Seamless Cross-Domain Connectivity for Enabling Domain Autonomy in a Federated SOA

Ignacio Silva-Lepe, Isabelle Rouvellou, Rahul Akolkar, Arun Iyengar
IBM T.J. Watson Research Center, 19 Skyline Drive, Hawthorne NY 10532, USA
{isilval,rouvellou,akolkar,aruni}@us.ibm.com

1 Introduction

To tackle SOA projects that span across various boundaries, enterprises are adopting a federated SOA approach in order to manage reuse across service domains [1, 3]. A service domain, as used in this paper, is a collection of services that reflects the underlying organizational, geographical or governance structure of an enterprise. Managing service reuse involves sharing a subset of the services that are provided within a domain and fulfilling references to services required by applications or other services in a domain. While this is a major goal, it is also important for a federated enterprise to enable the autonomy of its multiple service domains. For the purposes of this paper, we consider as domain autonomy the ability to (1) decide what domain-provided services to share and how, without the need for a central federation authority, and (2) reuse required services without the need for explicit domain membership knowledge or the existence of a federation architect to fulfill required references. This paper proposes an approach for cross-domain connectivity that enables domain autonomy and that preserves across domains properties such as location transparency, dynamic selection, and asynchronous connectivity, that are taken for granted by services within a domain. We introduce a cross-domain connectivity capability and show how this capability allows for services to be shared and reused without the need for a federation architect, and how it preserves intra-domain properties, thus enabling domain autonomy in a federation.

2 Cross-domain Connectivity

Consider a situation where a portfolio manager (PM) application in an e-business domain uses a stock quote service (SQS) that is located in an independent an autonomous financial services domain. This situation is illustrated in Fig. 1. Each service domain uses a service registry to record descriptions of service providers, and of consumer requirements. To handle the infrastructure differences across domains, a cross-domain proxy (SQSP in the figure) can be manually added to the e-business domain (and its registry); this proxy can be found and invoked by the portfolio manager. However, performing connectivity maintenance by manually adding and removing a cross-domain proxy when its target service is added to or moved across domains means that dynamic selection or location transparency of the service is lost across domains. We introduce cross-domain connectivity automation that addresses this problem by relying on three elements:

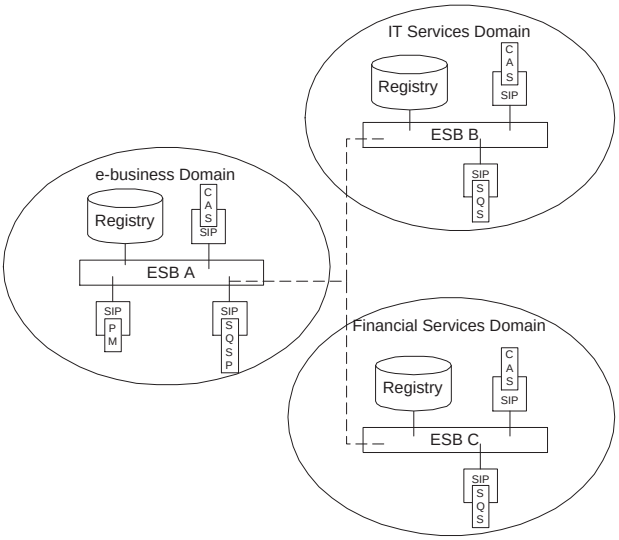

Figure 1. Connectivity Across Multiple Autonomous Domains

(1) The ability for a service consumer to express and record its interest in a service satisfying a particular description; this is in addition to the ability of a service provider to advertise and record its availability as satisfying a particular description. A consumer expresses interest using a required service reference (RSR), which at any given time may be fulfilled or unfulfilled. A RSR contains the description of the required service and, if it is fulfilled, one or more cross-

domain proxies to the target services.

(2) The bilateral dissemination of interest and availability information that are matched at either the consumer end as incoming availability with interest, or at the provider end as incoming interest with availability. Given the ability of service consumers and providers to express interest and advertise availability, a service domain takes these notifications and exchanges them with other domains; when an exchange results in a match, connectivity can be established by the addition of a suitable proxy. This procedure is embodied in a Connectivity Automation Service (CAS) that is part of a service domain's infrastructure. Fig. 1 shows our example domains also including their respective instances of the CAS. The CAS behaves as a representative of either a consumer expressing interest, keeping track of required service references, or of a provider advertising availability, keeping track of provided service entries. In our example situation, the portfolio manager domain's CAS takes the interest notification and sends out an interest message to available domains. Independently, the financial services domain's CAS takes the stock quote service availability notification and sends out an availability message to interested domains. When the financial services domain's CAS receives an interest message, and if the service description in the message matches the description of an available service, then connectivity can be established.

(3) The cooperative handling of proxy maintenance between a provider and a consumer that automates the connectivity across domains. One goal of cross-domain connectivity automation is to eliminate the need for manual intervention in the addition and removal of proxies. This is accomplished as a cooperative effort between a provider domain and a consumer domain when interest and availability find each other, or when a service becomes unavailable at a provider domain. For example, when the financial services domain's CAS determines that connectivity can be established it sends out a connectivity message. When this connectivity message arrives at the portfolio manager domain's CAS, and assuming that the service description in the message matches the stock quote service, a transition to the fulfilled state occurs.

3 Enabling Domain Autonomy

Services within a domain can enjoy properties such as dynamic selection, location transparency and asynchronous connectivity. We outline how connectivity automation enables these properties to hold across domains.

Dynamic selection allows the determination of the ultimate target of a service interaction to be delayed until the outgoing message is sent. Across domains, interest and availability dissemination and matching, and proxy handling automation, allow a fulfilled required service reference to maintain the latest proxy to the target service.

Location transparency refers to the ability of a provided service to change its endpoint address without impacting any of its consumers. Across domains, a provided service that changes location takes advantage of availability dissemination and matching to have its old and new proxies automatically removed and added from and to the domains of its consumers.

Asynchronous connectivity allows a sender to not be constrained by the availability of the provider, assuming the domain infrastructure has a way to hold on to an outgoing message. Across domains, availability of the provider is given by an availability message that matches the interest expressed by a sender, a straightforward new condition for the holding of an outgoing message by the domain infrastructure.

4 Implementation

A prototype of the connectivity automation techniques described in this paper is being implemented and integrated with the SOAlive hosted environment [2], which also provides a minimal ESB function.

5 Conclusions

We have introduced cross-domain connectivity automation as a mechanism to enable domain autonomy in a federation. We have also outlined how intra-domain properties such as dynamic selection, location transparency and asynchronous connectivity are made available across domains.

References

[1] G. Flurry and M.-T. Schmidt. Exploring the Enterprise Service Bus: Part 4: Federated connectivity in the enterprise. `http://www.ibm.com/developerworks/websphere/library/techarticles/0901_flurry/0901_flurry.html`, January 2009.

[2] I. Silva-Lepe, R. Subramanian, I. Rouvellou, T. Mikalsen, J. Diament, and A. Iyengar. SOAlive Service Catalog: A Simplified Approach to Describing, Discovering and Composing Situational Enterprise Services. In *Proceedings of the 6th International Conference on Service-Oriented Computing (ICSOC 2008)*, Sydney, Australia, December 2008.

[3] M. Yunus. Federated SOA: A Pre-Requisite for Enterprise Cloud Computing. `http://cloudcomputing.sys-con.com/node/1233457`, January 2010.