

GridMIST

Middleware Services and Tools for managing Resource Sharing in Virtual Organizations

Organizations are increasingly using the Internet and Grid to offer their own services and to utilise the services of others. Presence of a wide variety of services and resources creates new opportunities for providing value added, inter-organizational services by composing multiple existing services into new Composite Services (CSs). This naturally leads to resource sharing across organizational boundaries. An inter-organizational business relationship is commonly referred to as a virtual organization (VO). Whether in the context of large-scale scientific experiments or eCommerce, eGovernment or any other collaborative effort, organizations need cost-effective ways of finding, purchasing and managing services performed by other organizations. It is therefore necessary for organizations to be able to set-up and manage business links with other organizations in a rapid, dynamic and flexible manner. However Virtual Organisations blur the distinction between outsiders and insiders, yet organizations forming a VO wish to preserve their individual autonomy and privacy. A central problem in VO management therefore concerns how organizations can regulate access to their resources by other organizations in a way that ensures their individual policies for information sharing are honoured. Regulating access to resources by other organizations is difficult because the organizations might not trust each other. Organizations will therefore require that their interactions with other organizations be strictly controlled and policed.

In the paper-based world, businesses have been conducted using contracts. The concept and the use of contracts are not new to today's society. Legal contracts can be

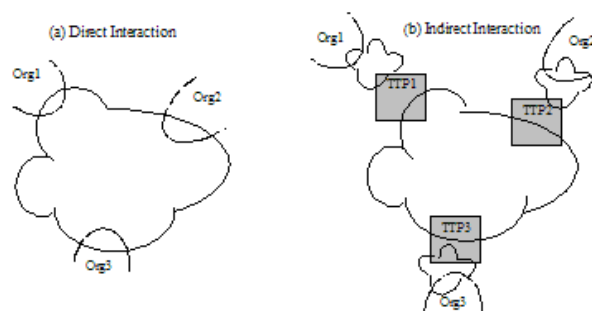


Figure 1. Inter-organisation Interactions

traced back to ancient times. To form and manage VOs, we need to emulate electronic equivalents of contract-based business management practices; this means that relationships between organizations for information access and sharing will need to be regulated by *electronic contracts*, and then enforced and monitored.

In the Gridmist project, we have begun by investigating the types of middleware services required. We will investigate whether it is possible to determine a small set of generic services that can be used to support arbitrarily complex interactions between organizations. Figure below shows two basic interaction styles, where a cloud represents middleware services shared between the organizations: (a) organizations trust each other sufficiently to interact directly with each other; and (b) no direct trust exists between the organizations, so interactions take place through trusted third parties (TTPs) acting as intermediaries. A given contractual relationship could be implemented by either interaction style. In a dynamic setting, it may be that interactions initially take place through TTPs and once sufficient trust has been established, organizations agree to interact directly.

We have investigated how a contract can be represented electronically (an executable version). Each entry in a contract is called a *term* or a *clause*. The clauses of a

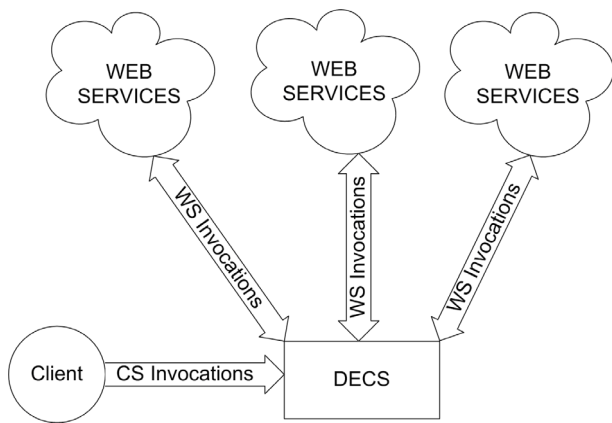


Figure 2. Centralised Coordination of a Composite Service

contract stipulate how the signing parties are expected to behave. In other words, they list the rights and obligations of each signing party.

A *right* is an action that a signing entity can perform if it wishes to. For example, a contract might stipulate that Alice, as a manager of enterprise E1, has the right to send an offer to Bob, the manager of enterprise E2.

Because this is a right, it is up to Alice to send or not to send the offer to Bob; Bob need not be disappointed if he does not receive the offer. Similarly, an *obligation* can be defined as a duty that an entity is expected to perform.

A failure to perform such a duty means a breach of the contract. For example, a contract might stipulate that upon receiving an offer to sell from Alice, Bob has the obligation to reply to her with an *OfferAccepted* or *OfferRejected* message.

Of course, there are no easy ways of automatically transforming a contract written in a natural language into an executable version, but several systematic, semi-automatic approaches have been proposed and are under investigation. We have de-

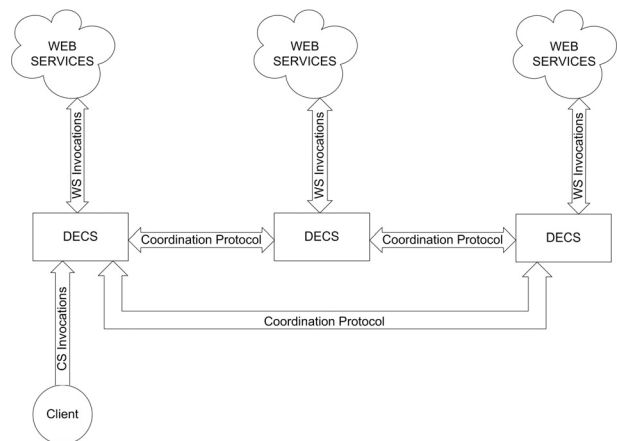


Figure 3. Decentralised Coordination of a Composite Service

veloped a systematic way of representing contracts as finite state machines (FSMs) and described how rights and obligations can be monitored and enforced.

Secondly, we have developed a composite service (CS) coordination system that can be used for enactment of business processes of a VO represented as workflows. The system provides the option of both centralised coordination as shown in figure 2 and decentralised coordination, as shown in figure 3. Should the coordination of the service be spread across multiple servers as in figure 3, a higher level of fault tolerance is provided. In such cases, each server makes the invocations of the constituent Web/Grid services for its part of the CS, and communicates via a coordination protocol with its peers to orchestrate the overall execution. Should a coordinating server fail or leave, it is possible to move the CSs that server was coordinating to another server. The cost of doing so is proportionate to the number of CSs being coordinated and the complexity of those CSs.

Further work is required in integrating contract management. This work will be undertaken with the help of future e-science projects.