ABSTRACT

Electronic contracts mirror the paper versions exchanged between businesses today, and offer the possibility of dynamic, automatic creation and enforcement of restrictions and compulsions on service behaviour that are designed to ensure business objectives are met. Where there are many contracts within a particular application, it can be difficult to determine whether the system can reliably fulfil them all, yet computer-parsable electronic contracts may allow such verification to be automated. In this chapter, the authors describe a conceptual framework and architecture specification in which normative business contracts can be electronically represented, verified, established, renewed, and so on. In particular, they aim to allow systems containing multiple contracts to be checked for conflicts and violations of business objectives. They illustrate the framework and architecture with an aerospace aftermarket example.

DOI: 10.4018/978-1-61520-686-5.ch031
INTRODUCTION

It has often been argued that independent entities, such as business services, interacting in a common system, society or environment need to be suitably constrained in order to avoid and solve conflicts, make agreements, reduce complexity, and in general to achieve a desirable social order (Conte & Castelfranchi, 1993; Conte, Falcone & Sartor, 1999). For many, this role is fulfilled by norms, which represent what ought to be done by a set of services (when performing functions on behalf of their owning business). Views of norms differ, and include fixed laws that must never be violated as well as more flexible social guides that merely seek to bias behaviour in different ways. Yet the obligations, prohibitions and permissions that may affect service behaviour in a normative system can also be documented and communicated between services in the form of contracts.

Electronic contracts, mirroring the paper versions exchanged between businesses today, offer the possibility of dynamic, automatic creation and enforcement of such restrictions and compulsions on service behaviour. However, where there are many contracts within a particular application, it can be difficult to determine whether the system can reliably fulfil them all; computer-parsable electronic contracts may allow such verification to be automated.

In a peer-to-peer system, organisations, and the services performing functions on their behalf, act as independent peers, with no overall authority, and contracts are necessary to add predictability to behaviour between them. Where there is multiple, independently owned alternatives for a resource or service, contract technology is of particular use. By providing and monitoring contract compliance, applications can make better decisions on which resources or services to take advantage of in the future, a particular problem on Grid systems with a range of reliability issues.

There are two pre-requisites to realistically applying an electronic contracting approach in real-world domains. First, to exploit electronic contracts, a well-defined conceptual framework for contract-based systems, to which the application entities can be mapped, is needed. Second, to support the management of contracts through all stages of the contract life-cycle, we need to specify the functionality required of a contract management architecture that would underlie any such system, leading to ready-made implementations for particular deployments of that architecture. The CONTRACT project (CONTRACT, 2008) aims to do just this. Funded by the European Commission as part of its 6th Framework Program, the project seeks to develop frameworks, components and tools that “make it possible to model, build, verify and monitor distributed electronic business systems on the basis of dynamically generated, cross-organisational contracts which underpin formal descriptions of the expected behaviours of individual services and the system as a whole.” In this context, this chapter documents the CONTRACT project’s work on both of the pre-requisites identified above. More specifically, the technical contributions described in this chapter are:

- The specification of a model for describing contract-based systems
- The specification of an architecture for managing such systems
- The mapping of an aerospace application to those models

Our approach is distinct in several respects. First, its development is explicitly driven by a range of use cases (Jakob et al., 2008) provided by a diverse set of small and large businesses. One consequence of this diversity is that our approach must account for different practices and possibilities in each stage of the lifecycle of a contract-based system. It is therefore defined in terms of abstract process types, to be instantiated in different ways for different circumstances. We provide a non-exhaustive set
Related Content

Hilbert Index-based Outlier Detection Algorithm in Metric Space
[www.igi-global.com/article/hilbert-index-based-outlier-detection-algorithm-in-metric-space/172504?camid=4v1a](www.igi-global.com/article/hilbert-index-based-outlier-detection-algorithm-in-metric-space/172504?camid=4v1a)

The Red Storm Architecture and Early Experiences with Multi-Core Processors
[www.igi-global.com/chapter/red-storm-architecture-early-experiences/64449?camid=4v1a](www.igi-global.com/chapter/red-storm-architecture-early-experiences/64449?camid=4v1a)

Enhancing the Grid with Multi-agent and Semantic Capabilities
[www.igi-global.com/chapter/enhancing-grid-multi-agent-semantic/58752?camid=4v1a](www.igi-global.com/chapter/enhancing-grid-multi-agent-semantic/58752?camid=4v1a)

Using Free Software for Elastic Web Hosting on a Private Cloud
[www.igi-global.com/chapter/using-free-software-elastic-web/64512?camid=4v1a](www.igi-global.com/chapter/using-free-software-elastic-web/64512?camid=4v1a)