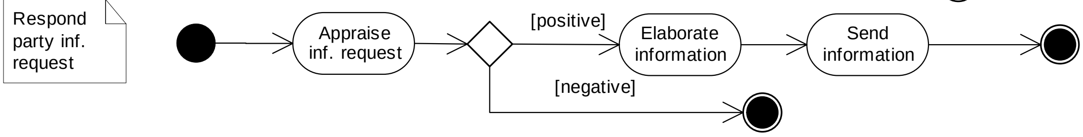


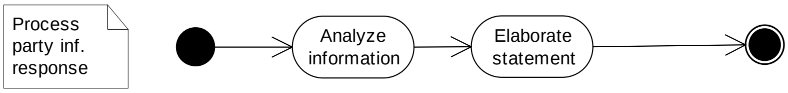


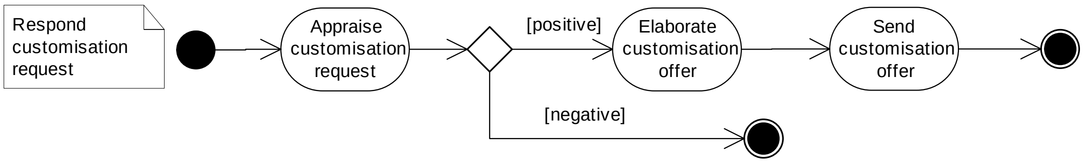
Pre-contracting phase



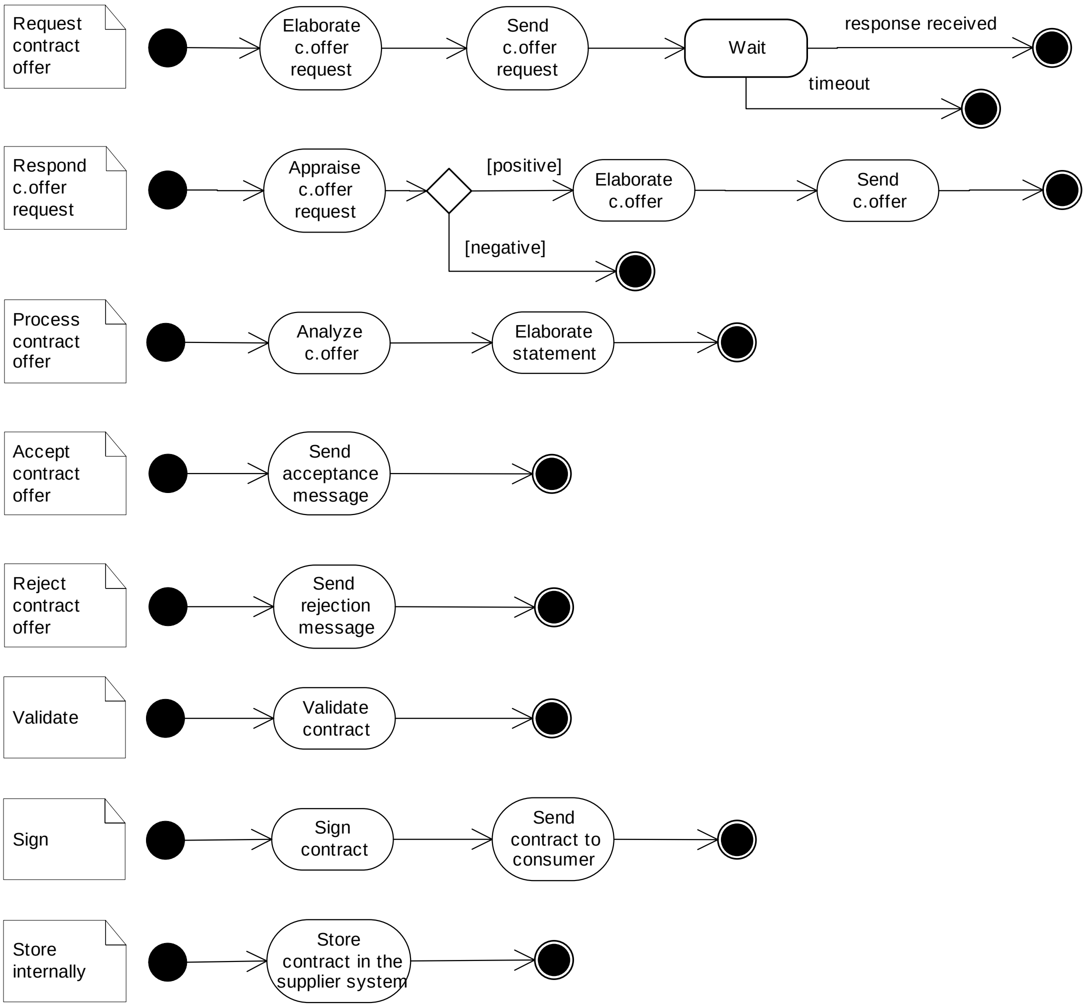




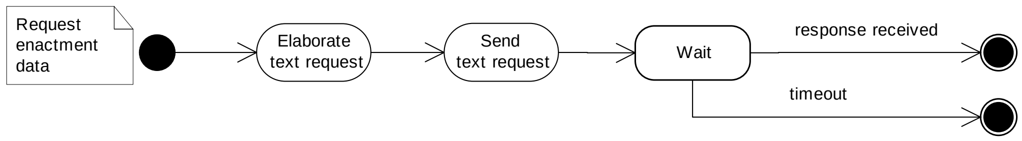


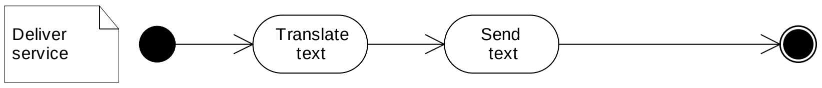

Respond customisation request
Appraise customisation request
[positive]
Elaborate customisation offer
Send customisation offer
[negative]

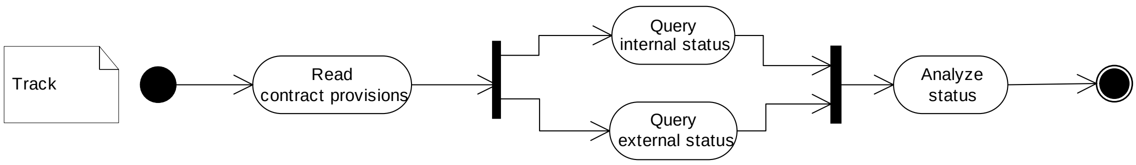
Contracting phase

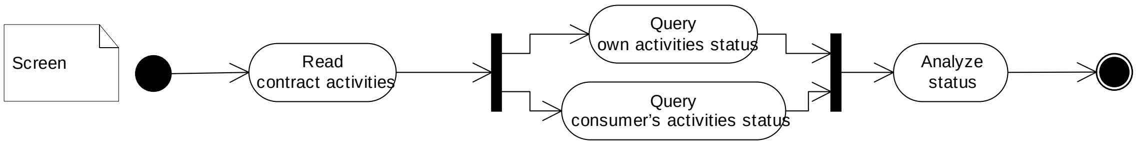


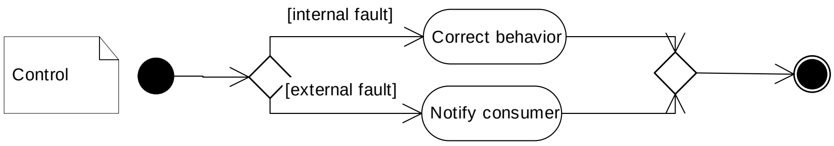
Enactment phase

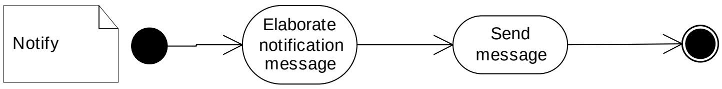


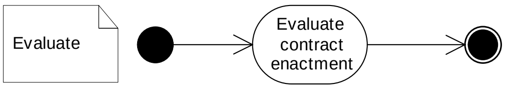


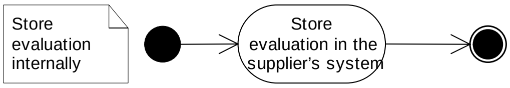


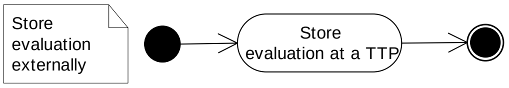

Screen
Read contract activities
Query own activities status
Query consumer's activities status
Analyze status








Store evaluation internally
Store evaluation in the supplier's system


Store evaluation externally
Store evaluation at a TTP