JTangSynergy 3.0: A Framework and Software Tool for Integrating Cross-Organizational Applications

Hanwei Chen, Jianwei Yin, Haifeng Liu
College of Computer Science and Technology
Zhejiang University
Hangzhou, China
chw@cs.zju.edu.cn, zjuyjw@cs.zju.edu.cn, haifeng@gmail.com

Calton Pu
College of Computing
Georgia Institute of Technology
Atlanta, USA
calton@cc.gatech.edu

Abstract—**Enterprise Service Bus is one of the most promising infrastructures for simplifying Enterprise Application Integration (EAI). However, in order to exploit its full potential for integrating large scale cross-organizational applications, a flexible and reliable distributed environment management infrastructure and an appropriate security control mechanism are required. This paper presents a domain-based cross-organizational ESB (CO-ESB) platform named JTangSynergy 3.0 to facilitate the integration of large scale cross-organizational applications. This platform is being used in the design and implementation of e-Government system integration in Hangzhou, China.**

Keywords-Enterprise service bus; Cross-organizational application integration; Distributed environment management

I. INTRODUCTION

Due to the loosely coupled and decentralized interactions and standardization, Enterprise Service Bus (ESB) is becoming a leading software tool for simplifying the development of Enterprise Application Integration (EAI) applications. However, large scale cross-organizational EAI presents new challenges due to the security and autonomy boundaries between organizations. We illustrate the new challenges in two areas in which ESB does well within an organization but have difficulties when crossing traditional organizational boundaries: large scale distributed application management and integration of heterogeneous data services.

Cross-Organizational large scale distributed application management. Each organization has its own security policies, administrative processes, and information exchange controls. At a base level, information exchange between organizations must be secured appropriately for all the participating partners. In a more complex situation, the cooperation between two organizations may have different security, privacy, and confidentiality policies for different projects. For example, two companies may collaborate in one project, but compete in another.

Cross-Organizational integration of heterogeneous data services: Although a global schema can simplify data integration within an organization, cross-organizational integration requires the support for autonomous evolution of heterogeneous data services from different organizations. At a base level, different organizations may use different terms and API definitions, which require a distributed ontology for flexible translation. At a abstract level (e.g., business-level interaction protocols), beyond typical web services [1], an appropriate translation mechanism is in need to support autonomous evolution.

Many ESB products have been delivered by companies and open source organizations in recent years. However, few of them can meet the above cross-organization EAI requirements very well. Most ESB products make simplified assumptions about security and heterogeneity requirements, limiting their support for cross-organizational EAI. There are a few products present the federated ESB pattern and brokered ESB pattern to support cross-organizational integration [2], but some important functions such as brokered ESBs organization, evolution flexibility, and security control of boundary information still need to be implemented and refined.

To solve the above problems, we describe a cross-organization ESB (CO-ESB) framework and software tool, called JTangSynergy 3.0, which can facilitate the integration of large scale cross organizational applications. The issues of the framework's dependability and heterogeneous data service integration have been addressed in previous papers [3] [4] [5] [6]. In this paper, we focus on our experience in using JTang-Synergy 3.0 for the design and implementation of cross-organzation e-Government (e-Gov) system integration in Hangzhou city of China. Besides the e-Gov example, JTang-Synergy has been applied in large enterprises including China Aerospace Science and Technology Group, and Alibaba (a widely used B2B marketplace).

These experiences demonstrate the advantages of JTangSynergy 3.0 in supporting cross-organization heterogeneous application integration. JTangSynergy 3.0 leverages the domain controller for messaging and coordination of the different security policies between domains. The Synergy container and the service directory in the CO-ESB container cluster facilitate the integration of component applications into the CO-ESB environment and the collaboration of components with each other.

The rest of this paper is organized as follows: section II introduces the case study of Hangzhou e-Government system integration using JTangSynergy3.0; section III shows the architecture of JTangSynergy3.0; section IV describes the application collaboration through CO-ESB; and section V is the sum-

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8314
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8314

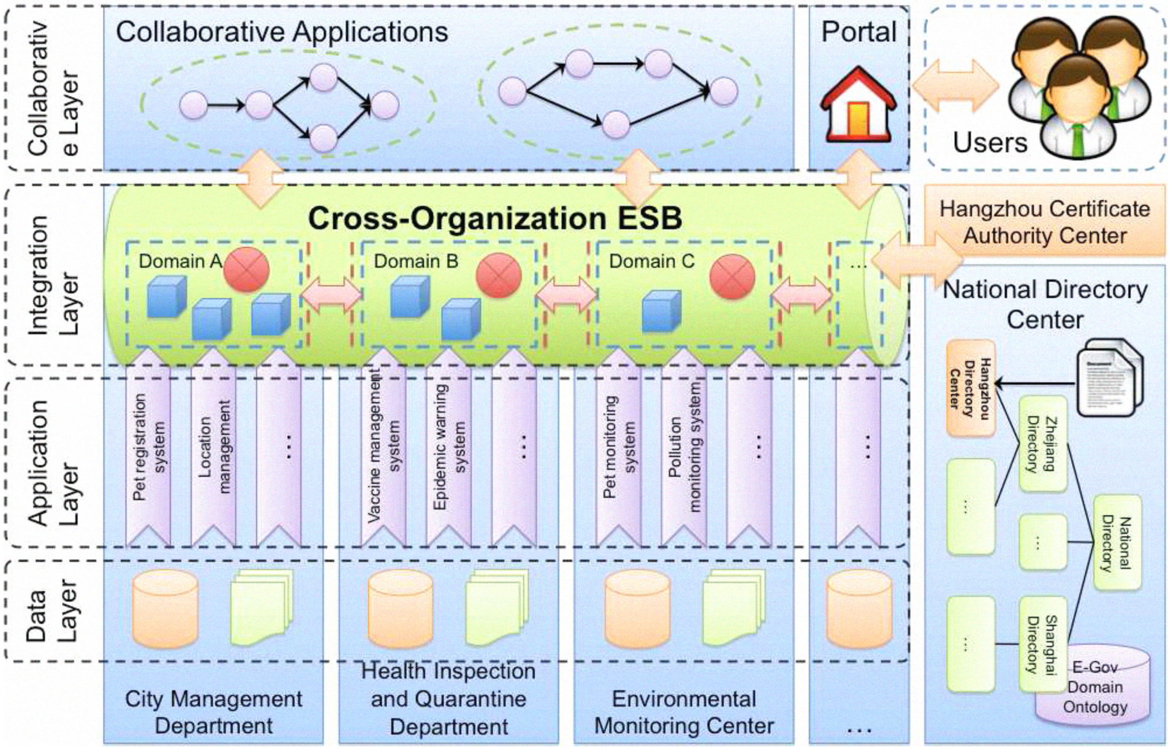


Figure 1. E-Gov system integration using JTangSynergy 3.0

mary of this paper.

II. Case Study: Hangzhou e-Government System Integration

A. Application Overview

With the rapid economic development in China, e-Gov systems grew in numbers and scope during the past ten years. Each government agency has deployed its own e-Gov systems, making it difficult for these heterogeneous e-Gov systems to communicate with each other directly because they were developed by different vendors who had little awareness of others' systems.

To help e-Gov systems from different government departments and organizations work together, Chinese government plans to build the National Directory Center (NDC), which is a hierarchical service registration center. The NDC allows a government organization to publish applications as services and to query services registered by other organizations. But a secure and loose-coupled backbone is still needed for cross-organizational application interaction. So we cooperate with the Insigma company (http://www.insigma.com.cn/) who is responsible for building the Hangzhou Directory Center (HDC, a part of NDC shown in Fig. 1) to integrate e-Gov systems in Hangzhou city, taking the digital city management project as a test bed. This project uses the JTangSynergy 3.0 platform (described in more detail in section III) as the key middleware infrastructure for a successful integration.

TABLE I. E-GOV SYSTEMS ABOUT PET MANAGEMENT

E-Gov system	Organization	Description
Pet registration system (PRS)	City Management Department	Register the pets' basic information. (Developed as an EJB)
Vaccine management system (VMS)	Health Inspection and Quarantine Department	Manage the vaccine information. (Developed using C# and exposed as Web service)
Pet monitoring system (PMS)	Environmental Monitoring Center	Statistic the number of pets in every location of Hangzhou city. (Developed based on Spring framework)
Pet security records center (PRC)	Public Security Bureau	Store the pet security records. (Developed based on MFC)

In Hangzhou, dozens of e-Gov systems, which provide various functions and are distributed in different organizations, need to be integrated. For concreteness, we take pet management as an illustrative example, which involves e-Gov component systems shown in Table I. These heterogeneous systems developed using different technologies provide specific functions of pet management. The data formats used in these systems have not been consistent. And the security policies in these organizations are also different. For example, the input message of the pet registration system (PRS), which includes much information of the pet such as the ID (stored in the RFID

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8314
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8314

chip planted in the pet), name, type, date of birth, and its owner, is implemented as a Java object. While the input message of the vaccine management system (VMS) is a SOAP message which consists of the pet's ID and date of birth. And in the City Management Department, the messages only need the signature, but in the Health Inspection and Quarantine Department, besides the signature, the message should be encrypted by RSA algorithm. Therefore without EAI system support for data integration and component interoperation, these systems could not collaborate with each other to complete any business process. If a citizen in Hangzhou wants to raise a dog, he has to submit required information to each system individually, since these systems could not exchange information with each other directly. And the data in those systems are isolated and duplicated, which may cause data inconsistency.

B. Integrating E-Gov Systems Using JTangSynergy 3.0

The heterogeneous e-Gov systems (such as the PRS and VMS) involved in the pet management were integrated using JTangSynergy 3.0 as shown in Fig. 1. In each organization, we built a CO-ESB application domain which consists of a domain controller and a CO-ESB container cluster. The domain controller (section III.A) helps to manage the distributed environment and coordinate the different security policies between organizations based on the Hangzhou certificate authority center, which includes the PKI and PMI systems. And the CO-ESB container and service directory in CO-ESB container cluster (section III.B) are responsible for integrating the heterogeneous systems and exchanging service information with HDC respectively. To enhance the message matching and service discovery, we also built the e-Gov domain ontology repository using Protégé and the OWL-plug-in, which covers 368 concepts defined in government information resource exchange standard (part 3): data interface specification (a Chinese National Standard).

Integrated by JTangSynergy 3.0, the heterogeneous e-Gov systems can collaborate with each other, securely and reliably, to achieve pet management processes (such as pet registration process), which can be easily defined by the integrated business logic model tools (section III.C). For example, during a pet registration, users do not need to cooperate with the e-Gov systems one by one. They only need to submit the related information once and invoke the pet registration process through portal, which can save users' effort and avoid data inconsistency.

Up to now, we have integrated about 35 applications in three areas (including public service, market supervision, and social management), which are distributed in 15 government organizations. And these integrated applications provide e-Gov services for about five million citizens every year, gain more than 20% improvement in the information processing efficiency and citizen satisfaction, and achieve about 75% reduction in the human resource cost for information synchronization.

III. JTangSynergy 3.0 Framework

A. Domain-based CO-ESB Environment Management

To facilitate the application integration partner network, and provide the administration flexibility and information security, we propose a domain-based CO-ESB environment management mechanism by dividing the whole application integration environment into different CO-ESB Application Domains (named as CEAD for short). CEAD is a logical administration unit and different CEAD can adopt different information administration policies. In general, it is a counterpoint to an individual organization.

Every CEAD consists of a domain controller (shown in Fig. 2) and a CO-ESB container cluster. CO-ESB container cluster (Session III.B) is designed to support large scale application integration and to manage the dynamic environment in one domain. Domain controller is an application integration gateway to bridge some special CEADs. The missions of domain controller involve domain administration, information gateway, security policy translation, cross-domain communication infrastructure and so on. Fig.2 illustrates the key components of a domain controller.

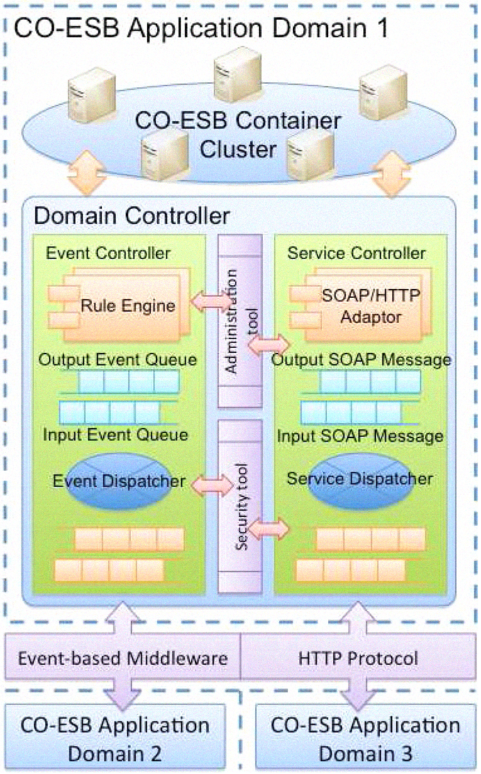


Figure 2. Key components of domain controller

1) Event Controller

First, event controller is responsible for efficiently and securely publishing the events, which are sent from the internal applications in the producer CEAD to the consumer CEAD. Second, it is also responsible for collecting the events from other CEADs and dispatching the events to its CO-ESB container cluster.

Event dispatcher is an engine to send and receive, encrypt and decrypt the events in event controller. To improve the efficiency of network transmission and protect the exchanged mes-

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8314
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8314

sages from revealing, event dispatcher sends event data using an event routing table, which contains event type, source domain, destination domain list, encryption and decryption algorithm, data expression algorithm, persistent and transaction options. Every event producer must register its event route record to its domain controller before it can publish events. Domain controller will disseminate the updates of route table to its neighbor domain controls using settled protocol. If event producer does not register his event route record and no domain default policies are settled, i.e., the event type of route table record is denoted as "*", event dispatcher will broadcast its event data using the original format. Rule engine is used to facilitate the communication between CO-ESB container cluster and domain controller.

Events received from CO-ESB container cluster or from event dispatcher will be stored in two queues. One is output event queue and the other is input event queue. Event dispatcher consumes the output events, and then encrypts the events and sends to the corresponding sending queue according to the event route table. After that, domain controller calls the APIs of event-based middleware to disseminate the events. The process for receiving events from the other domain controller is similar but converse.

2) *Service Controller*

Traditional Web service can be accessed by any requester only if requestors know their WSDL, whenever the requesters and responders are connected by HTTP. This helps simplify the EAI mechanism but incurs uncontrollable access and security issue in large scale cross-organizational application integration environment. Service controller is an efficiently configurable module to solve the above issues.

Similar to event dispatcher, service dispatcher is a module to intercept and filter all SOAP messages exchanged between different CEADs. It authenticate the Web service requester, encrypt and decrypt the SOAP message, and perform the Web service access control using the service routing table which contains the service name, service annotation, service URL, access control list (ACL) and QoS options. Service annotation is used to support the intelligent service discovery and orchestration. ACL is a list of access permission policies. Each record of ACL tells which domain can access the Web service, how to identify the requester, what privilege a requester can hold and how to encrypt and decrypt the soap message. SOAP/HTTP adapter connects the CO-ESB container cluster to the domain controller.

For instance, when a requester from the other domain triggers a web service invocation to domain D1, the service dispatcher of D1 will intercept the request and then authenticate the requester and check its privilege according to the ACL. If the request is permitted, the service dispatcher will decrypt the SOAP message using the settled algorithm in the service routing table and dispatch the SOAP message to the CO-ESB container cluster.

3) *Security and Administration Tools*

Since applications in different domains may use different security policies, both the event controller and the service controller need security tool to translate security policies. Security tool is a common security infrastructure for other modules in the domain controller. As WS-Security is a defacto standard in Web service security domain, security tool is implemented in accordance with the WS-Security 1.1. Furthermore, it provides some popular security encryption and decryption toolkits (such as AES, RSA), expressed and unexpressed algorithms (such as LZ77, RICE).

Administration tool is a system tool to manage the domain controller. It can configure and optimize the parameters of domain controller, such as the size of output event queue and the location of the file that stores the event routing table. It can identify the domain controller access requester and take actions according to the administration commands issued from the CO-ESB container cluster or the other domain controller, the updates of routing table, registration ACL and the requester, etc. Furthermore administration tool can be used as a transaction coordinator or participant to support the distributed EAI transaction.

B. *CO-ESB Container Cluster*

Besides the domain controller, every CEAD contains a CO-ESB container cluster, which can improve the reliability and performance. Each node of the cluster consists of a CO-ESB container (named Synergy container) and a service directory (shown in Fig. 3). The Synergy container provides the key functions of CO-ESB such as adaptor framework, message routing and transformation. And the service directory is used for service registration, service query, service lifecycle management and so on. And considering the extensibility, the JTangSynergy 3.0 is developed based on OSGi framework and each module is implemented as OSGi bundle.

1) *Synergy Container*

To make the JTangSynergy more dependable, the CO-ESB container cluster refines the traditional master-slave cluster architecture and classifies the Synergy container into four types: prime container, delegate container, backup container and standard container. Generally, a CO-ESB container cluster is constructed by a prime container, a backup container and several standard containers. The backup container is upgraded to the delegate container only after the prime container is crashed.

The prime container (or delegate container) uses the synergy environment manager (SEM) module to manage the cluster configuration and monitor its membership. And the SEM implements an algorithm called prime and successor list (PSL) [3] for rapid recovery of the container failures. Except for SEM module, the other key functional modules in all types of Synergy container are listed below:

- The adaptor framework contains rich adaptors, which can provide various connection mechanisms for external services and applications.
- The message broker module provides two core functions of ESB: message routing and transformation. The router submodule has two routing mechanisms [7]: itinerary-based routing and content-base routing. And the

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8314
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8314

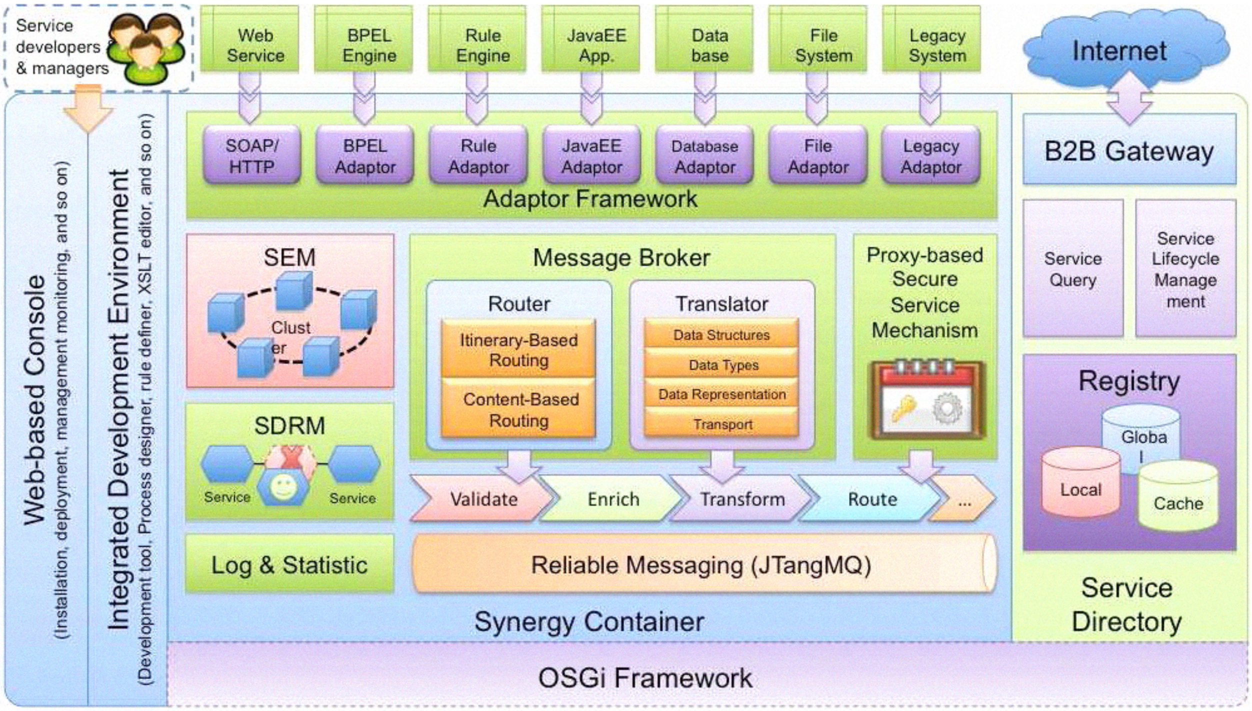


Figure 3. Synergy container and service direcotry

translator submodule can be used to transform messages in four layers [8]: data structures, data types, data representation and transport.

- The proxy-based secure service mechanism (PBSSM) module provides an end-to-end security solution, which is similar to the CORBA Security Service Specification mechanism. Different from the security tool in the domain controller, the PBSSM is used to secure messages within a CEAD.
- The reliable messaging module leverages our JMS product JTangMQ to implement WS-ReliableMessaging standard, which can provide reliable message delivering mechanism for JTangSynergy.
- The dynamic service replacement (DSR) [3] module is used to detect errors during composite service execution and find an equivalent service to replace the broken component services.

2) Service Directory

Different from the previous version of JTangSynergy [3], the service registration has been moved out of Synergy container and is built as an individual module named service directory, which is developed according to the UDDI 2.0 standard. The reasons of separating the service directory from Synergy container are: a) make functional modules more independent and clear, b) make the registered service information more reliable and easy to manage in the distributed environment, c) provide an interface to communicate with the HDC.

The service query module can be used to query the service information either in registry of service directory or in the HDC based on the e-Gov domain ontology repository. The service lifecycle management module is responsible for the service deployment, start, stop and shutdown. To support cross-organizational integration, the registry module contains three types of service registries. The service descriptions registered in the local registry can only be accessed inside the same domain. The service descriptions registered in the global registry, which is also registered in the HDC, can be accessed by any other service in the CO-ESB. And the cache registry is used to cache the service descriptions getting from the HDC. Each service description in the cache registry has a time-to-live (TTL) property to keep its valid time.

C. Administration Console and IDE

To enhance the usability, JTangSynergy also has an administration console and integrated develop environment (IDE). The administration console is a web-based administration tool, by which administrator can configure the component parameters, monitor the system states, tune the performance and perform the security management. IDE is an Eclipse-based development tool for integration application developers, which is integrated with some popular business logic model tools, such as workflow designer (shown in Fig. 4), business rule definer, message transformation editor, and adapter development wizard. With IDE, developers can easily build new service adapters or applications, debug and deploy them into the JTangSynergy CO-ESB platform.

IV. CO-ESB SUPPORT FOR CROSS-ORGANIZATION APPLICATION COLLABORATION

The application collaboration is one of the most challenging issues in cross-organization application integration. To illu-

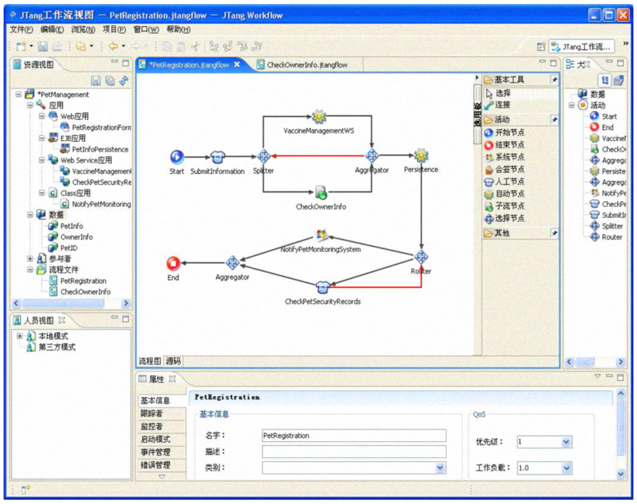
Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8314
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8314

strate the secure and reliable cross-organization application collaboration using JTangSynergy 3.0 and HDC, we take the pet registration system (PRS) and the vaccine management system (VMS) as an example (shown in Fig. 5).

A. Application integration

Connection, configuration, orchestration and deployment are the four steps to integrate applications in CO-ESB application domains.

Firstly, we need to analyze every application, and connect every application into CO-ESB container by developing or using the corresponding adapters. In the pet management scenario, the PRS was developed as an EJB that can only send JMS message and the VMS was developed using C# that can be accessed through Web service technology. So a JavaEE adaptor and a SOAP/HTTP adaptor are needed for the PRS and the VMS respectively. To connect an application to the corresponding adaptor, we should configure the adaptor to add an endpoint in the CO-ESB environment, which will expose the application as a service. Secondly, we should configure the security policies using the security tool in the domain controller since the applications in different organizations also have their own security policies. For example, the input message of the VMS should be encrypted by RSA algorithm, which cannot be provided by the PRS. So we should add the RSA security policy in the domain of PRS, by which the PRS can interoperate with VMS using RSA encrypted message. Thirdly, we use IDE workflow design tool (shown in Fig. 4) and rule definer tool to orchestrate the application integration process (such as the pet registration process) and define message mediation rules (such as message validation, message enrichment, message routing and message securing, shown in the left of Fig. 5). At last, we deploy this integration project, which include adaptor configuration, security policy configuration, integration process and the message mediation rules etc., into the related CO-ESB application domains after debugging and evaluation.



PetRegistration

名字： PetRegistration
描述：
类别：
优先级： 1
工作负载： 1.0

Figure 4. User Interface of integrated workflow designer in JTangSynergy

B. Application Interoperability Support

Service registration and domain controller configuration are the two steps to support application interoperability between different CEADs.

To make an application accessible as a service in CO-ESB environment, a service description is registered in the service directory by Synergy container after the deployment. And the service directory will further register this service description to HDC if the corresponding service allows to be accessed by other applications outside its domain. For instance, the service description of the VMS, which need to be invoked by PRS in the pet registration process, were registered in the HDC.

Supported by the HDC and service directory, the applications connected into the CO-ESB are visible to each other as services. But the service contracts between the heterogeneous systems (such as the PRS and VMS) may be still not compatible. To make the heterogeneous systems interoperable crossing organizations, Synergy container and the domain controller can process the message using the message routing rules and security control policies. Any message delivered in the CO-ESB environment should be wrapped as a self-describing canonical message, which contains the meta information in the header and the original message content in the body (shown in the right of Fig. 5). The meta information contains mediation rule references, security policies and reliability policies, which help Synergy container and domain controller to deliver message correctly, securely and reliably. The mediation rule references point to message mediation rules that are used to route and translate the message automatically. The security policies define the security mechanisms used by the PBSSM module of Synergy container and the security tools of domain controller to build secure and transparent communication. And the reliability polices, which is based on WS-ReliableMessaging standard, are used to guarantee the reliability of message exchanging in the CO-ESB environment.

To illustrate how Synergy container and domain controller can work together to support cross-organizational application interoperation, we describe the interoperation steps between the PRS and VMS in detail. Step 1, the PRS sent the JMS message to the JavaEE adaptor, which would wrap the request message as a canonical message and send it to message broker of Synergy container. Step 2, Synergy container processed the message based on the message routing rules and delivered the message to the domain controller. Step 3, when the canonical message arrives at the domain controller, the domain controller got the VMS's public key from Hangzhou Certificate Authority Center, encrypted the message using RSA algorithm, and then sent the message to the destination domain. Step 4, the domain controller in the destination domain forwarded the message to a Synergy container. And the Synergy container uses the message translator to transform the message from Java object into XML source type. Step 5, the SOAP/HTTP adaptor got the message body out of the canonical message and sent it to the VMS.

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8314
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8314

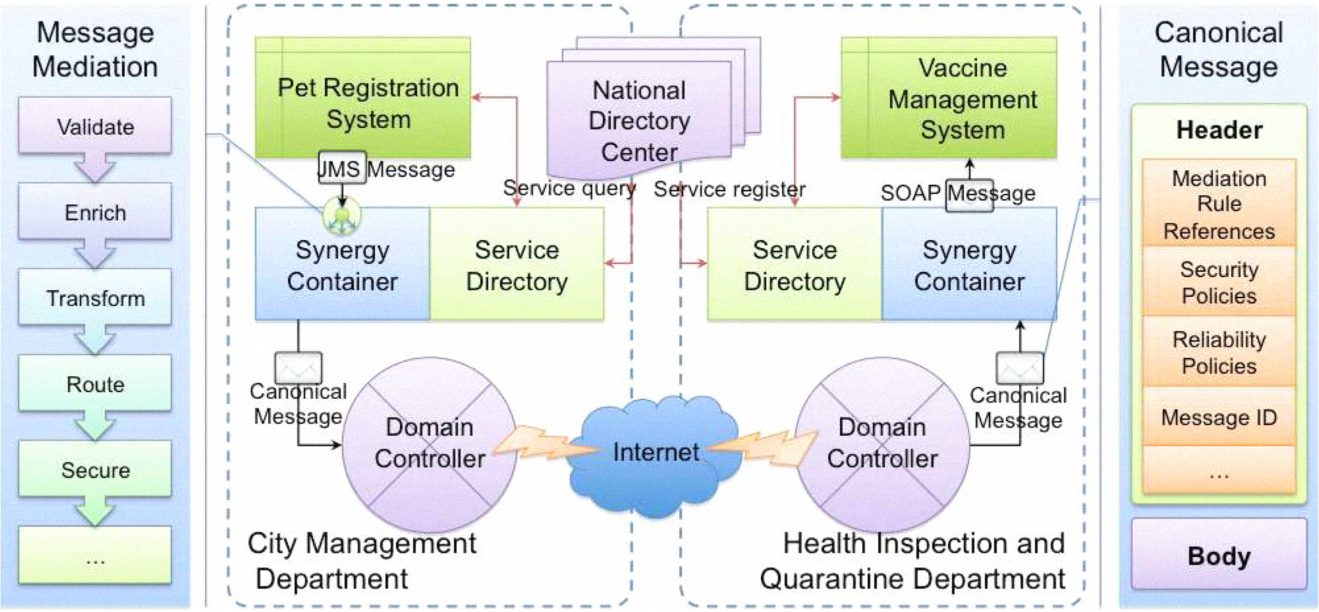


Figure 5. Application collaboration crossing organizations

In general, JTangSynergy3.0 supples good mechanisms to meet two challenges in the cross-organization application integration. HDC and service directory make cross-organizational service accessible. Domain controller can manage the distributed application environment appropriately, implement the secure and reliable service interoperability, and supply an appropriate heterogeneous data translation mechanism.

V. Conclusion

Large scale distributed application management and integration of heterogeneous data services are two challenges in the cross-organization application integration, which cannot be overcome by traditional ESB. In this paper, we introduce a cross-organizational ESB framework and software tool named JTangSynergy 3.0 that supports large scale cross-organizational application. The architecture of JTangSynergy 3.0 and its implementation (domain controller, Synergy container and service directory) provides several good mechanisms and functionalities to support the distributed environment management and heterogeneous application integration. The effectiveness of JTangSynergy 3.0 in integrating cross-organizational applications is demonstrated by Hangzhou e-Gov integration project, which can provides e-Gov services for about five million citizens every year.

Acknowledgment

The work has been supported by the National Supporting Program, China (No.2006BAH02A01), the National High-Tech. R&D Program, China (No.2006AA01Z170, No.2007AA01Z124), the National Natural Science Foundation, China (No. 60703042), and the and Zhejiang Natural Science Foundation of China (No.Y106045)

References

[1] Benatallah B., Reza H., Nezhad M., Casati F., Toumani F. and Ponge J. Service Mosaic: A Model-Driven Framework for Web Services Life-Cycle Management. IEEE Internet Computing, vol.10, no. 4, 2006, pp. 55-63.

[2] Grund V., Rexroas C. Enterprise Service Bus implementation patterns. http://www.ibm.com/developerworks/websphere/library/techarticles/0712_grund/0712_grund.html, 2007.

[3] Yin J.W., Chen H.W, Deng S.G, Wu Z.H, Pu C. A Dependable ESB Framework for Service Integration. IEEE Internet Computing, vol 13(2), 2009, pp.26-34.

[4] Shi D.C.,Yin J.W., Li Y.Y., Fei Q.J., Dong J.X. JTangPS: An RDF-based Publish/Subscribe System. 3rd International Conference on Semantics, Knowledge, Grid, IEEE Press, 2007, pp342-345.

[5] Chen H.J, Wang Y.M., Wu Z.H., et al. Towards a Semantic Web of Relational Databases: A Practical Semantic Toolkit and an In-Use Case from Traditional Chinese Medicine. 5th International Semantic Web Conference, IEEE Press, 2006, pp 750-763.

[6] Zhou X.Z., Wu Z.H. et al. Ontology Development for Unified Traditional Chinese Medical Language System. Journal of Artificial Intelligence in Medicine, vol 32,no.1, 2004, pp 183-194.

[7] Chappell D. Enterprise Service Bus. O'Reilly, 2004.

[8] Hohpe G., Woolf B. Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. Addison-Wesley, 2003.

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8314
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8314