Modeling Enterprise Intelligence Component Based on Multi-Agents

Rui Fan and Lingxi Peng

School of Software, Guangdong Ocean University, Zhanjiang 524088, P.R. China
fanrui@gdou.edu.cn scu.peng@gmail.com

Abstract. The large-granularity software component is the basis for structuring complex enterprise software system. However, current components are small, and their coupling is close. Furthermore, a software entity is broken up and distributed in tiers, and different entity pieces in same level are interweaved. All these limitations lead to unclear component boundary, complicated internal structure and inflexible interaction, which increase difficulties for component to be updated, replaced and maintained. This paper proposes a Multi-agent (M-Agent) component formal method, which encapsulates the enterprise subject domain in the enterprise entity component, encapsulates the enterprise subject process in the intelligent connector, and dynamically assembles them in P2P nodes. The theoretical analysis proves that the proposed method will change enterprise intelligence component to a large granularity, loose coupling and independent evolvement grid component model.

Keywords: *Enterprise intelligence component, Enterprise entity component, Intelligence connector, P2P*

1. INTRODUCTION

For adapting dynamic changes of complex enterprise information system, several solutions are already presented. The virtual enterprise whole world supply chain organization net is proposed, which the modeling concept, the method as well as reference architecture is given [1].An M-Agent ERP prototype system is proposed, which completes the widespread enterprise integration. Agents can be assembled fast, which deals with the change demands of enterprise [2]. The dynamic organization network's software component architecture is presented. A engine component runs above the ERP, which provides the essential intelligence and flexible for the dynamic supply chain integration [3].The active services is proposed, which use program mining frame on Internet, dynamically retrieves related components and assembles software system, which satisfies user need [4].

The dynamic changes demand that complex software system can evolve independently. The great granularity component may be the key. Currently, granularity of component is small, their coupling is close, and it is broken up and distributed in tiers, in same level different entity pieces are mixed together. That lead component boundary is unclear, internal structure is complicated, and interaction is inflexibility, increase difficulties for component to renew, replace and maintain. With

Please use the following format when citing this chapter:

Fan, R., Peng, L., 2007, in IFIP International Federation for Information Processing, Volume 254, Research and Practical Issues of Enterprise Information Systems II Volume 1, eds. L. Xu, Tjoa A., Chaudhry S. (Boston: Springer), pp. 455-460.

the M-Agent component formalization method, encapsulates the enterprise subject domain into enterprise entity component, encapsulates the enterprise subject process into the intelligent connector, and dynamically assembles them in P2P nodes. Let the enterprise intelligence component become into a large granularity, loose coupling, and independent evolvement grid component model.

2. ENTERPRISE MODEL TO M-AGENT COMPONENTS

By customizing enterprise subject domain and process with M-agent components, the enterprise model is transformed into the enterprise intelligent component model.

2.1 The Enterprise Entity Component

Define 1 The Enterprise entity component, which is shown in Figure 1

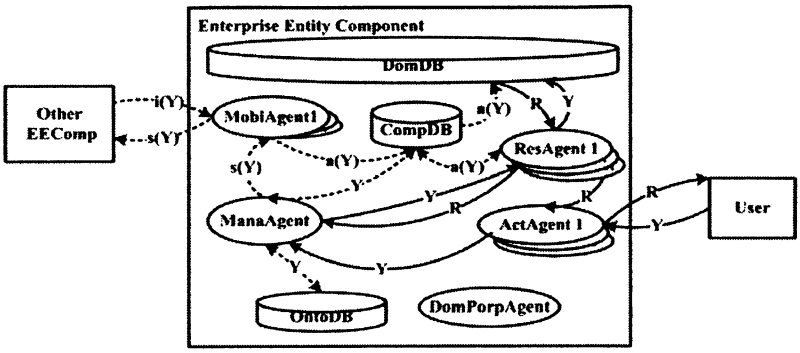

Figure 1. The Enterprise Entity Component Model

$$EEComp ::= (ID, DomDB, OntoDB, CompDB, ManaAgent, MobiAgentSet, ResAgentSet, ActAgentSet, EC) \quad (1)$$

Thereinto:
ID is an only identifier;
DomDB is a subject domain database;
OntoDB is an ontology knowledge library;
CompDB is a base components library;
ManaAgent is a management agent, which is control and administrative center;
MobiAgentSet is mobile agent set. MobiAgent plays a role of dynamical interface, which goes to goal nodes for carries out servers and returns results from nodes;

ResAgentSet is resources agent set. ResAgent carries on the inquiry, renewal to the subject domain database as well as other internal resources;

ActAgentSet is participant agent set. ActAgent is interactive windows for user;

DomPropAgent is an attribute value agent, which is a window for viewing and customizing attributes of EEComp.

EC is the π-calculus description about evolution of EEComp. In figure 1, the evolution is described as follows:

$\overline{\text{User}}\,Y.\text{ActAgent}(Y).\overline{\text{ActAgent}}\,Y.\text{ManaAgen t}(Y).([Y \in \text{RegTable}].\ \text{ResAgent}(Y).\overline{\text{ResAgent}}\,Y.$

$\text{DomDB}(Y).\ \overline{\text{DomDB}}\,R.\text{ResAgent}(R).\overline{\text{ResAgent}}\,R.\text{ActAgent}(R).\overline{\text{ActAgent}}\,R.\text{User}(R)\ +$

$[Y \notin \text{RegTable}].([Y \in \text{OntoDB}].([\ Y \in \text{CompDB}].(R\ \text{esAgent}(a(\ Y))\ |\ \text{DomDB}(a(Y))).$

$\overline{\text{DomDB}}\,R.\text{ResAgent}(R).\overline{\text{ResAgent}}\,R.\text{ActAgent}(R).\overline{\text{ActAgen}}\,t(R).\text{User}(\ R)\ +$

$[Y \notin \text{CompDB}].\text{Mo biAgent}(s(\ Y)).\overline{\text{MobiAgent}}\,s(Y).\text{Other}\ (s(Y)).\overline{\text{Other}}\,l(Y).\text{MobiA gent}(l(Y))\ .$

$\overline{\text{MobiAgent}}\,a(Y).\text{CompD B}(a(Y)).(R\ \text{esAgent}(a(\ Y))\ |\ \text{DomDB}(a(Y))).\overline{\text{DomDB}}\,R.\text{ResAgent}\ (R).$

$\overline{\text{ResAgent}}\,R.\text{ActAgent}\ (R).\overline{\text{ActAgen}}\,tR.\text{User}(R)\ )$

$+\,[Y \notin \text{OntoDB}].\text{Mo biAgent}(s(\ Y)).\overline{\text{MobiAgent}}\ (s(Y)).\text{Oth er}(s(Y))))$

When request Y is sent to ManaAgent via ActAgent, ManaAgent inquires the RegTable for confirming relative services, activates related agents and resources to carry out the corresponding services, and returns the result R. If doesn't have related services, a compare is made with the OntoDB, if Y belongs to this subject, ManaAgent demands related agents to add base components from the CompDB for new functions. If doesn't have need components in CompDB, ManaAgent dispatches MobiAgent to search outside, download and load need components to agents and resources for evolution. If Y doesn't belong to this subject, ManaAgent dispatches MobiAgent to interact with other entity components for request Y.

2.2 The Intelligent Connector

Define 2 The Intelligent connector, is shown in Figure 2

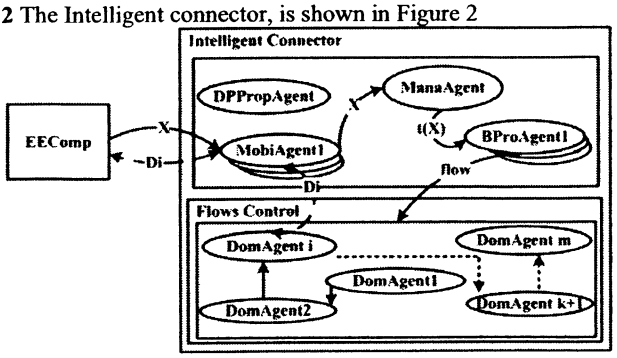

Figure 2. The Intelligent Connector Model

$$IConn ::= (ID, ManaAgent, MobiAgentSet, DomAgentSet, BProAgentSet, EI) \quad (2)$$

Here:
ID is an only identifier;
ManaAgent is a management agent;
MobiAgentSet is mobile agent set;
DomAgentSet is domain agent set. DomAgent is a proxy of EEComp;
BProAgentSet is subject process agent set. BProAgent dynamically rebuilds and controls business chain;
PPropAgent is an attribute value agent, which is a window for viewing and customizing attributes of IConn.
EI is the π-calculus description about evolution of IConn. In figure 2, the dynamic intelligent evolution mechanism is described as follows:

$$\overline{EEComp}X.MobiAgent(X).\overline{MobiAgent}X.ManaAgent(X).\overline{ManaAgent}t(X$$
$$\overline{BProAgentflow}.(\overline{DomAgent}ID1.MobiAgent1(D1).\overline{MobiAgent}ID1.EEC$$
$$\overline{DomAgent2}D2.MobiAgent2(D2).\overline{MobiAgent2}D2.EEComp2(D2)|...|$$
$$\overline{DomAgentm}Dm.MobiAgentm(Dm).\overline{MobiAgentm}Dm.EECompm(Dm))$$

When X send from EEComp, MobiAgent transmits X to ManaAgent. ManaAgent outputs tasks t(X) to BProAgent. With the contract net, BProAgent classes tasks and sends to each DomAgent, which has registered. According to own ability, DomAgent gives up or returns the bid. BProAgent evaluates, and puts selected DomAgents into the flows control model; the dynamical evolution of structure is achieved by rebuilding business chains. Then, BProAgent controls these DomAgents to interact with related enterprise entity components, cooperatively finish related services.

2.3 The Enterprise Intelligent Component

Define 3 The Enterprise intelligent component is shown in Figure 3.

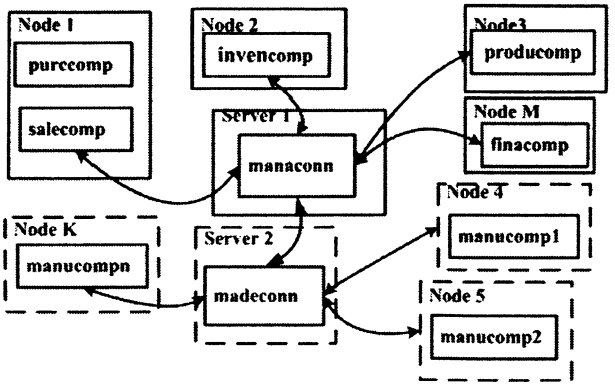

Figure 3. The Enterprise Intelligent Component Model

$$EIComp ::= (EEComp, IConn, EE) \quad (3)$$

Thereinto:

EEComp is the enterprise entity component;

IConn is the intelligent connector;

EE is the π-calculus description about evolution of EIComp;

Figure 3 shows the deployment of EIComp in P2P nodes, the formal description is as follows:

```
IConn : ManaConn, MadeConn; / * management ,..., manufacture intelligent connector * /
EEComp : SaleComp, InvenComp, PurcComp, ..., ManuCompn; / * sale, inventory,
                                                    purchase, ..., makesn entity component * /
Conf(node1 ,(PurcComp, SaleComp)) ; / * purcComp & saleComp in node1 * /
Conf(node2 , InvenComp) ; / * invenComp in node2 * /
...
Conf(nodek , MauCompn); / * mauComp n in nodek * /
Conf(Server1, ManaConn); / * manaConn in severl node * /
Conf(Server2, MadeConn); / * MadeConn in server2 node * /
...
manaconn_flow(salecomp, invencomp, producomp, finacomp);
madeconn_flow(manucomp1, manucomp2, ..., manucompn) .
...
```

The Conf ( ) describes the dynamic deployment about node with EEComp or IConn, and the IConn_flow ( ) describes the dynamic rebuilding business chain, which include those EEComps.

3. CONCLUSIONS

In order to construct the great granularity autonomy software component, this paper propose the M-Agent component formal method, it encapsulates the enterprise subject domain into reusable, autonomous enterprise entity component. It also encapsulates the enterprise subject process into the dynamic intelligent connector. Furthermore, it dynamically assembles them in the P2P nodes. Finally, the method makes the enterprise intelligence component become one kind of big granularity, clear boundary, loose coupling, and independent evolution grid component model.

REFERENCES

1. A. Zaidat, X. Boucher, and L. Vincent: A framework for organization network engineering and integration, *Robotics and Computer-Integrated Manufacturing.* Volume 21, pp.259-271, (2005).
2. B.R. Lea, C. Mahesh, W. Gupta, and B. Yu, A prototype multi-agent ERP system: an integrated architecture and a conceptual, *Technovation.* Volume 25, pp.433-441. (2005).
3. M. Verwijmeren: Software component architecture in supply chain management, *Computers in Industry.* Volume 53, pp.165-178, (2004).
4. Y. Zhang and C. Fang, *Active Services: Concepts, Architecture and Implementation* (Science publish house: Beijing, 2005).