A Resource Negotiation Infrastructure for Self-Managing Applications

D.G.A. Mobach, B.J. Overeinder, and F.M.T. Brazier
Vrije Universiteit Amsterdam, de Boelelaan 1081a,
1081 HV Amsterdam, The Netherlands
{mobach,bjo,frances}@cs.vu.nl

1 Introduction

Resource access and regulation are necessary for all applications and run-time support systems to function, including distributed Internet applications. These applications often span multiple system domains. An important specific class of these applications is self-managing. This class of applications plan and coordinate their own resource access. The systems from which resources are acquired are, in turn, heterogeneous autonomously managed systems. The coordination between self-managing applications and such autonomous systems mandates an infrastructure with a uniform interface that respects the autonomy of both self-managing entities to support resource negotiation.

An infrastructure for resource negotiation respects the autonomy requirement: both application and system can set and negotiate the terms of an agreement. A two level negotiation model is proposed: the first level is that of the resources themselves, the second that of virtual domains in which aggregated sets of resources are offered to applications. The negotiation protocol and language used to specify resource requirements are both based on the WS-Agreement specification with application dependent domain ontologies for specific resources.

2 Negotiation Model

Negotiation infrastructure for resource access in an Internet environment, demands solutions that deal with the dynamics and heterogeneity of self-managing hosts and applications. Hosts are autonomous entities that provide resources to applications with specific usage and access policies. Each host is represented by a *host manager* which maintains information regarding resource availability and usage on the host, and policy information regarding these resources. Hosts are aggregated into virtual domains, represented by *domain coordinators*.

Applications enter into negotiations for resource access with domain coordinators using a pre-defined interaction interface. Each domain coordinator then negotiates with its host managers to provide the resources requested. These resources may well be acquired from different hosts at the same time. A domain coordinator presents a proposal with a possible aggregation of resources from different hosts to an application.

The result of a two-tiered negotiation is a time-limited contract between a domain coordinator and an application specifying which resources may be accessed during the duration of the contract, and under which conditions the resources may be used. Figure 1 shows the main elements of the negotiation model: each host runs a host manager process (HM), and one of the manager processes is the domain coordinator (DC).

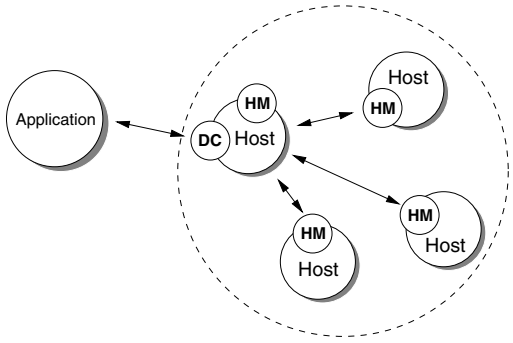

Figure 1. Negotiation model.

The negotiation protocol and language used in the negotiation model are based upon the *WS-Agreement specification* [1]. This specification defines an XML-based language for specifying agreements between resource providers and consumers, and a protocol for establishing these agreements (these agreements are time-limited contracts in our model). Agreement *terms* are used to describe the (levels of) service negotiated. Two types of terms are distinguished for agreement specifications: (i) *Service Description Terms*, describing the services to be delivered under the agreement, and (ii) *Guarantee Terms*, expressing the assurances on service quality (e.g., minimum bounds) for the services described in the service description terms. The specification of domain-

Proceedings of the Second International Conference on Autonomic Computing (ICAC'05)
0-7695-2276-9/05 $ 20.00 **IEEE**

specific term languages is explicitly left open. The WS-Agreement interaction model defines that consumers can request agreements from resource providers by issuing an agreement *offer* based on available agreements *templates*, which, if accepted, result in new *agreements*.

Host Manager
In our model a host manager manages its own resources and resource negotiation with a domain coordinator. This includes negotiation, creation, and enforcement of agreements. Negotiation is based on templates. Templates specify which resources can be made available at any given point in time (e.g, allowing for load balancing). The offer a host makes on request of a domain coordinator are based on these templates. After the negotiation phase, the host manager monitors and controls the resource usage to ensure that agreements are honoured. The agreements in the model are time-limited contracts (agreements that expire after some predetermined time).

Domain coordinator
A domain coordinator is responsible for resource access negotiation with applications and its enforcement. Agreement requests from applications are received by a domain coordinator. A domain coordinator, in turn, requests and receives information on availability of resources from its hosts, and combines this information to construct application directed composed templates.

Applications use these templates to construct specific resource requests. Once a template-based request is received from an application, a domain coordinator requests offers from its hosts, decides which offers are optimal based on host and domain policies, and returns a proposed agreement to the application if possible. If a proposed agreement is accepted, the domain coordinator is responsible for the effectuation and enforcement by the hosts providing the resources.

3 AgentScape Implementation

The negotiation model described above is the basis for the negotiation architecture for autonomous mobile software agents in the AgentScape framework [3]. Agents wanting to move to another location first decide to which location to migrate based on the result of resource access negotiation with a number of domain coordinators. Domain coordinators with the best proposals are chosen: agents migrate to these domains. Within AgentScape, negotiable resources are specified in the XML Schema language and include: *CPU time*; *Communication bandwidth*; *Memory*; *Web service access*; *Disk space*. Additional resources will be defined in the future, as the functionality and services offered by AgentScape are extended.

4 Discussion

The negotiation infrastructure presented hides the complexity of managing access and usage of heterogeneous and distributed resources, by providing a uniform negotiation infrastructure to aggregate resource access within a virtual domain. For additional implementation details regarding the application of the negotiation architecture in the AgentScape middleware, see [2].

The focus of our current and future work includes extension of the architecture and model with application level components, facilitating the integration and implementation of resource negotiation interactions into applications. For example, in the AgentScape middleware, a WS-Agreement based Agent Communication Language would allow agents to interact with the resource negotiation infrastructure in a way that facilitates knowledge sharing, knowledge modeling, and expression of performatives that agents are permitted to use.

Furthermore, the addition of more expressive and flexible negotiation protocols are being devised to allow both applications and resources more fine-grained control of the negotiation process. Other future work studies negotiation strategies in various settings including the dynamics of agreements, i.e., when agreements cannot be met, or when they are deliberately violated or cancelled.

References

[1] A. Andrieux, K. Czajkowski, A. Dan, K. Keahey, H. Ludwig, J. Pruyne, J. Rofrano, S. Tuecke, and M. Xu. Web Services Agreement Specification WS-Agreement (draft), 2004.

[2] D. G. A. Mobach, B. J. Overeinder, O. Marin, and F. M. T. Brazier. Lease-based Decentralized Resource Management in Open Multi-Agent Systems. In *Proceedings of the 18th International FLAIRS Conference*, Clearwater Beach, FL, USA, May 2005. (to appear).

[3] B. J. Overeinder and F. M. T. Brazier. Scalable middleware environment for agent-based Internet applications. In *Proceedings of the Workshop on State-of-the-Art in Scientific Computing (PARA'04)*, Copenhagen, Denmark, June 2004.