

# Pro-active Monitoring of Electronic Contracts

Lai Xu <sup>a</sup>      Manfred A. Jeusfeld <sup>b</sup>

<sup>a</sup> Free University, Department of Computer Science,  
De Boelelaan 1081a, 1081 HV Amsterdam.  
{xu@few.vu.nl}

<sup>b</sup> Tilburg University, CRISM/Infolab  
5000 LE, Tilburg.  
{Manfred.Jeusfeld@uvt.nl}

*This paper is published at The 15th Conference on Advanced Information Systems Engineering (CAiSE), June 16-18, 2003. Also in Lecture Notes of Computer Science, Springer-Verlag.*

## 1 Introduction

A contract between multiple business partners contains some statements about their business relationship, in particular on their physical and informational actions. One purpose of such a contract is to distinguish expected and acceptable behaviour from forbidden behaviour.

The introduction of workflow systems and enterprise resource planning systems increases the automation of business contract execution. To the same degree, the demand for automated monitoring increases just because more information about the contract execution has to be processed by the business partners.

The business partners are interested in answers to the following questions:

- 1 Given the current state of contract execution, which actions are expected from a partner in the future.
- 2 Is a contract violation imminent, i.e. likely to happen within short time? Which partner has to be remind to fulfil her obligation?
- 3 Which partner is responsible for a contract violation?

We address the problem by regarding it as a *formalisation* problem: Given a paper contract, formalise it into suitable representations such that the three main questions can be answered. Essentially, we map informal requirements (the paper contract) into formal specifications that are subject to automated processing very much like system requirements are mapped into implementations.

## 2 Monitorable Contract Model

This paper proposes an approach to formalise electronic contracts into a set of representations that enable automatic monitoring. The formal contract model

consists of two core components: a monitorable element and a monitoring mechanism.

The **monitorable element** include

- trade process
  - actions: all activities mentioned in a contract.
  - commitment: a sequence of activities promised by some partner to become true in the future.
- logic relationships
  - contract constraints: a statement about the well-formedness of a contract execution by using temporal logic.
  - guards of constraints: the right order of actions checks what obligations remain to be realized after the occurrence of the guarded action.

The **monitoring mechanism** consists of:

- monitoring module
  - commitment graph: an overview of commitments between contractual parties.
  - pro-active detect algorithm
  - maintaining guard algorithm
- reactive modules: they respectively support anticipation and avoidance before anomalous action occur, detection and compensation after anomalous actions occur

### 3 Conclusions

Our contract model is more suitable for electronic contract executions. Imminent contract violations may be forecasted by checking the state of the so-called guard expressions ahead of the formal deadline of an expected action. This feature allows pro-active use of formal contract representations in order to avoid real violations. Rather than passing a violation case to a legal law suit, the failing partner can be forced to commit to a compensation that creates value for all partners. Without automatic monitoring, the detection of compensation opportunities is simply too costly to justify complete monitoring.

Further research has to be undertaken in the area of quality safeguards in electronic contracts. Lack of trust between partners may be dealt with by introducing a trusted third party which sub-divides actions into parts that are then irrevocable or provide monitoring services. An electronic contract can be analysed prior to its execution in order to avoid incomplete commitment structures. Specifically, one may verify whether any violation of a contract constraint can be traced back to a commitment, i.e. a partner who is responsible for the violation.