Business Process Outsourcing Modeling

Lai Xu
SAP Research Switzerland

Paul de Vrieze
SAP Research Switzerland

ABSTRACT

Organizations in the new millennium now face relentless pressure to perform better, faster, and cheaper, while maintaining high level of guaranteed results.To remain competitive, enterprises have to integrate their business processes with their customers, suppliers and business partners. Increasing collaboration includes not only a global multi-national enterprise, but also an organization with its relationship to and business processes with its business partners. Standards and technologies permit business partners to exchange information, collaboration and carry out business transaction in a pervasive Web environment. There is however still very limited research activity on modeling multi-party business collaboration underlying semantics. In this paper, we demonstrate that an in-house business process has been gradually outsourced to third-parties and analyze how task delegations cause commitments between multiple business parties. Finally we provide process semantics for modeling multi-party collaborations.

Keywords: electronic commerce; outsourcing; business process outsourcing(BPO); business collaboration; multi-party business collaboration; modeling business collaboration

INTRODUCTION

Outsourcing has been a worldwide phenomenon for the past four decades (Gereffi & Sturgeon, 2004). Growth of outsourcing is driven by a number of business forces, such as, competition escalation, organizational reengineering, and new technology trends. Over the past decade, the number and quality of suppliers offering price-competitive and high-quality services has increase significant. It is able the organization to focus on the company's main skills.

In addition, large sizes of organizations are not longer a necessary advantage in production of products or services, and neither are small size; quality, flexibility, agility and the ability to meet diverse consumer demands count for more (Drucker, 1992). Firms now respond to change by outsourcing when they face heightened competition pushes. Traditionally, after the part of business services is assigned, the initial organization can hardly monitor or get to control of the outsource services. Even a minor change of service is not such easy.

The advent of global digital networks, the Internet, the World Wide Web, and more recently Web services has drastically lowered the cost of coordination between firms and improved the possibilities for organizations and individuals to communicate in an effective and standard manner. New environment, newer technology and rapid technological change provides an avenue for reducing human and equipment resources that do not fit with a company's strategic direction, for meeting latest needs with up-to-date resources at competitive rates by outsourcing those business processes. Furthermore, the current technologies are also allow to get control of outsource business processes.

Rather than outsourcing an entire business process to a single supplier, multi-souring using more than one supplier is used. A classic example is Alcatel. Alcatel has outsourced supply chain management and R&D functions to Wipro, and its SAP and ERP environment work to Infosys (Pinto & Harms, 2005).

Business process multi-outsourcing causes business collaboration. As business collaboration increases between different enterprises, the need for semantics also increases as a mediator between the structure and content of the different knowledge based. There will be a need, not just for semantics to mediate the structure and content, but also for the services themselves. Semantics of multi-party business collaboration has been recognized as a major problem for a long time, but relatively little fundamental research has been devoted it. From the semantic perspective, we model the

way organizations co-operate in a multi-party involved situation. A high-level view of the collaboration is provided, in terms of the parties involved, the roles they perform and the way their related, also in terms of business functions they fulfill and the interactions between those.

In the rest of this section, we argue the reasons why multi-party business collaboration needs to be modeled. In the rest of this paper, we provide the definitions of outsourcing, business transactions and business collaborations. The following section starts by elaborating how an in-house business process has been gradually outsourced by using a multi-party business collaboration case. Moreover, we explain the issues in multi-party business collaboration modeling. We furthermore define all concepts needed for modeling multi-party business collaborations. We first introduce our meta-model of multi-party business collaboration language which defines the attributes and relationships of the modeling concepts. Then, multi-party collaboration language is given with a concrete graphical syntax. We also evaluate relate work in this area. The paper concludes with an summary and directions for further research.

The Benefits of Multi-party Business Collaboration Modeling

Outsourcing business processes is highly complex. They consist of several organizations interconnected through networks and working together using sophisticated computer applications. When trying to understand, reorganize or develop systems to support multi-party business collaborations one is confronted with that complexity. As in any modeling activity, modeling multi-party business collaboration can help to deal with this (Rechtin & Maier, 1997; Wolstenholme, 1990).

There are many possible reasons to create a model of multi-party business collaboration. The goal of multi-party business collaboration modeling may be

- to understand the functioning of an existing multi-party business collaboration
- to provide a starting point for analysis of requirements of design and for the redesign of an outsourcing business process.
- to offer a starting point for the implementation of computer applications to support multi-party business collaboration.
- to serve as a basis for analysis, for example, answer ``what-if'' question, to evaluate the responsibilities between involved parties, or simulate an inter-organizational business process before implementing it.

The benefits of using models to support these objectives, rather than a textual description, are as follows (Biemans et al., 2001; Booch et al., 1999):

- Models help us to visualize a system as it is or as we want it to be. it create a common frame of reference for those who have to understand a design, and facilitate the communication between clients, business analysts, designers and engineers.
- Models permit us to specify the structure or behavior of multi-party business collaboration by representing only their essentials and abstracting from what we consider irrelevant. Abstraction can be done in two ways: by abstracting from details or by abstracting from aspects.
- Models give us a template that guides us in constructing a system. Properties of models can be thus analyzed. This can be used to assess the expected performance of business collaboration, or check conformance to a set of business rules.
- Models document the decisions we have made, motivate changes or point alternatives.

Modeling is definitely true that the larger and more complex the system, the more important modeling becomes. Involving multi-party business collaboration is a complex systems. To be able to model this system, we have to understand concepts and issues in multi-party business collaboration which will be discussed in the following sections.

TERMINOLOGY

In this section, we will discuss terms like outsourcing, transaction, transaction properties, business transaction and business collaboration. Finally, we provide our definition of multi-party business collaborations.

Outsourcing is a generic term of farming out certain company activities or processes, usually performed by company employees, to external contractors specializing in such activities or processes. It is used in most branches of industry and services worldwide.

Transactions are a fundamental concept in building reliable distributed applications. A transaction is a mechanism to insure all the participants in an application achieve a mutually agreed outcome. Traditionally, transactions have held the following properties collectively referred to as ACID (Haerder & Reuter, 1983):

- Atomicity: if successful, then all the operations happen, and if unsuccessful, then none of the operations happen.
- Consistency: the application performs valid state transitions at completion.
- Isolation: the effects of the operations are not shared outside the transaction until it completes successfully.
- Durability: once a transaction successfully completes, the changes survive failure.

According to the International Organization for Standardization (ISO), a *business transaction* is "a predefined set of activities or processes of organization to accomplish an explicitly shared business goal" (ISO/IEC, 1997). A business transaction may involve any number of participants, it may be instant or last for years, and it can have various degrees of complexity.

Following the ebXML Business Process Specification Schema (ebXML, 2001), the concepts of business transaction and business collaboration are defined as follows,

- a *business transaction* involves two parties, and is an atomic unit of work that can result in either a success or a failure.
- a *business collaboration* can involve any number of parties and is a combination of choreographed business transaction, defining the ordering and transition between them.

According above ebXML definitions of business transaction and collaboration. It implicates that a collaboration is a number of two-party transactions. This assumption is not reasonable. For example, if there is a broadcast message or an activity benefits to multiple parties, this assumption can not hold anymore.

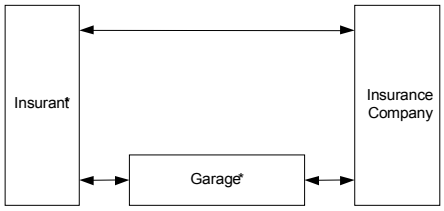
In the next section we will analyze *multi-party business collaborations* that involve more than two autonomous parties that are coordinated and lead to the accomplishment of some result. Looking at such a collaboration one can not just see it as a number of two-party transactions.

OUTSOURCE AND MULTI-PARTY BUSINESS COLLABORATIONS

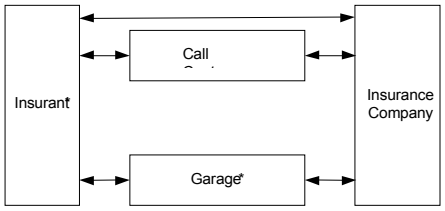
We provide a car insurance case for explaining how a car insurance business is gradually outsourced and in which collaborations are involved afterward. This case is an abstract version of a car insurance scenario involving AGF Irish Life Holdings Plc. It was taken from the CrossFlow project (Browne & Kellett, 1999).

At the start time, a car insurance company probably only involves a group of garages to assess car damages and to repair damaged cars for an insurant who has bought car insurance from the car insurance company. The insurance company deals with the rest of the issues. More precisely, after the occurrence of car damage, a process starts, including many interactions among the insurant, a garage and the insurance company (see Figure1(a)).

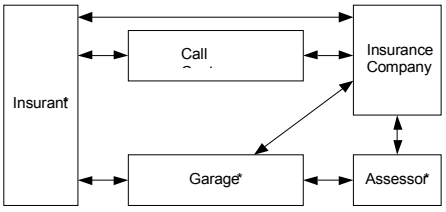
Figure 1: Car Insurance Business Processes



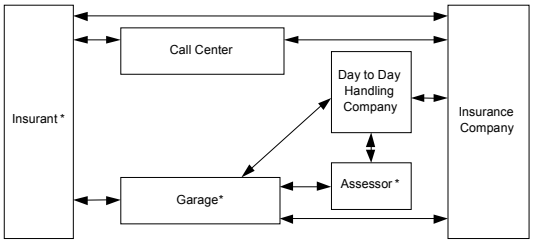
(a) Initial business model



(b) Outsourcing the call center



(c) Outsourcing damage assessment



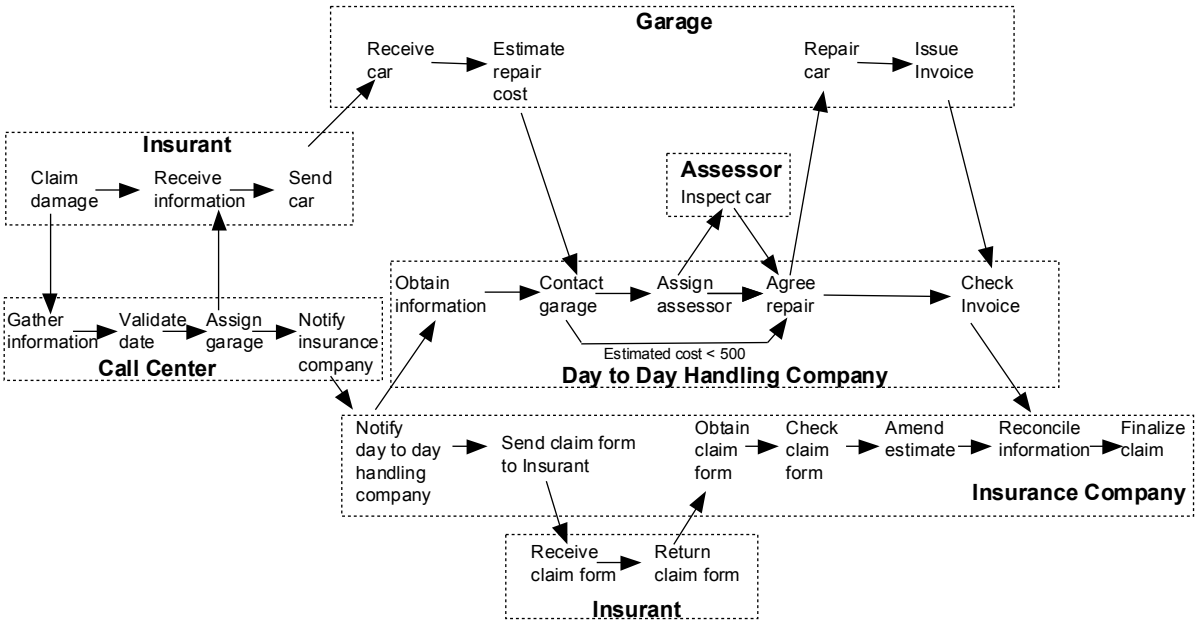
(d) Outsourcing day to day handling

After some time, the insurance company decides to outsource its phone service to a call center. The business process is consequently changed (along the line of Figure1(b)). The call center is responsible for registering the insurant information, suggesting an appropriate garage (most time a close by garage is assigned) and notifying the insurance company about the insurant's claim. Except the phone service, the insurance company still needs to handle the rest of services for the insurant.

Continuing it could be an alternative to outsource the inspection of damaged vehicles to an association of assessors. In this business model (see Figure 1(c)), the assessors conduct the physical inspections of damaged vehicles and agree upon repair figures with the garages. After the call center, the garages and the assessors finish their obligations, the insurance company performs the rest services.

Due to the increasing amount of different insurance businesses, the insurance company might finally decide to outsource the daily service to a day to day handling company. The day to day handling company coordinates and manages the operation on a day-to-day level on behalf of the insurance company (see Figure1(d)). The detailed obligations of the day to day handling company are provided in Figure 2. After receiving the forward claim from the insurance company, the day to day handling company will agree upon repair costs if an assessor is not required for small damages; otherwise, an assessor will be assigned. After finishing repairs, the garage will issue an invoice to the day to day handling company, which in turn will check the invoice against the original estimate. The day to day handling company returns all invoices to the insurance company monthly. As a result the workload of the insurance company is significantly reduced.

Figure 2: The Process Diagram of Car Insurance Case



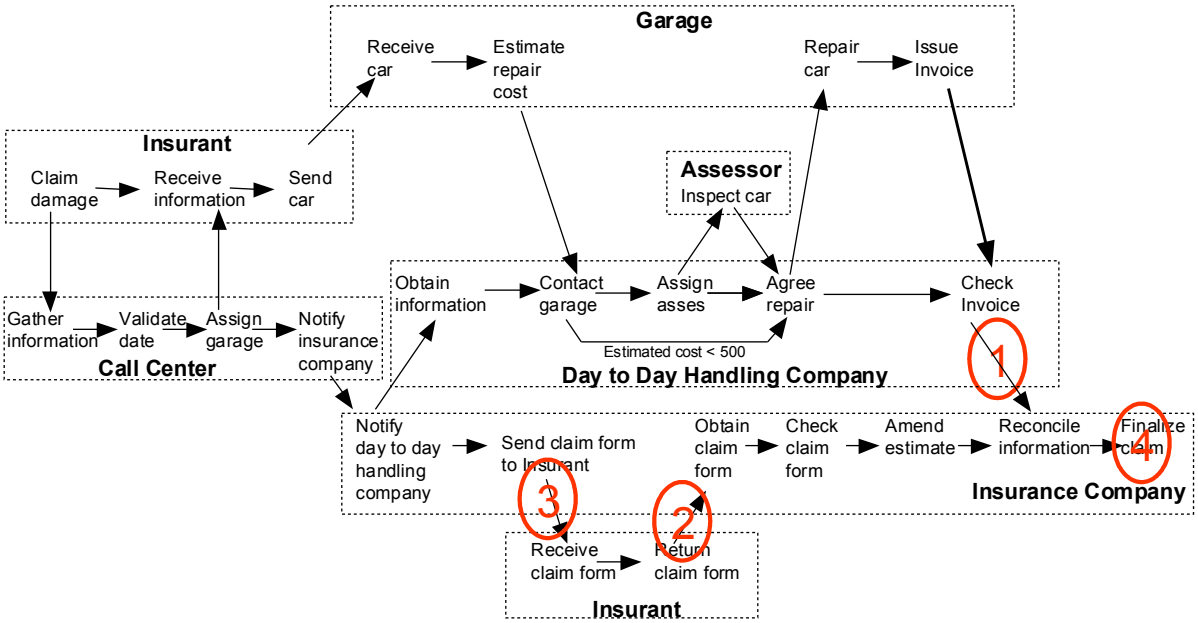
Changes of the business models do not necessarily go through from Figure1 (a) to (b), then from (b) to (c) and finally from (c) to (d). Changes can happen, for example, directly from (a) to (d). Figure1 demonstrates that how a business process is collaborated by more business parties in different circumstances. It also shows some essential characters of multi-party collaborations. One of them is that it is critical to understand *when* and *who* did, is doing or will do *what* in a multiple parties involved business process.

ISSUES IN MULTI-PARTY BUSINESS COLLABORATION MODELING

A distinction between a bilateral business process and multi-party business collaboration is the way of interactions. In a multi-party collaboration, many business activities which involved different parties can be parallel performed. The orders of interactions are no longer sequential. Therefore to use absolute orders to specify the collaboration which many bilateral collaborations are specified in this way, is not suitable for specifying multi-party collaborations.

In addition, because different parties can operate different activities at the same time, it is difficult to find all violators responsible for even an unsatisfied performance or a mis-performence. Therefore, modeling multi-party business collaborations are different with a bilateral collaboration modeling. Because in total only two parties involved in a collaboration, modeling a bilateral collaboration is not a important issue to recognize who does not do what. In a multi-party collaboration, it is however an important concern.

Figure 3: The Violation of Car Insurance Case



There are many potential violations in the case as outlined in (Xu & Jeusfeld, 2004; Xu et al., 2005). For example, after sending invoices to the day to day handling company; the garage does not get money back from the insurance company. It could be caused by

- the day to day handling company, because the day to day handling company did not forward the invoices to the insurance company (see Figure 3 mark 1);
- the insurant, because the insurant did not return the completed claim form to the insurance company (see Figure 3 mark 2);
- the insurance company, because the insurance company forgot to send the claim form to the insurant (see Figure 3 mark 3) or simply because the insurance company did not pay the garage in time (see Figure 3 mark 4).

or some combinations of the above. Therefore a model of parties' responsibilities is needed. In multi-party business collaboration, one contract violation can be caused by more than one contractual party, which motivates us to model multi-party business collaboration from different perspectives.

CONCEPTS FOR MODELING MULTI-PARTY BUSINESS COLLABORATION

Many business applications can involve multi-party business collaboration like supply chains, electronic markets, online auctions, virtual enterprises and multi-supplier business process outsource. The central concept of modeling multi-party business collaboration is that multi-party business collaboration is not to model a number of bilateral business collaborations. In some simple cases, a chain relationship between multiple parties for example, it is possible to break down a multi-party business collaboration into a number of bilateral business transactions. However, in complicated multi-party collaborations, this conversion results in information of relations being lost or hidden. Consequently this option to split the multi-party collaborations up into several bilateral transactions won't work. Especially when something is going wrong during automation of multi-party business collaboration, to find all violators is a big challenge.

During a bilateral collaboration execution process it is easier to discover the responsible party for an existing violation. In multi-party business collaboration, an existing violation can be as a result of a set of directly or indirectly involved parties (some examples are provided from the previous section). This thus raises the issue of finding all responsible parties for an unsatisfied or mis-performance in the execution of multi-party business collaboration. Those concepts should be reflected to our modeling of multi-party business collaborations.

The use of modeling has a rich history in all the engineering disciplines. One of four basic principles of modeling is that no single model is sufficient (Booch et al., 1999). Every nontrivial system is best approached through a small set of

nearly independent models. Following this basic principle, we distinguish three aspects that are relevant for modeling multi-party business collaboration.

- in terms of its structure: which are the parties involved and how are they interconnected?
- in terms of the commitments associated with those parties: which responsibility do they commit to each other and which roles may the parties play within a commitment?
- and in terms of its processes: what actions are performed by which parties after which properties are satisfied according to per-defined rules and which output can be expected, etc.

Parties and Channels

In order to survive, companies are forced to constantly revise their business processes. Due to decision of multi-supplier business process outsource, involved parties and interconnection are modeled as parties and channels. Depending on the chosen scope of the model, *parties* may represent individual people, organizational units, such as departments, or an entire organization. Furthermore, A *party* represents an active organizational entity. Parties can be assigned a name to distinguish them.

Parties may be connected by *channels* through which they interact. For example, parties may be linked by internet, by telephone or by truck. Usually channels can connect two parties, but multi-access channels can also be envisioned, for example a broadcast channel. Channels can further specified as bi-directional, uni-directional channels. A *channel* thus represents an interconnection between parties for the exchange of objects or information. Channels may be named by the medium (e.g. Internet, Post, EDI, etc.)

Commitments and Roles

To model responsibilities between multi-parties, we provide speech act theory and its extension. Applying those theories we provide our definition of commitments and roles.

Part of Austin's work on speech act theory (Austin, 1975), is the observation that utterances are not implied propositions that are true or false, but attempts on the part of the speaker that succeed or fail. Perfomatives, acts, or actions are organized as speech acts and non-speech acts. An individual speech act is either a *solicit*, which explains an attempt to achieve mutual belief with the addressee that the sender wants the addressee to perform an act relative to the sender's wanting it done, or an *assert*, which expresses an attempt to achieve mutual belief with the addressee that the asserted statement is true.

The model of speech acts and repartee developed by Longacre recognizes two kinds of relations among successive utterances: replay and resolution in (Longacre, 1976) and another two kinds of relations: resolves and completes by Van Dyke Parunakin in (Van Dyke Parunak, 1996). In the model of speech acts and repartee, every utterance in a conversation except for the first must “respond”, “reply”, “resolve” or “complete” to another, otherwise there would be no conversation. Analyzing relations between utterances some characters can be split (Dooley, 1976).

In the business process domain physical actions and messages convey information between participants. An initial proposal can be triggered by a certain action and later on be finished by another action. During multi-party business collaboration, several proposals are initiated by different business parties. Each of them is followed and eventually finished by some actions. Actions are thus sorted into different commitments.

A *commitment* is a guarantee by one party towards another party that some action sequences provided that some “trigger”, “involve”, or “finish” action happens and all involved parties fulfill their side of the transactions (Xu & Jeusfeld, 2003; Xu, 2004; Xu & Jeusfeld, 2004). To finish a commitment, more than one party must finish relevant actions. A commitments is one party towards a guarantee to another party. It allows more than one proposal in a commitment.

From this point of view, the concept of our commitment is different from the definition of a commitment in papers (Verdicchio & Colombetti, 2002; Ervin, 2002), where a commitment only refers to two parties, a debtor and a creditor (Verdicchio & Colombetti, 2002), or a vendor and customer (Ervin, 2002). The notion of commitment in this paper is not related to beliefs, desires, or intentions. In the research of Cohen and Levesque (1995), commitments are related to establishing common beliefs about a certain state of the world.

All parties involving multi-party business collaboration fulfill different roles in different commitments. One role cannot be part of another role. However, it is possible for an actor to fulfill multiple roles in business collaboration. Each role is specified by using different names. Some examples, such as how to identify commitments, which party plays which role is going to show.

Interaction and Properties of Actions

Multi-party interaction models are composed of actions, commitments, properties of party and connectors. An action is the atomic unit of behavior. The orders between actions are causal, conjunctive “splits”, disconjunctive “splits”, conjunctive “joins” and disconjunctive “joins”.

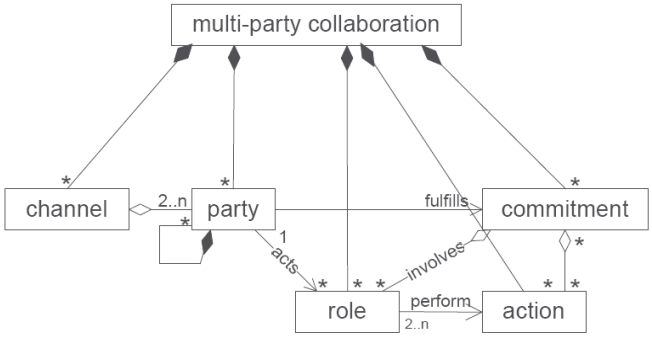
Each party has a set of per-determined properties. The party's properties consist of three parts which are inputs, outputs and rules. The inputs and outputs of properties are domain related. The input of the property specifies a list of the domain elements, which will be formulated as pro-conditions. The rules of property are specified as the set of rules using predicate logic. The rules of properties provide a logic formula to specify the pro-conditions of fulfilling the action and also some domain elements as a consequence of performing the actions. The output of the property specified those domain objects which are the results of the executing action. When the party attempts to execute an action, it first checks whether the input of the property is satisfied, and subsequently generates the output of the property according the rules.

METAMODEL OF MULTI-PARTY BUSINESS COLLABORATION MODELING LANGUAGE (MBCML)

In this section, we define the metamodel for MBCML. A concrete syntax is proposed in the next section. MBCML is used to define multi-party business collaboration models (MBCMs). A MBCM define the cooperation of a number of organizations to involve a common business process. In order to capture all relevant aspects of collaboration. A variety of diagrams can be used, each offering a different view on the model. We distinguish three different diagram types, one for each conceptual domain which has defined respectively. The notation used for representing the metamodel of MBCM is the Unified Modeling Language (UML) (Booch et al., 1999).

An overview of the basic modeling concepts and their relationships are given as a metamodel in Figure 4. A multi-party collaboration consists of role, party, channel, commitment and actions. Parties perform different roles in a commitment and fulfill different commitments. The roles perform actions. A channel connects two or more parties. A commitment aggregates many actions.

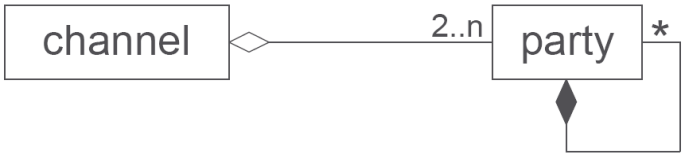
Figure 4: Metamodel for Multi-party Collaboration



Multi-party Collaboration Structure Metamodel

Parties involved and channels connected all parties of business collaboration are interrelated in Figure 5. How the parties are link by the channels are presenting in a collaboration structure model. A party may contain other parties. A channel connects two or more parties, and as an option, attributes like medium and direction can be indicated.

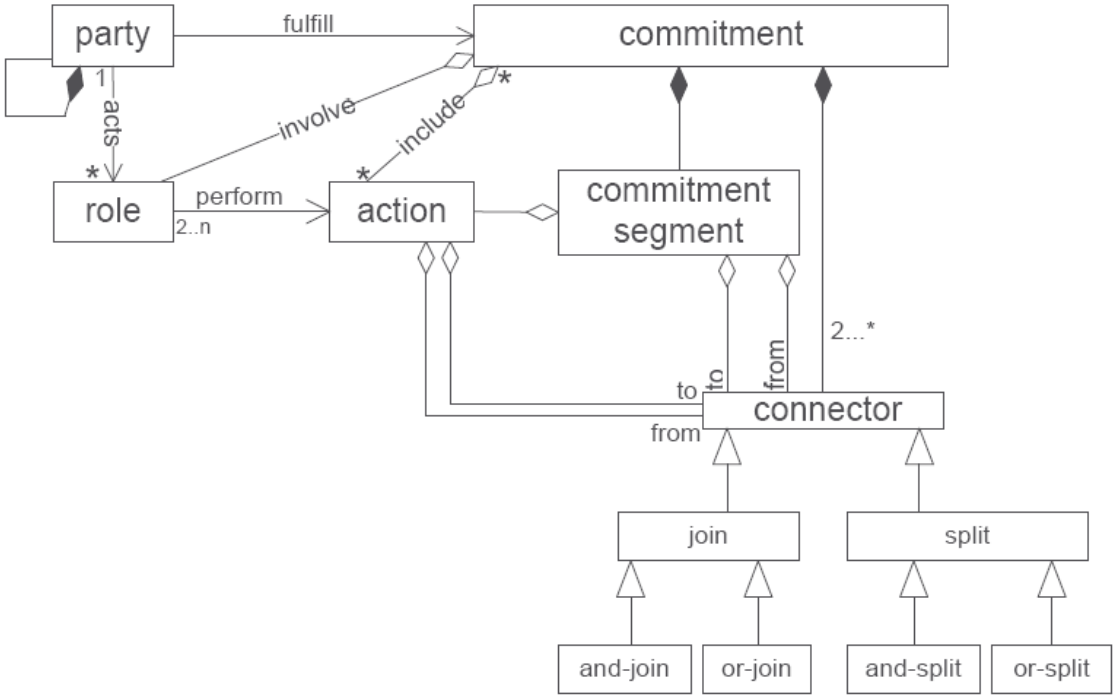
Figure 5: Metamodel for Multi-party Collaboration Structure



Multi-party Collaboration Commitment Metamodel

Commitments and roles construct a commitment model. All concepts involved in the commitment model are roles, actions, commitments, commitment segments and connectors. The relationship between those concepts are provided in Figure 6.

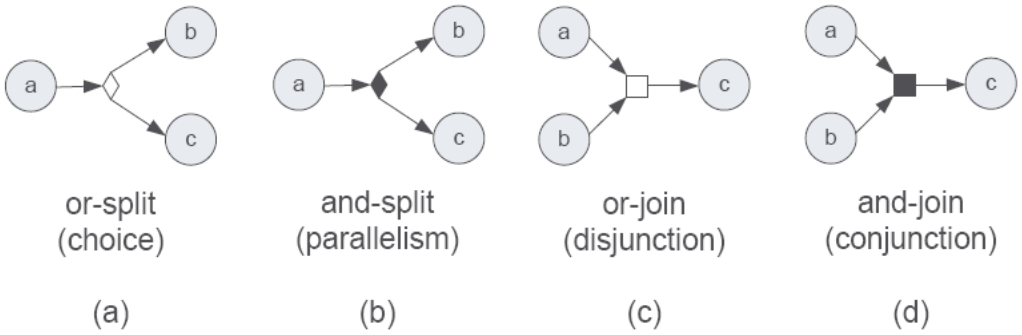
Figure 6: Metamodel for Multi-party Collaboration Commitment



A party guarantees another party to perform a commitment to another party. A party acts different roles in different commitments. A commitment is a set of actions and is made up of one or more commitment segments. An action contains zero or more connectors that interconnect these commitment segments to make up the commitment. A role (roles) performs (perform) an action within a commitment.

The connectors are used to introduce relationships between commitments or commitment segments. There are four notations for specifying the connectors. The connectors can be specialized to “and-join” connector, “or-join” connector, “and-split” connector and “or-split” connector. Commitments can be linked by causality, or-split, and-split, or-join, and and-join as Figure 7. Besides, the connectors may contain other connectors to specify more complex relationships. The causality relation is represented by an *arrow* from one node to another node. The “or-split” relation is represented by an *empty-diamond*, which means that a commitment from a role triggers exactly one of multiple commitments from other roles (see Figure 7(a)). The ``and-split'' relation is expressed by a *solid-diamond*, which means that a commitment from a role triggers other multiple commitments (see Figure 7 (b)). The “or-join” relation is denoted by an *empty-box*, which means that one of multiple commitments triggers another commitment (see Figure 7(c)). The “and-join” relation is shown by a *solid-box*, which means that multiple commitments together trigger a commitment (see Figure 7(d)).

Figure 7: Connector Representation

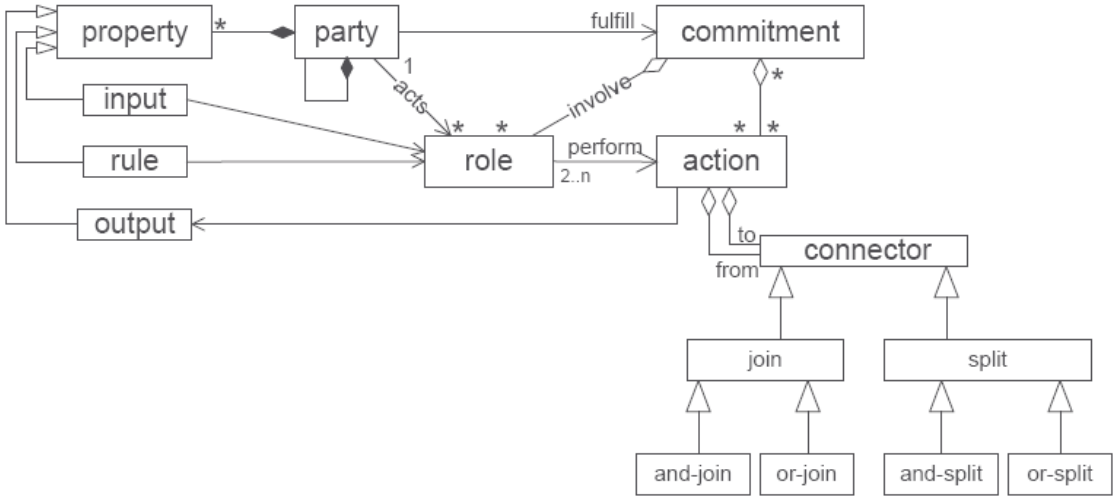


A multi-party business collaboration consists of a set of commitments. A collaborating party can thus be involved in different commitments playing different roles and an action may be involved in more than one commitment.

Multi-party Collaboration Interaction Metamodel

The part of the metamodel concerning interactions is refined in Figure 8. Interaction model consists of actions, connectors, parties, properties of parties. The property of a party includes inputs and rules and outputs.

Figure 8 : Metamodel for Multi-party Collaboration Interaction



Each party has a set of per-determined properties namely inputs, outputs and rules. After the input property of a party is satisfied, the role of the party performs the action(s) and the result of actions will provide some output. Besides, the actions can be also linked by the connectors which are provided in Figure 7. Those connectors represent the occurrence relationship of the actions.

MULTI-PARTY COLLABORATION MODELING LANGUAGE

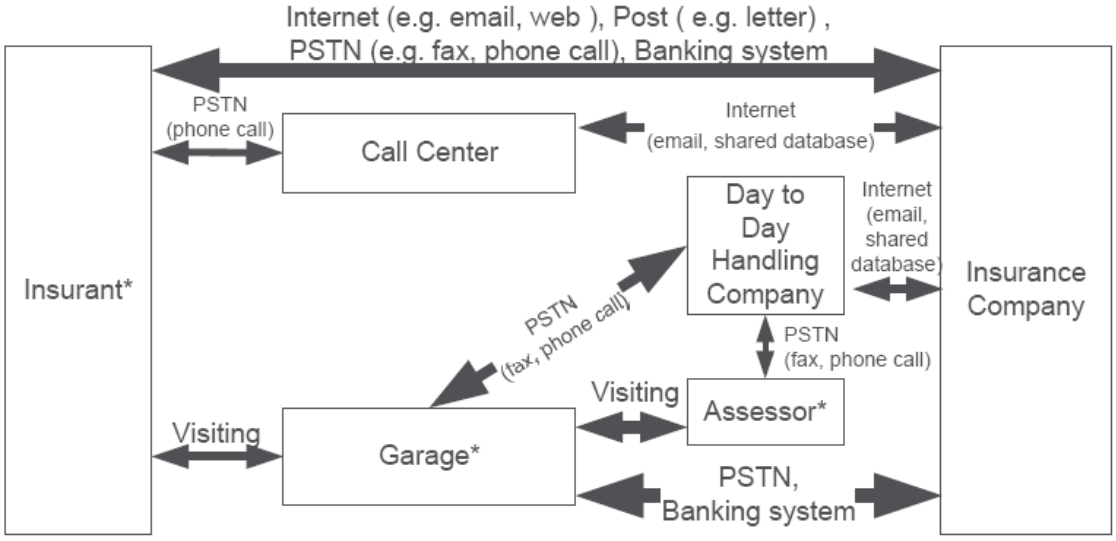
In the business domain, we need to provide detailed and precise descriptions of multi-party business collaborations. In order to represent construct in the business domain, a language for modeling multi-party collaborations should be sufficiently expressive to represent a multi-party collaboration in terms of its structure, commitments and interactions. We separate views to represent a structural model, a commitment model and a interaction model in the following sections respectively.

Collaboration Structure Model

Collaborated parties involved and channels in a business process are modeled in collaboration structure model. Figure 9 shows collaboration structure model for the six parties involved care insurance case. It depicts how the insurance

company and its co-operators are interconnected. It is an important model for future mapping to a specific implementation suits/toolkit.

Figure 9 : Collaboration Structure Model



For example, the insurance company and its insurant can be connected by internet such as, web and email; by post office so they could send a letter to each other; by telephone network like they could use fax or phone call to contact each other, or by a bank system which means they can transfer money. It is not always necessary to have all communication channels. It is up to the custom requirements. However, for each communication channel, it does require different implementation.

Modeling collaboration structure is useful to identify the parties involved in business collaboration. It also provides a further step to clarify the responsibilities of parties.

Commitment Model

To model commitments between multiple parties, we use concepts of commitments. The approach of how to determine the commitments in business collaboration which have introduced in Section C*ommitments and Roles*.

We use the six parties involved in the car insurance case (see Figure 2) to explain how to determine the commitments. For example an insurant phones a call center for a claim. Action A_phoneClaim triggered a conversation between the insurant and the call center to deal with the claim. Actions A_sendInfo and A_assignGarage follow, and action A_notifyClaim finishes the conversation. Actions A_phoneClaim, A_sendInfo, A_assignGarage and A_notifyClaim are sorted within a commitment which records obligations of the call center to deal with the insurant claim. Action A_phoneClaim is a trigger of a conversation; actions A_sendInfo and A_assignGarage are the actions as "involved"; and action A_notifyClaim is an action to “finish” the conversation.

Furthermore, six commitments of the care insurance case are identified according to the theory introduced in Section *Commitments and Roles*.

Table 1: Commitments, Actions and Action Abbreviations

Commitment	Classification of Actions and Commitments			Labels
	Trigger	Involved	Finish	
C_phoneService (PS)	A_phoneClaim			PS.1
		A_sendInfo		PS.2
		A_assignGarage		PS.3
			A_nitifyClaim	PS.4, CF.1, DS.1

C_repairService (RS)	A_sendCar			RS.1
		A_estimateRepairCost		RS.2
	A_agreeRepairCar			RS.3, DS.7
			A_repairCar	RS.4, DS.8
C_claimForm (CF)	A_notifyClaim			CF.1, PS.4
		A_sendClaimForm		CF.2
			A_returnClaimForm	CF.3, PR.2
C_dailyService (DS)	A_notifyClaim			DS.1, PS.4, CF.1
		A_forwardClaim		DS.2
		A_contactGarage		DS.3
		A_sendRepairCost		DS.4
		A_assignAssessor		DS.5, IC.1
		A_sendNewRepairCost		DS.6, IC.3
			A_agreeRepairCar	DS.7, RS.3
	A_repairCar			DS.8, RS.4
		A_sendInvoices		DS.9
			A_forwardInvoice	DS.10, PR.1
C_inspectCar (IC)	A_assignAssessor			IC.1, DS.4
		A_inspectCar		IC.2
			A_sendNewRepariCost	IC.3, DS.5
C_payRepairCost (PR)	A_forwardInvoices			PR.1, DS.10
	A_returnClaimForm			PR.2, CF.3
			A_payRepairCost	PR.3

According to Table 1. There are six commitments.

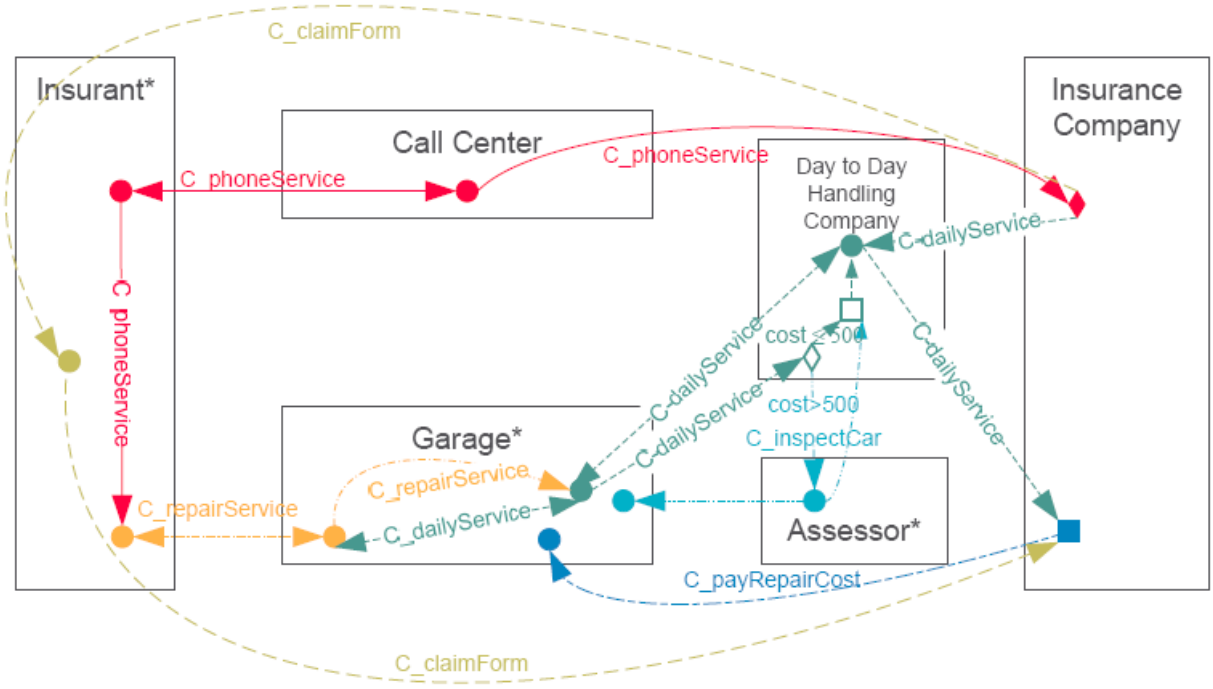
C_phoneSercie {(A_phoneClaim, *tr*), (A_sendInfo, *in*), (A_assignGarage, *in*), (A_notifyClaim,*fi*)}
C_repairService {(A_sendCar, *tr*), (A_estimateRepariCost, *in*), (A_agreeRepairCar, *tr*), (A_reparicar, *fi*)}
C_claimForm {(A_notifyClaim, *tr*), (A_sendClaimForm, *in*), (A_returnClaimForm, *fi*)}
C_dailyService {(A_notifyClaim, *tr*), (A_forwardClaim, *in*), (A_contactGarage, *in*), (A_sendRepairCost, *in*), (A_assignAssessor, *in*), (A_sendNewRepairCost, *in*), (A_agreeRepairCar, *fi*), (A_repairCar, *tr*), (A_sendInvoices, *in*), (A_forwardInvoices, *fi*)}
C_inspectCar {(A_assignAssessor, *tr*), (A_inspectCar, *in*), (A_sendNewRepairCost, *fi*)}
C_payRepairCost {(A_forwardInvoices, *tr*), (A_returnClaimForm, *tr*), (A_payRepairCost, *fi*)}

In commitment C_dailyService, actions A_agreeRepairCar and A_forwardInvoice are both marked as "finished". Action A_agreeRepairCar finishes a proposal by action A_notifyClaim and action A_forwardInvoice finishes a proposal by action A_sendInvoice.

It is difficult to represent commitments graphically. In our commitment model, a party is represented as a rectangle with a name. A node denotes a role which should stay in a rectangle. A commitment is indicated by a set of nodes (or commitment connectors) and a set of arrows.

Figure 10 depicts the six parties and six commitments of the case in Figure 9. For example, in party "insurance company", the solid-diamond connects commitments C_phoneService, C_claimForm and C_dailyService. It means that after commitment C_phoneService is fulfilled, both commitments C_claimForm and C_dailyService are triggered. A solid-box is also in party "insurance company", it connects commitments C_dailyService, C_claimForm and C_payRepairCost. It means that commitment C_payRepairCost will be performed after commitments C_dailyService and C_claimForm.

Figure 10 : Commitments of Multi-party Collaboration Model



In the commitment model, we present the position of each commitment in terms of which parties and roles are involved, and which commitments are triggered, chose and paralleled by other commitments. In the commitment model, we provide an overview of each party's responsibilities. This is very important for both the business side and the IT side as it helps creating a common understanding. In the next section, the behavior of the parties is modeled.

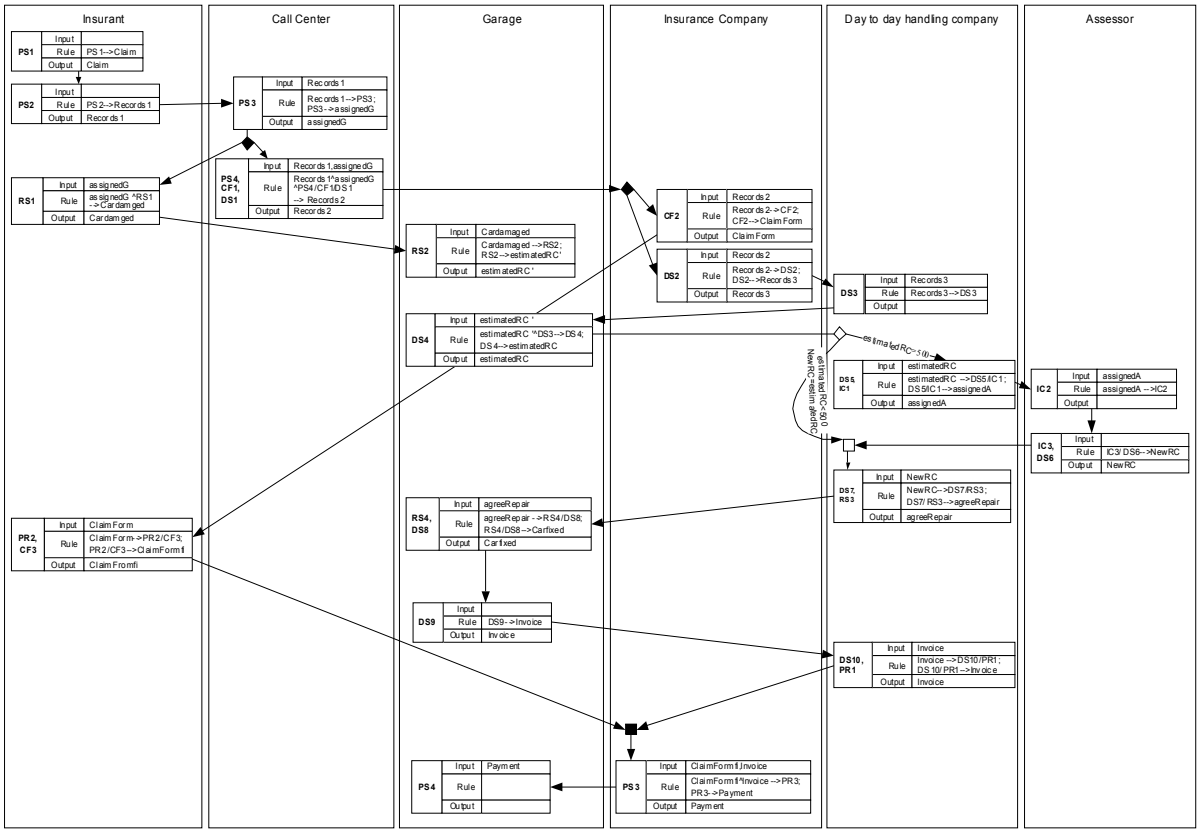
Modeling Behaviors of the Parties

In interaction collaboration modeling, behavior is modeled as inter- or intra-organizational business processes. The vertical dimension is the time axis; time proceeds down the page. Each party is represented by a vertical column. An action is the atomic unit of behavior. The causal ordering between actions is modeled by conjunctive and disconjunctive “splits” and “joins”.

Each party's behavior is determined by three parameters as found from the business collaboration. The inputs and outputs of a party are domain related. The rules of a party in our model are specified using predicate logic. The input parameter specifies the actions that this party expects to be involved in as object, while the output parameter specifies results of the action. When a party attempts to execute an action, it first checks whether the current input can trigger this action and subsequently generates the output which may be checked against the possible output.

In Figure 12, an action is noted as a table-box (see Figure 11), where the first column shows the action label which makes it possible to determine the commitment from this action label by looking it up in Table 1; the rest column shows the parameters involved in this action.

Figure 12: Interactions Between Parties



The diagram includes a multi-step interaction between participants. It clearly shows which parties will communicate with other parties for which matters. From the insurant's perspective, he only has contact with the call center, the assigned garage and the insurance company. From the insurance company's view, it receives and forwards the claim to the day to day handling company, sends the claim form to the policyholder, and finally pays the repair costs to the garage.

Each party parameter is also included in the diagram. According to the party properties, each party can determine which actions should or may occur. For example (see Figure 11) after having received an input from the party property ``Records2", the insurance company will perform actions A_sendClaimForm (as *CF2*) and A_forwardClaim (as *DS2*) according to rules ``Records2 → CF_2" and ``Records2 → $DS_2$ " respectively that are joined by ``and-join" connector. After finishing action A_sendClaimForm (as *CF2*), ``Clam from" is sent. After performing action A_forwardClaim, ``Records3" is recorded.

Figure 11: Examples of Actions and Their Parameters

CF2	Input	Records2
	Rule	Records2 → CF2; CF2 → ClaimForm
	Output	ClaimForm

DS2	Input	Records2
	Rule	Records2 → DS2; DS2 → Records3
	Output	Records3

As time passes (from top to down) and satisfying the party parameters, each participant takes actions while the business process is moving forward. A complex multi-party business process is divided into multiple commitments. Furthermore, business process steps are represented as the actions exchanged between the parties.

In this section, we present multi-party business collaborations from three aspects. At the collaboration structure model, we provide a view how business parties are connected. Different connects can determine different ways to collaborate. At the commitment model, the responsibilities of all involved parties are presented. Furthermore, the behavior model provides more details of commitment fulfillment.

RELATED WORK

Much work is available concerning representation of computer processes. Processes for computer systems are commonly represented using flow charts, data flow diagrams, state transition diagrams (which define finite state machines, push down machines, or even Turing machines), Petri Nets, and specialization.

A flow chart represents a process as a series of steps, with arrows between them, which represents an order in which the steps are to be performed. Some of the steps are decision points, so depending on the circumstances, different sets of steps might be performed. A data flow diagram is similar but represents how modules of a system are interconnected to perform the steps of a process but focuses on the ordering relationships imposed by the fact that data produced by some modules is used by other modules.

A state transition diagram represents a process in terms of the possible states of the system. The steps taken in the process are the transitions that move the system from one state to another. The most powerful representation which included a state transition diagram is a Turing machine, which can be used to describe any computer executing any computer program written in any computer programming language. A Petri net is similar to a finite state machine, but allows multiple states to be marked simultaneously. Transitions between states may be synchronized, since multiple states have to be marked at the same time for a particular transition to occur.

There are few works concerning multi-party business collaborations. We provide the limitation of UML and Petri Nets for multi-party business collaboration modeling, as well as other models such as ebXML BPSS, web services choreography and SAP C-Business Scenarios.

The UML has been used in enterprise and business process modeling (Eriksson & Penker, 2000; Jacobson et al., 1995; Marshall, 1999). According to (Steen et al., 2002), the UML is not suitable for modeling business. The UML does not support well all the concepts needed for business collaborations. Modeling business collaborations is mainly concerned with what happens at the business level and how it is organized. First, the UML does not support expressing responsibilities. In the business world, parties permit the commitments to each other to execute a business activity. In our approach, the commitment model represents collaborations between business parties and provides the relations of the commitments. Second, for business party behavior, business processes are normally not confined to single actions. Therefore, state diagrams (also for Petri Net) are not really useful here. Most business collaborators do not reason about

processes in terms of states but rather in terms of the activities performed and results produced. Although UML can be extended with stereotypes and profiles, the stereotyping mechanism just makes it possible to introduce new subtypes of existing concepts and profiles cannot introduce new concepts either (Henderson-Sellers, 2001).

Using the Petri Nets to specify a multi-party business collaboration process, the amount of states of the Petri Net can be significantly increased. Especially, because the multi-party business collaboration process focuses on when and who did, is doing or will do what. A Petri Net representation can be too trivial, even by using state-based workflow patterns (van der Aalst et al., 2003) because of a big amount of possible combinations of multi-party's behavior.

Two models for ebXML BPSS multi-party collaboration and web services choreography are presented in (Dubray, 2002; Webber, 2004) respectively. Other research (Haugen, 2002) on multi-party collaboration tries to break down a multi-party collaboration into a number of bilateral relations. A principle cause behind this is that current e-commerce environments only support bilateral executions. In some simple cases, the approach to support multi-party collaboration execution in current e-commerce environments is to assume the whole business process runs correctly according to a number of bilateral relations. However, in complicated multi-party collaborations this conversion results in information of relations being lost or hidden. Consequently this option to split the multi-party collaborations up into several two-party relations will not work for these complex multi-party collaborations (Dubray, 2002; Haugen, 2002).

SAP's collaborative business scenarios describe inter-enterprise business processes from three different perspectives (SAP, 2000), namely business view, interaction view and component view. The purpose of the business view is showing the business advantages of implementing a collaborative business scenario. Business relations per se are out of the scope of our research through. The interaction view describes the process design and detailed dependency relationship between the different activities and responsibilities of the participants. It is too simple to describe the relationship like the action relations with conjunctive and disconjunctive “splits” and “joins”. The component view describes the logical application components needed to support the business process. Different channels in a collaboration structure model can determine different ways to implement a multi-party collaboration. Commitment model and interaction model provide enough details of interactions between multi-parties. Those three collaboration models can easily map into a component level model by using specifically software implementation packages.

CONCLUSIONS

We have looked at a multiple supplier business process outsource case. In the current business process outsource case, the outsourcing initiator should be able to get control for the whole business process and its co-operators should also be able monitor the business process.

Moreover, in the modern business world, we see that explicit structural collaboration between organizations is becoming more and more important. This is reflected in the emergence of tightly-coupled supply chains, the service outsourcing paradigm, complex co-makerships, etceteras. Collaboration is not limited by geographical proximity, but is of an increasingly international character. As a result explicit multi-party business collaboration is becoming global. The need for a multi-party collaboration model for a business process is thus becoming evident. Our modeling approach can be applied in those cases.

Nevertheless most of the past initiatives have addressed only partial aspects of the problem, failing to understand and properly support the various business entities and their inter-relationships in complex and fast evolving business environments. In this paper, we have present multi-party business collaboration models from three perspectives for supporting monitoring multi-party business process. In the collaboration structure model, we provide a view of how business parties are linked. Different links can determine different ways of collaboration. In the commitment model, the responsibilities of all involved parties are presented. Finally, the behavior model provides details of commitment fulfillment. Further research has to map our multi-party business collaboration model to specific implementations like SAP or Baan' ERP systems. This would allow the semantics of the web of collaborating parties to be validated. This paper introduced the approach and current development towards the stated goals.

REFERENCES

Austin, J. L. (1975). *How to do things with words*. Oxford University Press, 2 edition.

Biemans, F., Lankhorst, M., Teeuw, W., and van de Wetering, R. (2001). Dealing with the complexity of business systesm architecting. *System Engineering*, 4(2).

Booch, G., Rumbaugh, J., and Jacobson, I. (1999). *The Unified Modeling Language User Guide*. AddisonWesley, Inc.

Browne, S. and Kellett, M. (1999). Insurance scenario description. Technical Report CrossFlow project deliverable: D1.a, CrossFlow consortium.

Cohen, P. R. and Levesque, H. J. (1995). Communicative actions for artificial agents. In Lesser,V. and Gasser, L., editors, *Proceedings of the First International Conference on Multi-Agent Systems*, pages 65–72, San Francisco, CA, USA. The MIT Press: Cambridge, MA, USA.

Dooley, R. (1976). *Repartee as a Graph*. Appendix B in [Longacre, 1976].

Drucker, P. (1992). The new society of organizations. *Harvard Business Review*.

Dubray, J.-J. (2002). A new model for ebxml bpss multi-party collaborations and web services choreography. http://www.ebpml.org/ebpml.doc.

ebXML (2001). Uncefact and oasis, ebxml business process specification schema v1.01.

Eriksson, H.-E. and Penker, M. (2000). *Business Modeling with UML: Business Patterns at Work*. John Wiley & Sons.

Ervin, R. (2002). Chains of commitment software architecture. *ACM SIGecom Exchanges*, 3(1).

Gary Gereffi and Timothy J. Sturgeon (2004). Globalization, employment and economic development. *Sloan Workshop Series in Industry Studies Rockport Massachusetts*, June 14-16, 2004.

Haerder, T. and Reuter, A. (1983). Principles of transaction-oriented database recovery. *ACM Computing Surveys*, 15(4):287–317.

Haugen, B. (2002). Multi-party electronic business transactions. http://www.supplychainlinks. com/ MultiPartyBusinessTransactions.PDF.

Henderson-Sellers, B. (2001). Some problems with the uml v1.3 metamodel. In *the 34th Annual Hawaii International Conference on Systems Sciences (HICSS-34)*.

ISO/IEC (1997). 14662, information technologies-open-edi reference model.

Jacobson, I., Ericsson, M., and Jacobson, A. (1995). *The object advantage: business process reengineering with object technology*. Addison-Wesley.

Jude Pinto and Brigitte Harms (2005). Soucing of it-services to offsoure destinations. 3A Strategy.

Longacre, R. E. (1976). *An anatomy of speech notions*. Lisse: de Ridder.

Marshall, C. (1999). *Enterprise Modeling with UML: Designing Successful Software through Business Analysis*. Addison-Wesley.

Rechtin, E. and Maier, M. (1997). *The Art of Systems Architecturing*. CRC Press.

SAP (2000). mysap.com collaborative business scenarios,whitepaper. In *Walldorf*.

Steen, M., Lankhorst, M., and van de Wetering, R. (2002). Modelling networked enterprises. In *the sixth international Enterprise distributed Object Computing Conference (EDOC'02)*.

van der Aalst, W., ter Hofstede, A., B., K., and Barros, A. (2003). Workflow patterns. *Distributed and Parallel Databases*, (3):5-51.

Van Dyke Parunak, H. (1996). Visualizing agent conversations: Using enhanced dooley graphs for agent design and analysis. In *The Second International Conference on Multi-Agent System (ICMAS'96)*.

Verdicchio, M. and Colombetti, M. (2002). Commitments for agent-based supply chain management. *ACM SIGecom Exchanges, 3(1)*.

Webber, D. (2004). The benefits of ebxml for e-business. *In International Conference on XML (XML'04)*.

Wolstenholme, E. F. (1990). *System enquiry: a system dynamics approach*. John Wiley & Sons, Inc., New York, NY, USA.

Xu, L. (2004). *Monitoring Multi-party Contracts for E-business*. PhD thesis, Tilburg University.

Xu, L. and Jeusfeld, M. A. (2003). Pro-active monitoring of electronic contracts. In *The 15th Conference On Advanced Information Systems Engineering in Lecture Notes of Computer Science*, volume 2681, pages 584–600. Springer-Verlag.

Xu, L. and Jeusfeld, M. A. (2004). Detecting violators of multi-party contracts. In *The Conference on CoopIS/DOA/ODBASE in Lecture Notes of Computer Science*, volume 3290, pages 526–543. Springer-Verlag.

Xu, L., Jeusfeld, M. A., and Grefen, P. (2005). Detection tests for identifying violators of multiparty contracts. *ACM SIGecom Exchanges, 5(3)*.

Dr. Lai Xu *is a Senior Research Scientist at CSIRO ICT Centre, Australia. Previously, She worked as a post-doctoral researcher at Organisation and Information Group of the Institute of Information and Computing Sciences of the Utrecht University (2004-2007), the Netherlands and Artificial Intelligence group of the Department of Computer Science of the Free University Amsterdam (2003-2004), the Netherlands respectively. She received her Ph.D. in Computerized Information Systems from Tilburg University, the Netherlands in 2004. Her main research areas are in process integration, multi-party process monitoring, web services and service-oriented applications, development method for product software. Her research has appeared in publications including the European Journal of Information Systems, ACM SIGecom Exchanges, numerous IS conference's proceedings and book chapters.*