A Multi-party Contract Model

Lai Xu
Department of Computer Science, Free University
xu@cs.vu.nl

Contracts between multiple business parties play an increasingly important role in the global economy where activities along the value chain are executed by independent, yet co-operating, companies. Information technology to enact a value chain is now being deployed in the form of ERP systems, workflow systems, web services and e-marketplaces. However, there is little known on how to formally model a multi-party contract. In this paper, we investigate how to model a multi-party contract in a manner convenient for detecting the parties responsible for contract violations.

Categories and Subject Descriptors: H.4.0 [**Information Systems Applications**]: General

Additional Key Words and Phrases: E-contract, Contract violation, Detecting contract violation

1. INTRODUCTION

Contracts between multiple business parties play an increasingly important role in the global economy where activities along the value chain are executed by independent, yet co-operating, companies. Information technology to enact a value chain is now being deployed in the form of ERP systems, workflow systems, web services and e-marketplaces.

A bilateral contract is not able to describe multilateral contractual relations. Most research [Haugen 2002], [Dubray 2002] on multi-party contracts tries to break down a multi-party contract into a number of bilateral contracts. A main reason behind is that existing e-commerce environments only support bilateral executions. In some simple cases, the solution to supporting multi-party contract execution in current eCommerce environments, is to assume the whole business process goes correctly according a number of bilateral contracts. However, in complicated multi-party relationships, this conversion results in the loss or hiding of information about the complex multi-party relationships. Consequently this solution won't work for these complex multi-party contracts.

In a bilateral contract execution process it is easy to find the responsible party for a contract violation. In a multi-party business process, a contract violation can be caused by a set of actions that should have occurred before. It can caused by direct and indirect contractual parties. This raises the problem of finding all

Author's address: Lai Xu, Department of Computer Science, Free University, De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands.

responsible partners for the contract violation. Our research thus focuses on how to model multi-party contracts for convenient detection of the parties responsible for a contract violation.

Various authors have proposed electronic contract models or languages based on different views. Kimbrough and Moore formalize and extend speech act theory as Formal Language for Business Communication (FLBC)[Kimbrough and Moore 1997] [Moore 2000]. *Deontic logic* based contract models [Weigand et al. 1997] [Norman et al. 1998] [Lee 1998] [Tan and Thoen 1999] [Weigand and Xu 2001] describes obligations, permissions, and forbiddances for finishing a business process. CrossFlow [Koetsier et al. 1999] and E-ADOME[Kafeza et al. 2001] use contracts for inter-organizational workflow process integration. Contracts in CrossFlow and E-ADOME describe the agreed workflow interfaces as activities and transitions, based on WfMC's WPDL (Workflow Process Definition Language). Contracts also specify what data objects in the remote workflow are readable or updateable. Grosof discussed a rule-based approach [Grosof and Poon 2003] to representation business contracts, which also deal with exceptions. They are a side effect of business automations, and as for now do not address the multi-party situation and particularly do not looking into detecting contract violations.

In this paper, we present a multi-party contract model and provide how to detect responsible parties for a multi-party contract violation by using our model. In Section 2 a standard multi-party car insurance case [Project 1999] is used to explain our model and to show that in a multi-partner contract it is more important and more difficult to find the responsible parties for a contract violation than in a bilateral contract. Section 3 introduces our multi-party contract model. Contract violations are discussed in Section 4. A detection method of a contract violation is presented in Section 5. The paper ends with conclusions and a short discussion of further work in Section 6.

2. MULTI-PARTY CONTRACT CASE

This case outlines the manner in which a car damage claim is handled by an insurance company (AGFIL). The contract parties work together to provide a service level which facilitates efficient claim settlement. The parties involved are called Euro Assist, Lee Consulting Services, Garages and Assessors. Euro Assist offers a 24-hour emergency call answering service to policyholders. Lee C.S. co-ordinates and manages the operation of the emergency service on a day-to-day level on behalf of AGFIL. Garages are responsible for car repair. Assessors conduct the physical inspections of damaged vehicles and agree upon repair figures with the garages.

The general process of a car insurance case is described as follows: the policyholder phones Euro Assist using a toll-free phone number to notify a new claim. Euro Assist will register the information, suggest an appropriate garage, and notify AGFIL, which will check whether the policy is valid and covers this claim. After AGFIL receives this claim, AGFIL sends the claim details to Lee C.S. AGFIL will send a letter to the policyholder for a completed claim form. Lee C.S. will agree upon repair costs if an assessor is not required for small damages; otherwise, an assessor will be assigned. The assessor will check the damaged vehicle and agree upon repair costs with the garage. After receiving an agreement for repairing the car from Lee C.S., the garage will then commence repairs. After finishing repairs, the garage will issue an invoice to Lee C.S., which will check the invoice against the original estimate. Lee C.S. returns all invoices to AGFIL. After AGFIL also receives the completed claim form from the policyholder, the payment is processed. In the whole process, if the claim is found invalid, all contractual parties will be contacted and the process will be stopped.

There are many potential contract violations in this case, for example, after sending invoices to Lee C.S., the garage does not get money back from AGFIL. It could be caused by

—Lee C.S., because Lee C.S. does not forward the invoices to AGFIL;

—policyholder, because the policyholder did not return the completed claim form to AGFIL;

—AGFIL, because AGFIL forgot to send the claim form to the policyholder or simply because AGFIL did not pay the garage in time.

or any combination from above.

The case study shows a rather complex business process between multiple parties. In particular, we provide an example that the contract violation could be caused by multiple parties.

3. MULTI-PARTY CONTRACT MODEL

A contract is an agreement between two or more parties that is binding to those parties and that is based on mutual commitments [Weigand and Xu 2001]. Our multi-party contract model consists of three core components: actions, commitments and a commitment graph [Xu and Jeusfeld 2003], [Xu 2003], [Xu 2004b]. An *action* describes what each partner should do. A *commitment* in this paper is defined as a guarantee by one party towards another party that some action sequence shall be executed completely, and all involved parties fulfill their side of the transaction. A *commitment graph* is an overview of the commitments between parties, which shows commitment relationships in a contract. These components will be explained in turn in the next sections.

3.1 Actions

Actions will form the edges in commitment graphs. An action is an atom in our contract model. Because a contract party can be involved in different commitments and play the different roles, we specify the roles of a party as $\mathcal{R}$. A set of total roles of a contract is denoted as $\mathbb{R}$.

***Definition* 3.1** A party can act under different roles in different commitments. Let ID be a domain of identifier; **roles of a party** $\mathcal{R}_x$ is defined as

$$\mathcal{R}_x \subseteq ID.$$

Let $\mathcal{P}$ be a set of parties, the **set of all roles** is

$$\mathbb{R} = \bigcup_{\forall x \in \mathcal{P}} \mathcal{R}_x.$$

Roles will form the nodes in commitment graphs

***Definition* 3.2** Let $\mathbb{R}$ be a set of all roles of all parties, ID be the domain ID, and $\mathbb{T}$ be the time. An **action** is specified as

$$action = (name, sender, receiver, deadline),$$

where $name \in ID, sender, receiver \in \mathbb{R}$ and $deadline \in \mathbb{T}$. We require all names of actions to be unique so they can be used as identifiers.

A set of actions $\mathbb{A}$ for a contract can be specified as

$$\mathbb{A} = \bigcup_{\forall x \in \mathcal{P}} \{action\}.$$

For example, Action (A_agreeRepairCar,L,G",3) describes that Lee C.S. agrees the garage to repair the car during the car damage claim received three days. For the car insurance case, all actions are specified in [Xu 2004a]. Although only a single receiver of the action are specified in the case, a list of action receivers can be extend in this model. The next section specifies commitments that are the key part of contracts.

3.2 Commitments

In this paper, a commitment is a guarantee by one party towards another party that some action sequence shall be executed completely provided that some "trigger, involve, and finish" action happens, and all involved parties fulfill their side of the transaction. To finish a commitment, more than one party must finish relevant actions. From this point of view, the concept of our commitment is differ from the definition of a commitment in papers [Verdicchio and Colombetti 2002], [R.Ervin 2002], which a commitment only refers to two parties, a debtor and a creditor [Verdicchio and Colombetti 2002], or a vendor and customer [R.Ervin 2002]. The notion of commitment in this paper is not related to beliefs, desires, or intentions [Cohen and Levesque 1995]. In Cohen and Levesque's research, commitments are related to establishing common beliefs about a certain state of the world. In our multi-party contract model, we do not reason about beliefs of the contractual parties involved, which Daskalopulu did in evidence-based contract performance monitoring research [Daskalopulu et al. 2002]. We also do not assess the of legal status and directives in business process automation [Abrahams 2002].

A multi-party contract includes one or more commitments, a commitment includes some actions which could be performed by multi-parties. Those actions can trigger, involve, and finish the commitment. For example, in the car repair service commitment, the garage first needs to receive the policyholder's car as a trigger of this commitment. The actions included in a commitment thus have different attributes, which we specify as *trigger, involve and finish.* In a contract preparation stage, the actions with "trigger" attribute need to be paid attentions whether some "enforceable" or "compensable" clauses are required for smoothly fulfilling the contract. The actions with "finish" attribute eventually finish the commitment. A commitment is described by a commitment name, sender of the commitment, receiver of the commitment.

***Definition* 3.3** Actions' **attributes** $\mathcal{U}$ can be specified as

$$\mathcal{U} = \{tr, in, fi\}.$$

Let ID be the domain ID, $\mathcal{P}$ be a set of parties, $N = \{1, 2, 3, \ldots\}$, $\mathbb{A}$ be a set of actions. A **commitment** is specified as

$$commitment = (name, sender, receiver, n, \{(a_1, u_1), (a_2, u_2), \ldots, (a_n, u_n) : a_i \in \mathbb{A}, u_i \in \mathcal{U}\}).$$

where name is an identifier, $name \in ID$; sender and receiver are the contract parties, $sender, receiver \in \mathcal{P}$; n denotes the total number of all actions involved, $n \in N$; $a_1, a_2, \ldots, a_n$ denotes all actions involved in the commitment and their attributes $u_1, u_2, \ldots, u_n$. We require all names of commitments to be unique so that they can be used as identifiers.

A set of commitments $\mathbb{M}$ can be specified as

$$\mathbb{M} = \bigcup_{\forall x \in \mathcal{P}} \{commitment\}.$$

Let $a_i \in \mathbb{A}$ and $m \in \mathbb{M}$, a sequence function $f_{position} : \mathbb{A} \times \mathbb{M} \rightarrow N$,

$$f_{position}(a_i, m) = \begin{cases} i & \text{iff } i \text{ is the sequence number} \\ & \quad \text{of action } a_i \text{ in commitment } m. \\ undef & \text{otherwise.} \end{cases}$$

$f_{position}(a_i, m)$ denotes the position of action a_i in the commitment m.

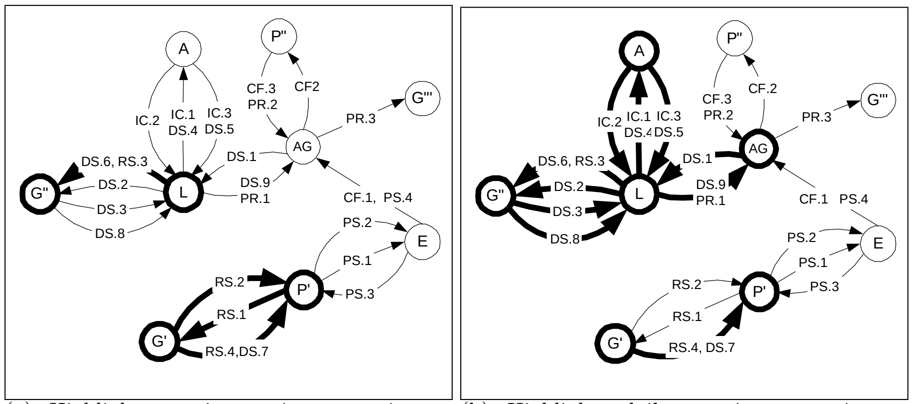
For example, in commitment C_repairService, the garage will offer the repair service to the policyholder. After the policyholder sends his/her car to the garage (action A_sendCar has a trigger attribute), the garage estimates the repair cost (action A_estimateRepairCost has a finish attribute). After the garage receives an agreement from Lee C.S. about the repair cost (action A_agreeRepairCar has a trigger attribute), the garage repairs the car (action A_repairCar has a finish attribute). Commitment C_repairService is specified as

$$(\mathsf{C\_repairService}, \mathsf{G}, \mathsf{P}, \{(\mathsf{A\_sendCar}, tr), (\mathsf{A\_estimateRepairCost}, fi), (\mathsf{A\_agreeRepairCar}, tr), (\mathsf{A\_repairCar}, fi)\})$$

For the car insurance case, all commitments are specified in [Xu 2004a]. The actions and commitments can be regarded as a direct mapping from a paper contract to an e-contract. Information about the actions and commitments can compare with contents between "$< action >$" and "$< /action >$" in TPA (Trading Partner Agreement) [Dan et al. 2001] from ebXML. The difference is that we specify a multi-party contractual process using the commitment concept, and TPA only specifies bilateral business process.

3.3 Commitment Graph

Commitments are an even more important concept, though, to specify multi-party contracts. A commitment graph shows complex relationships among commitments.



(a) Highlight repair service commitment C_repairService

(b) Highlight daily service commitment C_dailyService

Fig. 1. Commitment graphs

Commitment relationships are not only about a condition commitment [Venkatraman and Singh 1999] or a chain [Verdicchio and Colombetti 2002] [R.Ervin 2002] relationship. For example, if a contractee first ships goods to a contractor, the contractor will pay the cost of goods later; the commitment of shipping goods is a condition to activate a commitment of payment.

Figure 1 shows the commitment graph for the car insurance case. Table I provides all abbreviations and labels used in this commitment graph. For all notes of this commitment graph, we use the following abbreviations: P' and P'' for a policyholder, AG for AGFIL, E for Euro Assist, L for Lee C.S., G', G'' and G''' for garage, and A for assessor. Each note represents a role that can be played by a contractual partner.

The abbreviations for the commitments can be found from Table I. Each edge represents an action. Each action has one or more labels, where the first letter represents which commitments this action actually involves, the second number represents the order of a sequence actions within a commitment.

Being able to show how the commitment graph presents the complex commitment relationship, we give an example which shows a relationship between commitments C_dailyService (daily service commitment) and C_repairService (repair service commitment).

According to Section 3.2,C_dailyService and C_repairService are specified as follows:

$$\begin{aligned}(\mathsf{C\_dailyService}, \mathsf{L}, \mathsf{AG}, \{&(\mathsf{A\_forwardClaim}, tr), (\mathsf{A\_contactGarage}, in),\\ &(\mathsf{A\_sendRepairCost}, in), (\mathsf{A\_assignAssessor}, in),\\ &(\mathsf{A\_sendNewRepairCost}, tr), (\boxed{\mathsf{A\_agreeRepairCar}}, fi),\\ &(\boxed{\mathsf{A\_repairCar}}, tr), (\mathsf{A\_sendInvoices}, in),\\ &(\mathsf{A\_forwardInvoices}, fi)\}).\end{aligned}$$

Commitment	Classification of Actions and Commitments			Labels
	Trigger	Involve	Finish	
C_phoneService (PS)	A_phoneClaim			PS.1
		A_receiveInfo		PS.2
			A_assignGarage	PS.3
			A_notifyClaim	PS.4, CF.1
C_repairService (RS)	A_sendCar			RS.1
			A_estimateRepairCost	RS.2
	A_agreeRepairCar			RS.3, DS.6
			A_repairCar	RS.4, DS.7
C_claimForm (CF)	A_notifyClaim			CF.1, PS.4
		A_sendClaimForm		CF.2
			A_returnClaimForm	CF.3, PR.2
C_dailyService (DS)	A_forwardClaim			DS.1
		A_contactGarage		DS.2
		A_sendRepairCost		DS.3
		A_assignAssessor		DS.4, IC.1
	A_sendNewRepairCost			DS.5, IC.3
			A_agreeRepairCar	DS.6, RS.3
	A_repairCar			DS.7, RS.4
		A_sendInvoices		DS.8
			A_forwardInvoices	DS.9, PR.1
C_inspectCar (IC)	A_assignAssessor			IC.1,DS.4
		A_inspectCar		IC.2
			A_sendNewRepairCost	IC.3,DS.5
C_payRepairCost (PR)	A_forwardInvoices			PR.1, DS.9
	A_returnClaimForm			PR.2, CF.3
			A_payRepairCost	PR.3

Table I. Commitments, actions and action abbreviations

$$(\mathsf{C\_repairService}, \mathsf{G}, \mathsf{P}, \{(\mathsf{A\_sendCar}, tr), (\mathsf{A\_estimateRepairCost}, fi), \\ (\boxed{\mathsf{A\_agreeRepairCar}}, tr), (\boxed{\mathsf{A\_repairCar}}, fi)\})$$

In Figure 1 (a) and (b), edge $RS3$ and $DS6$ both denote A_agreeRepairCar according Table I. It means that A_agreeRepairCar is included in C_repairService as the third action ($R3$) and in C_dailyService as the sixth action ($DS6$). Another edge $RS4$ and $DS7$ both indicates A_repairCar. It means that A_agreeRepairCar is also included in C_repairService as the fourth action ($RS4$) and in C_dailyService as the seventh action ($DS7$).

The relationship between C_dailyService and C_repairService is a mixed relationship: after role L agrees with the repair costs in C_dailyService, the role G' can repair the car in C_repairService; after the role G' repairs the car and role G'' sends the invoice, C_dailyService will go on to execute its following actions. Commitments C_repairService and C_dailyService are mutually dependent on each other.

A commitment graph is a directed graph consisting of a set of nodes corresponding to all roles $\mathbb{R}$, a set of edges corresponding to actions and their labels, and commitment orders.

***Definition* 3.4** Let $\mathbb{A}$ be a set of actions, $a \in \mathbb{A}$, $\mathbb{M}$ be a set of commitments, $m \in \mathbb{M}$, and $X = \{1, 2, \ldots\}$, a sequence function $f_{position}(a, m)$, an edge is specified as a relation from $\mathbb{A} \times \mathbb{M} \times X$

$$edge = \bigcup_{\forall m, a \in m} \{(a, m, f_{position}(a, m)) : a \in \mathbb{A}, m \in \mathbb{M}, f_{position}(a, m) \in X\},$$

a set of all edges is

$$\mathbb{E} = \bigcup_{\forall a \in \mathbb{A}} \{edge\}.$$

***Definition* 3.5** Let $\mathbb{M}$ be a set of commitments. A commitment occurrence order is specified as a relation from $\mathbb{M} \times \mathbb{M}$:

$$order\_commitment = \{(m_1 \cdot m_2) : m_1, m_2 \in \mathbb{M}, m_1 \neq m_2\}.$$

If $m_1 \cdot m_2$ is a commitment order, we interpret it as follows: commitment m_2 is only active when commitment m_1 has been finished.

Let $\mathcal{P}$ be a set of parties. A set of commitment order lists all relationships in which a commitment occurs prior to another commitment, and is specified as follows:

$$\mathbb{O} = \bigcup_{\forall x \in \mathcal{P}} \{(m_1 \cdot m_2)\}.$$

For the car insurance case, examples of the commitment orders are presented in [Xu 2004a]. After specification of commitment graph notes, edges, and commitment occurrence orders, the commitment graph can be specified as follows:

***Definition* 3.6** Let $\mathbb{R}$ be a set of nodes, $\mathbb{E}$ be a set of edges, and $\mathbb{O}$ be a set of commitment order list. The commitment graph is defined as follows

$$G = (\mathbb{R}, \mathbb{E}, \mathbb{O}).$$

3.4 Multi-party Contract

Now that all elements of our multi-party contract model have been presented, a formal model is provided as follows:

***Definition* 3.7** Let $\mathbb{A}$ be a set of actions, $\mathbb{M}$ be a set of commitments and G be a commitment graph of a contract. The multi-party contract is specified as

$$Contract = \{\mathbb{A}, \mathbb{M}, G\}.$$

The next section will illustrate how to detect responsible parties after a contract violation.

4. CONTRACT VIOLATIONS

Contract violations refer to break or fail to comply with a term of the contract by contractual parties. The contract violation is an essential problem for contract executions. Especially, the contract violation can caused by more than one contractual parties in a contract performance. For example, in the car insurance case, the policyholder sent the car to the assigned garage. After the prescribed days, the policyholder find that his/her car did not fixed by the garage at all. Obviously the policyholder will directly complain to the garage. Actually, after sending an estimated repairing cost of the car to the Lee C.S., the garage maybe did not receive an agree-repair message from Lee C.S. Well, because the estimated repairing cost was too high, Lee C.S. has to send an assessor to check the car again. However, the assessor did not do his/her job properly. From the contract execution point of view, this contract violation is caused by the assessor, but Lee C.S. and the garage should also take care of the deadline of repairing the car. In this example, the garage did not repair the car in time, which is directly caused by the assessor. Normally the contract violations can be found by any contractual parties or a contract monitor.

5. DETECTING RESPONSIBLE PARTIES OF THE CONTRACT VIOLATION

The most common detection process is to retrieve all actions that should have already occurred. Although it is a solution, this process is rather inefficient. Our approach is that use of commitment graphs optimizes the detection process.

After finding a “missing” action, edge labels and commitment orders of a commitment graph will be used in a detecting process. The process of detecting responsible partners of a contract violation has the following steps. First we find all commitments that can be reached from the “missing” action by using the edge labels. These commitments are direct commitments. Next we find all commitments that should have been completed before the reachable commitments. These commitments are indirect commitments.

For the indirect commitments, the actions which have “fi” attribute, are checked, if they occurred, that indirect commitment is completed. For direct commitments, the occurrence of all actions up to the highest reachable actions need to be checked. If an indirect commitment was not completed, the actions of that commitment are recursively checked. For all actions found missing, the parties responsible for performing those actions are jointly responsible for the contract violation. The detection algorithm is shown in Algorithm 1.

This section explained the detection process that makes it possible to detect the parties responsible for a contract violation. This approach uses the multi-party contract model, particularly the commitment graph, to improve the efficiency of the detection process.

6. CONCLUSIONS

This paper proposes an approach to formalizing multi-party electronic contracts. The execution of a contract is seen as a stream of actions that happen over time. A paper contract is mapped into two parts. The first part is formed by the so-called contract actions. The second part of the contract is the commitments which are essentially guarantees by one partner to another partner that some action sequence will occur. The commitment graph is used to specify the relationships between commitments. One of our main focuses is how to find the party (or parties) responsible for a contract violation. We provide a method using the commitment graph to trace back the commitments after a contract violation and to locate the partners who violated the commitments. This research also provides a foundation for representing and automating contractual deals on web services, so as to help search, select and compose them.

Further research has to be undertaken in the area of pre-calculating the costs of multi-party contract violations from the point of view of one contractual party. Because of the autonomous, reactive and proactive features of agents, they can act on behalf of their owners and use individual strategies to handle conflicts between multiple contract executions. Some agents may use a remedial mechanism which might return the business processes to a normal course of action after a contract violation. How to pre-calculate the cost of the contract violation and trying to reduce the potential costs are very important for a particular contractual party.

Algorithm 1 Detecting Responsible Parties of the Contract Violation

Input: $a_{missing}$, $Contract = \{\mathbb{A}, \mathbb{M}, G\}$,
▷ /*Finding direct commitments and positions of the involved actions*/
$\mathcal{M}_{direct} = \bigcup_{\forall m \in \mathbb{M}} \{edge.m : \exists edge, (edge.a, edge.m), edge.a = a_{missing}\}$,
$(f_{position}, m) = \{(edge.f_{position}(a, m), edge.m) : edge.a = a_{missing}\}$,
▷ /*Finding all indirect commitments*/
$\mathcal{F}_{position} = \bigcup_{\forall m \in \mathcal{M}_{direct}} \{(f_{position}, m)\}$,
$\mathcal{M}_{mid} = \mathcal{M}_{direct}$, $\mathcal{M}_{indirect} = \emptyset$,
repeat
  $\mathcal{M}_{mid} = \bigcup_{\forall m \in \mathcal{M}_{mid}} \{m' : m' \cdot m \in \mathbb{O}\}$
  $\mathcal{M}_{indirect} = \mathcal{M}_{indirect} \cup \mathcal{M}_{mid}$
until $\mathcal{M}_{mid} \neq \emptyset$
▷ /*Checking all indirect commitments*/
for $\forall m \in \mathcal{M}_{indirect}$ **do**
  for $\{\forall a \in \mathbb{A} : m.(a, u) \wedge u = \text{“}fi\text{”}\}$ **do**
    if $a(m, m.n)$ already occurs **then**
      return(none)
    else
      for $x = m.n$ to 1 **do**
        if $a(m, x)$ not yet occurs and $x > 1$ **then**
          $x - 1$
      end if
        if $a(m, x)$ not yet occurs and $x = 1$ **then**
          return(none)
      end if
        if $a(m, x)$ already occurs **then**
          return($sender.a(m, x + 1)$)
      end if
    end for
   end if
  end for
end for
▷ /*Checking all direct commitments*/
for $\forall m \in \mathcal{M}_{direct}$ **do**
  for $y = m.f_{position} - 1$ to 1 **do**
    if $a(m, y)$ not yet occurs and $y > 1$ **then**
      $y - 1$
   end if
    if $a(m, y)$ not yet occurs and $y = 1$ **then**
      return(none)
   end if
    if $a(m, y)$ already occurs **then**
      return($sender.a(m, y + 1)$)
   end if
  end for
end for
if return(sender) $= \emptyset$ **then**
  return($sender.a_{missing}$)
end if

REFERENCES

Abrahams, A. 2002. An asynchronous rule-based approach for business process automation using obligations. In *the 3rd ACM SIGPLAN Workshop on Rule-Based Programming*. Pittsburgh, USA.

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

Dan, A., Nguyen, T. N. ., Dias, D. M., Parr, F. N., Kearney, R., Sachs, M. W., Lau, T. C., and Shaikh, H. H. 2001. Business-to-business intergration with tpaml and a business-to-business protocol framework. *IBM Systems Journal 40,* 1.

Daskalopulu, A., T, T. D., and Maibaum, T. 2002. Evidence-based electronic contract performance monitoring. *INFORMS Journal of Group Decision and Negotiation Special Issue: formal Modeling of Electronic Commerce*.

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

Grosof, B. and Poon, T. 2003. Sweetdeal: representing agent contracts with exceptions using xml rules, ontologies, and process descriptions.

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

Kafeza, E., Chiu, D., and Kafeza, I. 2001. View-based contracts in an e-service cross-organizational workflow environment. In *the second International Workshop on Technologies for E-Service*.

Kimbrough, S. and Moore, S. 1997. On automated message processing in electronic commerce and work support systems: Speech act theory and expressive felicity. *ACM Transactions on Information Systems 15,* 4, 321–367. ACM Press. New York, NY.

Koetsier, M., Grefen, P., and Vonk. 1999. Contract model. Tech. Rep. Deliverable D4b, Cross-Organisational/Workflow, Crossflow ESPRITE/28635.

Lee, R. 1998. Towards open electronic contracting. *Electronic Markets 8,* 3 (Mar.).

Moore, S. 2000. Kqml and flbc: Contrasting agent communication languages. *International Journal of Electronic Commerce 5,* 1.

Norman, T. J., Sierra, C., and Jennings, N. R. 1998. Rights and commitments in multi-agent agreements. In *Proceedings of the 3rd International Conference on Multi-Agent Systems (ICMAS-98)*. Paris, France.

Project, C. 1999. Insurance requirements. Tech. Rep. CrossFlow deliverable: D1.b, CrossFlow consortium.

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

Tan, Y.-H. and Thoen, W. 1999. A logical model of directed obligations and permissions to support electronic contracting in electronic commerce. *International Journal of Electronic Commerce 3,* 2, 87–104.

Venkatraman, M. and Singh, M. P. 1999. Verifying compliance with commitment protocols: Enabling open web-based multiagent systems. *Autonomous Agents and Multi-Agent Systems 2,* 3.

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

Weigand, H., Dignum, F., and Verharen, E. 1997. Dynamic business models as a basis for interoperable transaction design. *Information Systems*.

Weigand, H. and Xu, L. 2001. Contracts in e-commerce. In *9th IFIP 2.6 Working Conference on Database Semantic Issues in E-Commerce Systems (DS-9)*.

Xu, L. 2003. Monitorable electronic contract. In *The 2003 IEEE Conference on E-Commerce (CEC'03)*. IEEE Computer Society Press.

Xu, L. 2004a. Appendix: all actions and commitments of a car insurance case. `http://www.cs.vu.nl/~xu/appendix.pdf`.

Xu, L. 2004b. Monitoring multi-party contracts for e-business. Ph.D. 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*. Vol. 2681. Springer-Verlag, 584–600.