Flexible Business Process Integration for Clusters of Small-Medium Sized Enterprises in Heterogenous Environment

Bo Jiang
College of Computer and Information Engineering, Zhejiang Gongshang University, Hangzhou, China
Email: nancybjiang@zju.edu.cn

Yun Ling
College of Computer and Information Engineering, Zhejiang Gongshang University, Hangzhou, China
Email: yling@mail.zjgsu.edu.cn

***Abstract*—Industry clusters-groups of geographically proximate enterprises in the same industry are a striking feature of the geography of economic activity and greatly impact the overall economic development. However, most of them are low-cost-based clusters that are composed of small-medium enterprises which needs social electronic services and flexible business process integration. In this paper, we present an on-demand information service platform for small-medium enterprise clusters and propose loosely-coupled service-oriented business process integration scheme for this platform. Several key techniques such as business application definition language and graphical representation for integration in heterogenous environment, packaging and executive strategy for integrating services, dependency set and context of integration service, and mapping translation based on synonymous semantic definition are proposed. Professional services can be provided via this platform to promote the enterprises' innovative ability and help business to gain competitive advantages. The platform and the related techniques have been applied in realistic textile industrial clusters.**

***Index Terms*—industry clusters, business process integration, web service composition**

I. INTRODUCTION

Today's economic map of the world is dominated by what are called clusters [1]. Industrial clusters have not only become basic framework of regional economy, but also main form of spatial arrangement of global economy. It is known that clusters may help small-medium enterprises (SME) that concentrated to produce neighbor effect and social effect to gain competitive advantages. The question is how the small-medium enterprises, who based themselves upon lower phase of technology gradient, identify and attain the outer information and technology resource to strengthen their own capabilities with the background of global industry transfer. The issue has become more and more important for the industrial clusters of small-medium enterprises, especially in the developing countries.

According to the generalization of enterprise cluster and the analysis of their current situation and existing problems, from the angle of information administration, many problems arising in those small-medium enterprises need to be solved. Small-medium enterprises should share information and services. Therefore, providing information service platform and integrate social and corporate electronic services may solve the problems. Many experts [2, 3] research on the architecture and integrated environment of virtual enterprises. Based on information service integration, it is very crucial to quickly integrate business process among small and medium-sized enterprises (SMEs) in cluster. However, corporate business process integration is always carried out under the environment of heterogeneous geographical distribution, software and hardware platform. Former integration solutions, such as DCOM, CORBA, Java RMI, etc., are complicated to realize, and relate to specific platform and language [4]. Their integrations take time and vigor, and cannot meet the "instant" integration demands of small and medium-sized enterprises in cluster. Web service composition, by binding two or more existing services into a new one, is emerging as a promising technology for supporting large-scale, sophisticated business process integration in a variety of complex e-science or e-business domains. With highly dynamic nature of the internet, automated composition approaches have been the most promising methods to be applied in real world contexts. Currently, there exists different research efforts on service composition; such as Petri net based composition approaches [5], AI planning-based approaches [6, 7], context and non-functional properties-based approaches [8, 9, 10]. However, despite the merits and the importance of existing algorithms for service composition, they can hardly meet the requirement of quick integration in heterogenous environment.

In this paper, we present a business process integration framework and related schemes based on dynamic web services for quickly and automatically creating process models for resolving dynamic distributed business integration problems for small-medium enterprises in heterogenous environment. E_BPEL, a suitable definitional language for business application environment of cluster SMEs and related graphical

representation are proposed, which supports dynamic service through semantic web services. Related packaging and executive strategy for integrating services is presented. Further, based on dependency set and context of integration service that we proposed, executing state and relevancy of web services instances are maintained. To find and integrate qualified services more effectively, mapping translation based on synonymous semantic definition in heterogenous environment is proposed. The framework and schemes are realized and utilized by the organization of textile industry cluster.

This paper is as follows: Section 2 depicts the information service platform model for clusters of small-medium enterprises. Section 3 discusses the technique of business process integration based on dynamic web services integration. Finally, Section 4 concludes the paper.

II. INFORMATION SERVICE PLATFORM MODEL FOR CLUSTERS OF SMALL-MEDIUM ENTERPRISES

To solve the problems that caused by the shortage of socialized information service in the development of the cluster, we construct an open and service-oriented supporting platform. As shown in Fig. 1, the system is divided into 3 functional levels: basic service facilities, public service, and integrated service.

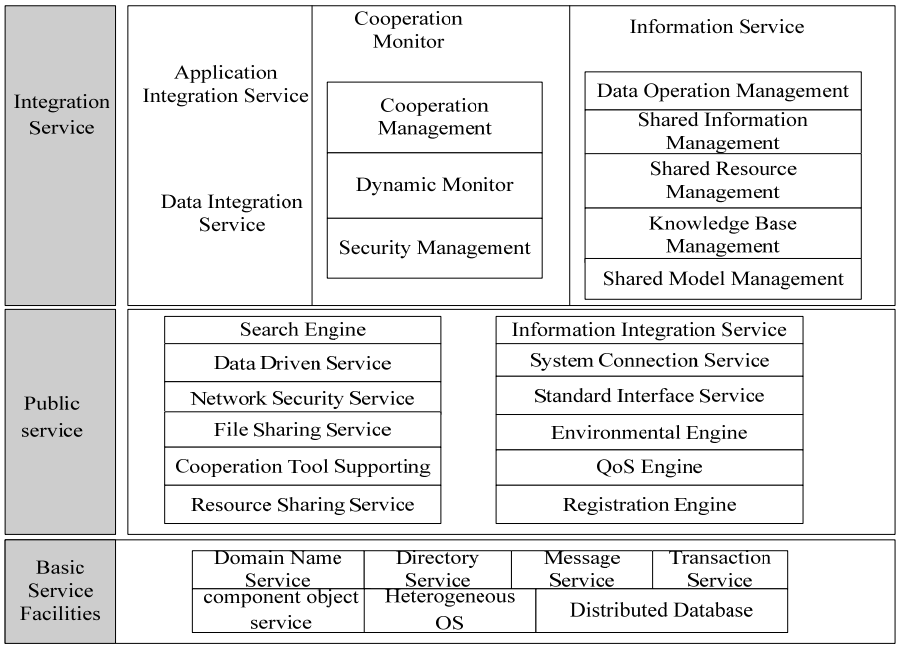

Figure 1. Information service platform model.

(1)Basic service facilities: by making use of TCP/IP, it shields the physical connection characteristics and provides transparent information communication service. The distributed heterogeneous information and data bases are packaged according to the criteria of distributed computing. By the way of service, synchronous mutual operation at object level in the network environment is realized. At the same time, the basic service level provides the elementary domain name service, directory service, transaction service, component object service and message service for the platform.

(2)Public service: it provides the enterprises a common set of public services, such as search engine, data driven service, network security service, file sharing service, cooperation tool supporting, resource sharing service, information integration service, system connection service, standard interface service, environmental engine, QoS engine and registration engine.

(3)Integration service: two main resources of enterprises are application service and data service. Application service provides software resources for the networks, while data service provides accessible database resources. The platform integrated service level realizes the registration, issue, query, modification, tailor, and configuration for the information and application resources. Application integrated service realized web-based encapsulation, directory service and the registration, issue, query, modification and call of the application resources which are based on standard interface technology. And they complete the management and configuration of the resources based on workflow technology. Data integration service covers information model management, shared data management, data manipulation management, information integration management, data relation management, knowledge structure management and so on.

By adopting component-based functional model, the development of the application system in enterprises will be much more flexible and can construct and configure different application system according to different requirement and scale. Thus it is not necessary for each network information service using all service

components. They can make use of only the lower-layer components rather than the higher-layer components.

III. BUSINESS PROCESS INTEGRATION BASED ON DYNAMIC WEB SERVICES

Web services technology has irrelevance of platform and language, while service components are loosely coupled. Web services technology has enabled enterprises to release themselves, but also to find out potential partners dynamically. More importantly, through web service, cooperative enterprises realize cross-platform and instant business process integration in deed. This is realized by combining the characteristics of cluster SMEs, and studying the dynamic integration of web service.

Generally speaking, present research of web service integration is based on traditional integration mechanism of workflow, for example, WISE system, WebTransact and so on. These researches only adopt management methods of workflow during the web service integration. However, we must consider that,

- Traditional workflow is mostly operated in a predictable and repetitive way, while web service integration in cluster SMEs has to be highly dynamic. Every day, as for a certain service needed by enterprise clusters, some newer and better services may appear. Hence, as for some parts of integrated process, appropriate services should be searched dynamically during operation, and should be instantly integrated into the process.
- Every web service in the integration is fully autonomous module, and external process does not have more control ability over it. Therefore, complicated transaction processing mechanism in present workflow cannot be directly applied to the transaction processing of web service integration.
- Stateless feature of web service requires the integrated system to maintain relevance for the operation of integration service.
- Proper security management is very important.

Therefore, process integration based on the dynamic integration of web service requires that,

- Integration service is easy to deploy, and is put into use as soon as possible;
- Dynamic search and binding of cooperative service is permitted during the operation of integration service;
- Feasible transaction processing and safety control are available.

In order to meet above-mentioned requirements, we put forward a business process integration framework based on the dynamic integration of web services.

A. Business Process Integration Framework

The realization of business process integration based on dynamic web services integration meets the following demands:

(1) Composition services are easily to be deployed and used quickly.

(2) Finding and binding of the cooperative service dynamically when the integration services is running.

(3) Provide feasible transaction process and security control.

Figure 2 shows the business workflow integration based on the dynamic web service composition.

The structure provides supports for each phases, definition, deploying, executing supervision, management and so on. The functional models are as follows:

Definition block of integration service: it provides support for the definition of integration service, and analyzes the definition then stores it into service base.

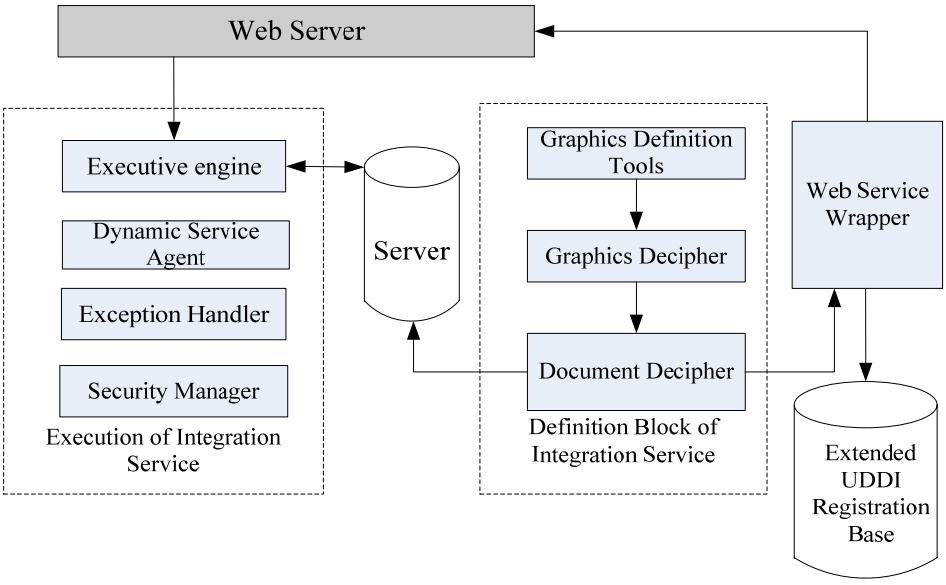

Figure 2. Business workflow integration based on dynamic web service.

Service base: stores analyzed conformity service definition.

Web service wrapper: wraps the user-defined service into common web service, provides the operation and realization in the service and bring forth the announcement and deploy of web service.

Executive engine: execute the requested samples, access the operations in the service base, and call the web service node which takes part in the process.

Dynamic service agent: handle the binding request during the execution of dynamic service. The function covers dynamic inquiry, optimum selection and the switch and call of dynamic service messages.

Exception handler: deals with the abnormal problems such as the error of service call, the failure of dynamic binding. It also includes event process, journal and recover and so on.

Authorizing controller: check up the ID and limits of authority of the users, provides security protection mechanism, and execute necessary encryption and decipher.

Supervisor: supervises the execution of the system, provides the elimination of short-time service examples and recovery of fault at system level.

Extended UDDI registration base: in addition to normal function of UDDI registration base, it supports the announcement of semantic web service.

- Definition and Release Process of Integration Service

With graphical tools, users give out graphical representation of integration service. When saving, graphical representation is transformed into document schema definition language. At the generation phase of integration service, this document is resolved. The resolving includes as follows.

Integration service, which is expressed by definition language, shall be converted to service library equivalently, and be saved, in order not to resolve the document schema definition language during implementation.

Relevant integrated web service is generated according to the definition of service, including the operation of web service and corresponding WSDL document.

Meanwhile, this web service is deployed, and released to the extended UDDI registry. This procedure is marked with solid line in the diagram.

- Execution Process of Integration Service

Execution process of integration service is shown with broken line in the diagram. When a certain web service is invoked, service request is resolved; the system determines invoked web service instance, and sends it to execution engine of integration service. According to the definition in service library, execution engine will invoke relevant web service that is deployed in the participant's site. In face of dynamic binding service during operation, the engine sends a request to dynamic service agent. According to user's definition in the extended UDDI registry, dynamic service agent searches for qualified service, chooses the optimal invocation, and returns the execution result to the engine. During the operation of integration service, potential exceptions are reported to exception handler, which is in charge of corresponding recovery and compensating operations.

B. Definition of Integration Service

The first step is to determine the definition of integration service. IBM and Microsoft have jointly raised Business Process Execution Language for Web Services (BPEL4WS or BPEL in short). At present, this has become an industry standard for definitional language of web service integration. Though BPEL provides a perfect mechanism for web service integration, BPEL has its disadvantages. BPEL only supports the static binding of existing web services. The binding between process and service is recognized to be a known condition, so BPEL doesn't have good support for dynamism. It is very difficult to use the large and complicated BPEL in ordinary application environment of cluster SMEs. Therefore, we must simplify BPEL, enhance its support for dynamic service through semantic web services, and construct a suitable definitional language for application environment of cluster SMEs, which can be named e_BPEL. e_BPEL can offer graphical representation of integration service definition.

- Basic components of integration service definition

As is shown in Figure 3, integration service definition is illustrated by graphics. Besides starting point and end-point, the process of integration service definition also include transition line, service invocation and message container. (Data lines in the diagram do not really exist in graphic representation.)

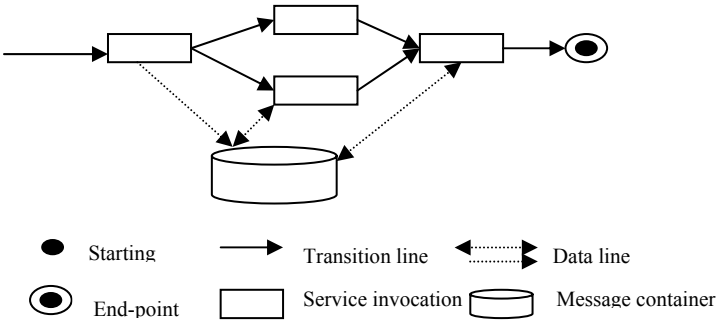

Figure 3. Basic structure of integration service definition.

(1) Message container is a container of all messages in overall integrative service, including the changes of data state in the process, receiving and sending of messages.

(2) Transition line. Complex control structure of the process is realized by setting the attributes of transition line and service calling point. Service process defines 2 types— unconditional transition and conditional transition. Unconditional transition only shows the sequence of 2 service invocations; conditional transition sets relevant attributes of transition line, stipulating that the process can finish this transition only after meeting these conditions. Through conditional transition and unconditional transition, many common structures are defined, such as sequence, selection and parallel, etc.

(3) Service calling point defines invoked action service. The definition of service calling point usually comprises service type, binding information of service and compensating operation. Binding information of service includes operation name to invoke service, binding address of service, input and output parameters of service operation, etc.

- Types of integration service

According to the type of invoking operation, integration service falls into receiving of asynchronous messages, service invoke and result replay; according to dynamism, there are static binding node, multi-binding node and dynamic binding node.

(1) Receive. Business process blocking is waiting for matching messages.

(2) Invoke. Business process invokes participants' one-way or request-response operation.

(3) Reply. Business process sends messages, and recovers the received messages of Receive.

(4) Receive corresponds with Reply. When the name and address binding of Reply are the same with that of foregoing Receive, this Reply is considered to correspond with that Receive. The process accepts input through this Receive, and outputs its response through corresponding Reply.

(5) Static binding. Bind address of invoked service has been determined in the process.

(6) Multi-binding. Bind address of invoked service is not a unique value, but a group of address. This group of services offers operations with equivalent functions.

(7) Dynamic binding. Invoked service is not bound in detail. There is only description information of this service and operation. The service is bound during the process execution.

An integration service process starts from Receive, and ends with Reply or Invoke. Any types of service calling points are in the middle. When a process ends with Reply, it means that integration service will probably get synchronous response. Otherwise, when the execution result is returned asynchronously, the process needs to invoke the proper message receiving operation of result recipient. It should be pointed out that not all Receive operations have corresponding Reply, for example, asynchronous response waiting for an Invoke in the process.

Generally speaking, there is mainly static binding service in the process, where the provider of this service point has been determined. However, under some circumstances, it is improper to give a definite binding. For example, the same role in cluster SMEs may find several participants; but when all participants may not be able to dispose the service entirely, the process can define multi-binding at corresponding service calling points, and bind the invoked service operations to the implementation of these participants. During invoking, the process distributes service load evenly and invoke all bindings in turn; or the process invokes service load of every binding point according to a certain priority. In this way, the integration service process achieves better load balance and higher response speed. Moreover, it is helpful to search the optimal service dynamic binding during the execution of process.

C. Packaging and Executive Strategy of Integration Service

- Generation of integration web service

In order to put integration service into use more quickly, the system deploys and releases the integration service as a separate web service, and generates corresponding web service according to user's definition of integration service. This service offers 2 types of operation, that is, request-response type and one-way operation. The system generates operations according to following rules.

(1) Request-response type operation. If Receive node defined by integration service has its corresponding Reply node, this node is packaged into a request-response type operation to generate web service. When invoked, this operation returns the result at once. Input message of the operation is the input information defined by Receive node, while output message is the output information defined by Reply.

(2) One-way operation. If Receive node defined by integration service doesn't have its corresponding Reply node, this Receive node is packaged into a one-way operation. One-way operation only has input, and its input message is the same as the input of Receive node. This kind of operation always serves as asynchronous message port for recipients of integration service.

In this way, integration service releases Receive, all accepting nodes of asynchronous messages, as the operation of web service. Through this web service,

consumers (service users of cluster SMEs) invoke corresponding integration service; by invoking the operation of this web service, participants of integration service (cluster SME members) pass asynchronous messages of integration service.

- Maintenance of executing state and relevancy

It is worth noting that above-mentioned web service is generated based on the definition of integration service. According to the model of integration service, the system generates and deploys a corresponding web service; it is not true that an instance of integration service generates a web service. Meanwhile, web service itself is stateless, and doesn't maintain relevant information of two separate invocations; two operations of web service are also independent, and invoke operations of web service in any order. However, as for an integration service, many instances are operating at the same time. As a result, when a message comes, the system must decide to create a new instance, or start an existing instance, and ensure that the message is sent to the correct instance of integration service. The life cycle of an instance starts from the creation of the instance, and ends with the revocation of the instance (the task is finished normally, or ends with fault). During this stage, the system must maintain the data and executing state of this instance. Within the life cycle of an instance, execution of the instance may pause for asynchronous messages. When the message comes, the system must start corresponding instance, and restore to the operating state before the pause, including the value of data and executing state of integration service process. Within the text integration framework, we make use of dependency set and context of integration service, so as to solve these two problems.

(1) Dependency set. Dependency set is a message attribute, or a group of message attributes in the message container. According to values of these attributes, the system execution can accurately distinguish different instances of an integration service. For example, a buyer/seller integration service process can use the order numbers as dependency set. According to order numbers, the system decides to start a new instance, or continue an existing instance for the trade.

(2) Context of integration service. Context of integration service (context in short) maintains history data information and executing state of instances, and thus guarantees continuity and consistency of instance execution in integration service. When an instance of integration service is created, a relevant context instance is generated, and the context changes with the status of service instance. When running instance pauses for asynchronous messages, context of the instance record current data and executing state, including operation nodes to be invoked at the next step, etc. When the waited asynchronous message arrives, corresponding service instance is activated; the system utilizes context information of this instance, and recovers its operating state. The structure of service context is as shown in Figure 4.

Service instance number	Instance number of message container	Operation node for the next step

Figure 4. Structure of service context.

Service context doesn't maintain the data of instance directly, but carries out indirect maintenance by referring to the message container of the instance. As has been said before, data changes of operating service instance are all reflected by message container. When service pauses for asynchronous messages, the value of message container is saved in database enduringly, and thus data information of service context is lasting. When the service instance is restarted, service context makes use of new received messages, and renews relevant values of the message container.

It should be pointed out that though service context contains the information of next-step operation nodes, the system doesn't really invoke next-step operation nodes according to this information. This information is to validate, because the system generates operations of integration web service according to the definition of integration service. In terms of executing the integration service, operations of integration web service are entry points for asynchronous messages at all stages of executing the integration service. Operations themselves contain positional information during the execution of integration service. Therefore, according to invoked web service operations, the system is able to decide the next step action inside the integration service. However, statelessness of web service cannot ensure that instances are always invoked in the correct order. Consequently, when one operation of integration web service is invoked, the system compares this operation with next-step operation nodes in service context of the instance to be started. In the case of disagreement, this instance will be inhibited from starting, and a failure report will be sent to the system.

When the operation of a web service is invoked, the actual executing process is shown in Figure 5.

In this way, the system turns the integration service into a separate web service, and acquires simple and immediate deployment. Besides, integrated web service is deployed to any server that supports web service, and thus makes full use of some supporting functions of the server, for example, message management, etc.

D. Strategies of Dynamic Service Management

- Structure and functional module of dynamic service agent

Dynamic service management is realized mainly through dynamic service agent in the system. During operation, on the basis of service description, the agent searches, binds and invokes proper service. Within the integrated framework implementation, we use DAML-S to describe the web service to be dynamically bound, resort to the semantic description ability of DAML-S, and

find qualified services more accurately and effectively. Isomerism exists between found service and the service defined by integration service, such as synonymous methods, parameters and so on, so proper mapping translation is necessary during operation.

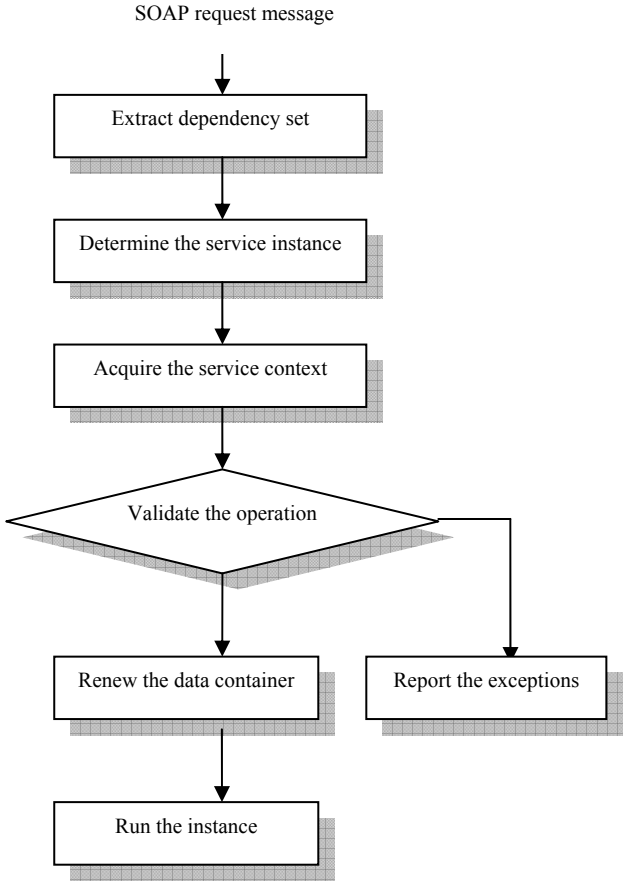

Figure 5. Operational process of web service.

Figure 6 is structure chart of dynamic service agent. Its main functional modules include as follows.

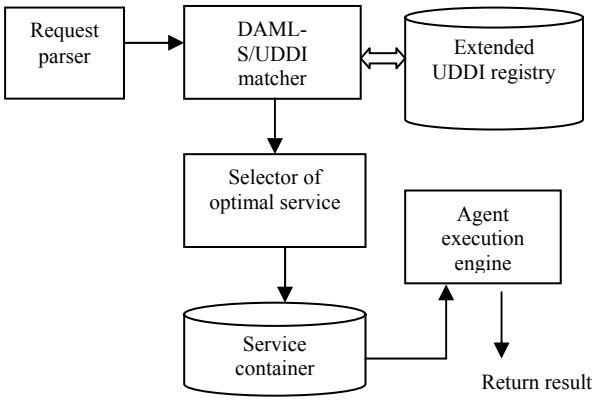

Figure 6. Structure of dynamic service agent.

(1) DAML-S/UDDI matcher. According to DAML-S service description, this DAML-S/UDDI matcher searches for qualified service in the extended UDDI service registry.

(2) Selector of optimal service. According to service matching degree, expenses and response time, etc., this selector ranks found service set.

(3) Service container. Service container saves the detailed binding of found services, as well as XSLT document, which records mapping relation between found service and the service defined by the process.

(4) Agent execution engine carries out the invocation of dynamic service.

- The implementation of dynamic service agent

The implementation of dynamic service agent is as follows.

(1) Before executing a dynamic service calling point, the execution engine of integration service requests a dynamic agent instance first.

(2) Dynamic agent instance inputs service description into a DAML－S/UDDI matcher, and the matcher searches for qualified service in the extended registry.

(3) The matcher ranks found services in the selector of optimal service, and then inputs the rank and mapping XSLT file of every service into the service container.

(4) When this dynamic service calling point is being executed, the execution engine of integration service invokes generated dynamic agent instance to execute.

(5) Dynamic agent starts agent execution engine, and the latter takes out a service binding from the service container in turn. If not usable, the service is deleted from the service container; and the next one is invoked, until the service container becomes empty (when the service agent reports to the system the failure of "dynamic service binding"), or until a service is invoked successfully. During operation, dynamic agent carries out heterogeneous solutions based on attached XSLT document.

(6) Dynamic agent returns the result of service invocation to the execution engine of integration service.

In order to improve efficiency, the system can require service agent to store the service container permanently. During operation, service agent may directly use the service in permanent container, in order to quicken the response; or the system may clearly request dynamic search, and let the service agent search for the newest service.

E. Other Management Strategies

Besides above-mentioned management mechanism of integration service, the system built on integrated framework should introduce some other management mechanisms, including

(1) Transaction management. It offers executing mechanism of exception handling and transaction processing. In the case of exceptions, exception handler carries out relevant transaction processing according to

the definition of process, and performs compensating or recovery operation. Transaction processing adopts 2 mechanisms, that is, single-point service compensation and service compensation of business area. Single-point service compensation is to define clear compensating or recovery operation for a single service; service compensation of business area is to define an overall compensating process for a section of process.

(2) Safety control. SOAP header message signature is invoked by web service to validate the identity of the invoker. Key information of messages is encrypted. XML and role-based access control mechanism is in place to enhance the safety control.

(3) System monitoring. Operating state of every integration service instance can be viewed. Overdue service running instances can be revoked manually, for example, an overdue running instance waiting for asynchronous message. Monitoring module also provides recovery for system-level failure.

IV. CONCLUSIONS

With the development of information technology and industry clusters of small-medium sized enterprises, it is significant to develop a network information service platform and dynamic business process integration. The flexible business process integration framework and schemes that support dynamic service integration are presented and proved can promote the integration flexibility and efficiency for small-medium clusters in heterogenous environment. Furthermore, related techniques are applied in texttile industry clusters. The development of the information service platform for clusters of small-medium enterprises in heterogeneous and the flexible business process integration will greatly promote the development of the small-medium enterprises clusters.

ACKNOWLEDGMENT

We are grateful for the foundation support by Zhejiang Provincial Natural Science Foundation of China under Grant No. Y106734.

REFERENCES

[1] Michael E. Porter, "Clusters and the new economics of competition," *Harvard Business Review*, Boston, Vol. 76, 1998, pp. 77-90.

[2] Kaijun Ren, Xiao Liu, et al, "A QSQL-Based collaboration framework to support automatic service composition and workflow execution," *Proc. of Grid and Pervasive Computing Workshops*, July 2008, pp. 87-92.

[3] Xiaofei Xu, Decheng Zhan, "The development of dynamic alliance and integration supporting system," *CIMS*, 2003(1), pp. 9-13.

[4] Piccinelli G., Zirpins, C., Lamersdorf W., "The FRESCO framework: an overview," *Proceedings of the 2003 Symposium on Applications and the Internet Workshops*. IEEE Computer Society (2003), pp.120-126.

[5] Dmytro Zhovtobryukh, "A Petri Net-based approach for automated goal-driven Web Service composition," *Simulation*. 83(1), 2007, pp. 33-63.

[6] Evren Sirin, et al., " HTN planning for Web Service composition using SHOP2," *Journal of Web Semantics*, 1(4), 2004, pp. 377-396.

[7] Incheon Paik, et al., "Automatic Web Services composition using combining HTN and CSP," *Proc. of the Seventh International Conference on Computer and Information Technology*. Fukushima, Japan, October 2007, pp.206-211.

[8] Swaroop Kalasapur, et al., "Dynamic service composition in pervasive computing," *IEEE Transaction on Parallel and Distributed Systems*, 18(7), 2007, pp. 907-918.

[9] Danilo Ardagna, et al., "Adaptive service composition in flexible processes," *IEEE Transaction on Software Engineering*, 33(6), 2007, pp. 369-383.

[10] Michael Mrissa, et al., "A context-based mediation approach to compose semantic Web Services," *ACM Transactions on Internet Technology*, 8(1):, 2007, pp. 41-43.

Bo Jiang was born in 1970. She received the PhD degree in Computer Science from Zhejiang University, China, in 2007.

She stayed in Ubiquitous Computing Research Laboratory (UCRL) Zhejiang University as a postdoctoral researcher since 2008. She is also an associate professor in College of Computer and Information Engineering, Zhejiang Gongshang University, P.R.China. Her current research interests include Electronic Commerce, CSCW, artificial intelligence and ubiquitous computing.

Bo Jiang is a member of IEEE.

Yun Ling was born in 1962. He received the Master degree in Computer Science from Zhejiang University, China.

He is a professor in College of Computer and information Engineering, Zhejiang Gongshang University, P.R.China. His current research interests include Electronic Commerce, Wireless Sensor Network.