Intelligent data-collaboration mechanism under the distributed application environment

Jian-Long Ding[1*], Weifang Chen[1], Ji Gao[2]

[1]*Zhejiang Shuren University, College of Information Science, Hangzhou, China*

[2]*Zhejiang University, The Artificial Intelligence Research Institute, Hangzhou, China*

Received 1 March 2014, www.tsi.lv

Abstract

Because of the complexity, the dynamic and uncertainty of the distributed applications environment, the data-collaboration crisis caused by isolated information island is serious day by day. Through the establishment of Data Cooperation-based Virtual Organization (DCVO), is conducive to meet the realistic demand of the on-demand dynamic data collaboration, which led to the distributed application to carry out the intelligent data collaboration in effective control, as well as realize the intelligent data retrieval across application domain. Through research of the Distributed Application System-based Data Cooperation Architecture (DASDA) , to straighten out the related technology and method of distributed collaborative, from the semantic specification (including the application of domain ontology, relational databases and ontology mapping mechanism, cooperative data transmission standard), rational of data-collaboration (policy representation and configuration), collaborative service personalization, data structure and model of collaborative content level and so on, to provides an important reference to solve the eliminate problem such as semantic fuzzy, dynamic expansion, uncontrollable, cooperative security, recall and precision of conflict which caused in the process of the data-collaboration.

Keywords: ontology, virtual organization, data collaboration, policy configuration

1 Introduction

With the development of information construction, the information system for kinds of application field become perfect gradually, except the application system under distributed environment witch face to information island and information gap problem. Service collaboration and data exchange requirements between application systems be increasingly urgent [1]. Because of the complexity, dynamic and uncertainty of the distributed environment, the distributed software system for specific application domains need to configure data-collaboration ability, in order to ensure that the system can be reliable, stable, accurate, and to provide cross-system data collaboration service for end users [2]. But because of the complexity of the distributed system and its lack of coordination ability, make the data-collaboration research difficult without the theoretical model support, and can't meet data cooperative crisis caused by growing information islands. In recent years, as the rising and developing of the cloud computing, data research and service architecture (SOA), promoting changing of distribution system development method, from the static sealing process of high cost, low efficiency, depends on the specific hardware environment to on-demand combined virtual business service process of dynamic, fast, low cost [3-6]. These changes promote the development of Data Cooperation-based Virtual Organization (DCVO) [7-10].

On this basis, to carry out the study of the Distributed Application System-based Data Cooperation Architecture (DASDA) research [11, 12], in order to eliminate the data-collaboration dilemma, which caused by the closure between business systems. Thus, to make sure the original distributed application have strong adaptability to face the all kinds of data-collaboration problem caused by non-deterministic data centre distributed operating.

2 Data collaborative DCAgent model

In order to improve the intelligent characteristics of each application point, One method is providing intelligent Agent called Data-Cooperation Agent (DCAgent) corresponding for each application point, which can improve the coordination ability. As the main body of DCVO, DCAgent's structure can be divided into two levels: application domain development layer and Agent engine) running layer as shown in Figure 1.

Development layer can provide the humanized man-machine interface to assist the developers to build the Agent model and application domain ontology, relational data and domain ontology mapping configuration file and external call component as collaborative service components which can provide basic services as personalized data synchronization, the domain ontology ConceptInstance dynamic construction etc. It can also support application developers to design all kinds of

[*] ***Corresponding author's*** e-mail: zjuding@163.com

DCAgents for different purposes, to meet the needs of organizations and Individual user. Running Layer provides Agent engine, which achieve the following functions.

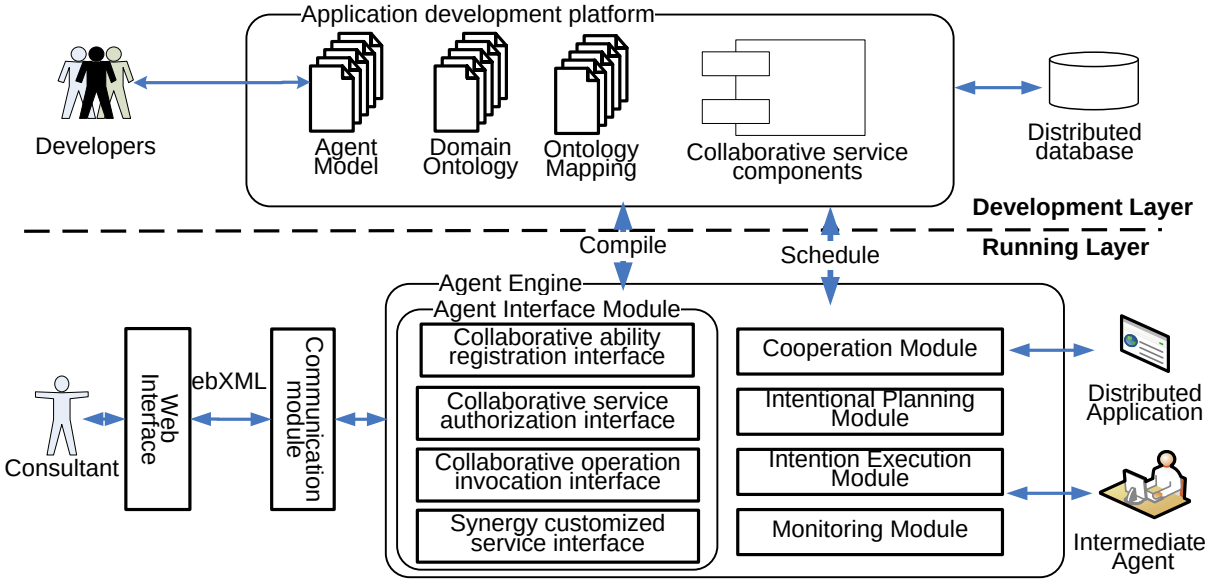

FIGURE 1 DCAgent Model diagram

2.1 PROVIDES THE INTERFACE OF AGENT SERVICE

This service is to support the interaction between Agent and human, mainly includes four aspects: collaborative ability registration interface, collaborative service authorization interface, cooperative operation interface, Synergy customized service interface. The four interface support human to manage and control Agent effectively.

Collaborative ability registration interface, this interface response to submit the concept set of collaborative field data to the intermediary DCAgent based on the domain ontology, and set the scope of cooperative operation authorization which can opening to the other DCAgents.

Collaborative service authorization interface mainly provides such functions as user management, role management, collaborative policy, configuration, etc. So as to effectively ensure the safety and reliability of the DCAgent-service calling.

Cooperative operation interface is mainly responsible for the user to invoke the DCAgent service execution process, including DCAgent service release, and authority audit, call, execution, tracking process at the levels of DCAgent, service, and operation. It can make the Agent convenient, reliable, and controllable.

Synergy customized service interface mainly provides the policy custom interface for personalized data-query, to realize the semantic level queries across distributed application by policy customization, which is used for the constraint and customization to query data.

2.2 PROVIDES THE UNITED SCHEDULING MECHANISM

The United Scheduling Mechanism consists of Cooperation Module, Intentional Planning Module, Intention Execution Module and Monitoring Module. According to the DCAgent model effectively develop and coordinate social activities, including the intermediary service request. To Establish and optimizes the cooperation relationship through rational negotiation, as well as properly handling exceptions occurred while processing of cooperation, so that, their behaviour comply with collaborative authorization policy constraints, and regulate it's the collaborative by customized control policy [13].

3 DCVO architecture under the distribute application environment

Distributed application system includes database and file system, application, in order to make the distribution of the application nodes with intelligent collaboration, need to develop and deploy DCAgent for each application node. The DCAgent will improve the data-coordination ability of application point. At the same time, in order to improve the data-coordination's performance, stability and failure recovery ability, also need to deploy high performance database (Crash level) for each application node [14]. An intermediary DCAgent should also be Set up to realize the data collaboration, data-collaboration rights, which include synchronization-role auditing, synchronization-authorization validating, synchronization-data cache, synchronous data assembly and forwarding, and synchronization commands issue and

execution. Thus the distribution of application nodes will Dynamic combine into a virtual organization called DCVO according to the specific needs of collaboration-data [15].

4 Data collaboration execution process under the distributed environment

Data collaborative virtual organization DCVO is not a static organization, but a dynamic forming organization according to data collaborative demand. Therefore the distributed application data collaborative process started by application point which data changing occurred. The triggering process is divided into two levels, the database level and process level. For the existing island type of distributed application system for data collaboration capability upgrading situation, due to the complexity of existing application system, can consider to trigger collaborative process by adding triggers in the database level. For the new distributed application system development situation, can consider to trigger collaborative process at the process level through the assembly calling way [16]. The initiator DCAgent must submit data-collaboration Role authorization application to the intermediary DCAgent in order to become the main organizer and then launched a series of coordinated operation. The specific process is shown in Figure 2.

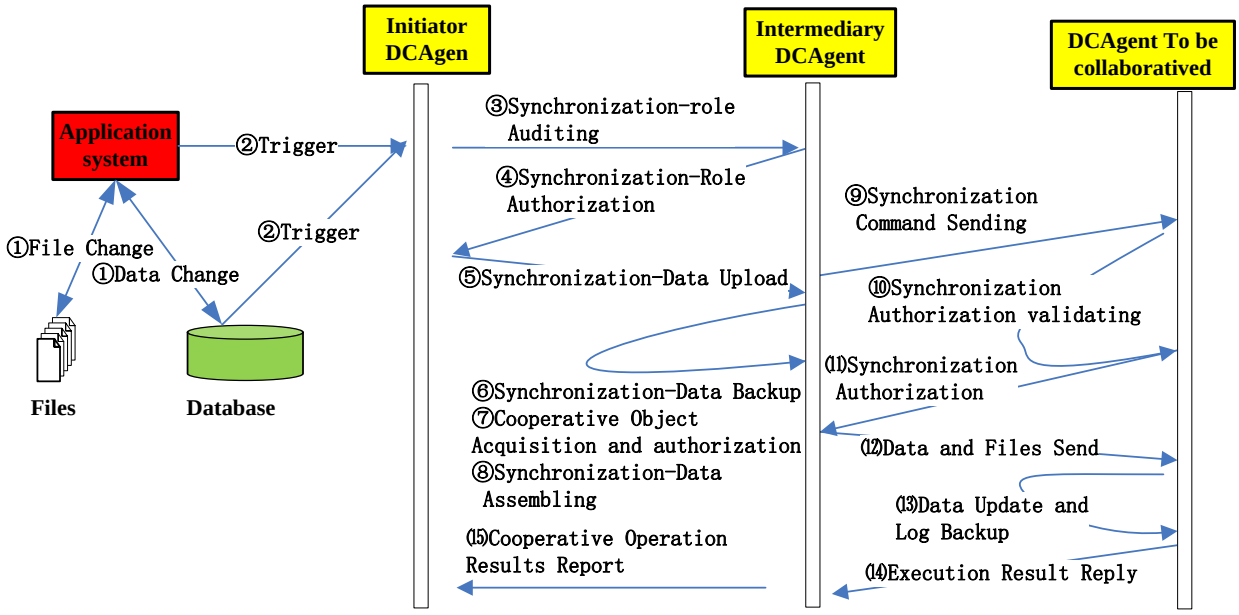

FIGURE 2 Data collaboration process description

5 Collaborative Dasda system based on distributed application data

The data collaboration under distributed application environment is a complex problem, the study of the DASDA method system will help straighten out the related technology and methods of distributed collaborative.

DASDA is defined as the following 5 elements: **DASDA = (CS, IS, RP, CI, DP).**

CS—Clear Semantics. By the research on sharing information modelling and ontology-based representation mechanism, and application domain relation database and domain ontology mapping mechanism, and data transmission format specification, etc., the DCVO will have clear semantics when data storage and exchanging.

IS—Intermediary Services. Through the establishment of DCVO oriented collaborative system of intermediary services, so that DCVO can accurate and convenient to obtain the relevant data changes to the application system at anytime, anywhere. And provides collaborative authorization registration, role application, and semantic information retrieval service cross distributed systems.

RP—Rational Process. To establish data-coordination specification set, So that, DCVO can Carry out authorization policy description and allocation, cooperative agreement, rollback model, data retrieval policy generation operation, rational and flexible propulsion data cooperative process, to achieve the Intelligent Collaborative higher.

CI—Cooperation Individuation. Book personalized service for each DCVO through data coordination authorization policy customization. Provide personalized data query service cross distributed application point by service-customization interface, man-machine interface generation component and Data query policy.

DP—Development Plat for DCVO. Through the application of domain ontology editor, local cooperative service component development guide, configuration files auxiliary generator, VO model assisted editing platform to provide convenient, transparent development assistance service for DCVO to the user.

5.1 CLEAR SEMANTICS CS

As the intelligent agent and executive body of application domain, DCAgent need experts to define the Agent service and the implementation process by constructing a DCAgent model, but the actual operation is realized by the cooperative service components which can be scheduled by DCAgent [17]. In order to realize the intelligent data - collaboration and semantic level data-retrieval, works such as completing semantic description transformation and building corresponding mapping mechanism for application domain by ontology technology must be finished. All those works is depend on domain ontology and ontology mapping file. To make the modeling information semantic content clear, CS can describe as following 3 elements.

CS = (DCL, OML, Mapping):

Mapping: DCL↣Realation DataBase;

the "↣" is a the symbol in Z language, indicating the injective function.

DCL—As knowledge representation language based-on ontology, which using in the Agent.

OML—A markup language based on Ontology, also be a communication language between Agent.

Mapping—An one-to-one mapping mechanism between DCL language and relation database.

5.1.1 Domain ontology description specification

Because of the variety of distributed application nodes on the data describing structure, property, storage way, and etc. An unified concept description standard for application domain must be established by ontology description language. As a domain concept language, DCL can be used to represent the domain ontology by the form of concept-relationship-attribute and constraint rules [17]. Its BNF form descript as follows:

Ontology <Ontology-Name> [<Version-declaration>]
[<Ontology-Citation>][<Synonymous-Concepts>]
[<Synonymous-Properties>] [<Property-Definitions>]
[<Concept-Definitions>] [<Type-Definitions>]
<Concept-Definitions> := {Concept <Concept-Name>
[Super: {<Super Class Name>}+]
{<Slot-Name>：{<Aspect-Name> <Aspect-Content>,};}**
*[Constraint：<Condition-Expression>] }**
<Aspect-Name> := val |type |mode |number |derive |restriction |unit |inverse |superSlot

Ontology is consist of the set of concepts, the super slot is used to establish the relationships between concepts when slot value is a single super class name. Slot is used to define the attributes or the relation and function parameters of object. Each slot include Type and Mode side, which respectively indicating the type of concept instance slot values (types can be another concept) and the provide way. Constraints express then relations between different slot.

5.1.2 Collaboration-data Transmission of language OML

Each DCAgent for application node need for information and data transfer during collaborative processing, which can realized by the high performance distributed middleware platform called ICE (Internet Communications Engine). Ontology Based Markup Language (OML) is designed for constrained ebXML, contain the DCL representation. Collaborative command and collaborative data is the main content of the communication, the bottom element is composed of concept instances, and therefore we need to use the ebXML to extend ICE to Expand ICE called E-ICE. Original message will by packaged into **ebXMLMessage** by **E-ICE**. It is defined as follows:

Concept ebXMLMessage
MessageType: type ebXMLMessageType mode necessary;
PartyID: type string;
Service: type string;
Action: type string;
ConversationId: type string;
CooperationRole: type CooperationRoleType mode necessary;
*MessagePayload: type*string;*
End ebXMLMessage

The message will be serialized and then be send through ICE. and then be deserialized after be received by the DCAgent terminal.

5.1.3 Mapping mechanism between database and domain ontology

In order to shield the structure and scope difference between data-synchronization nodes. the data in relational database must be processed and packed into concept instance mode before synchronization. Data-ConceptMapping configuration file will help DCAgent to identify the transformation relationship automatically [18].

Mapping: DCL↣Realation DataBase.
*<Mapping> ::={< ConceptMap >}**
<ConceptMap> ::=({Table|view}+,<Concept>) |< SlotMaps >
*<SlotMaps > ::={<SlotMap>}**
< SlotMap > ::=(<Field>,<Slot>)|<weight>

Mapping defines the relationship between the relational database and the application domain concepts. One <**ConceptMap**> node represents one mapping relationship, <Table> indicates data table be mapped <**Concept**> indicates the corresponding concept if the mapping object relates to multiple data tables, needs to establish a view first, then establishes the mapping relationship with the corresponding Concept. One <**SlotMap**> node indicates relationship between the data fields and slot, the weight property indicates the matching weight.

5.2 RATIONAL COLLABORATIVE PROCESS RP

There is a big difference in the operate scope and mode during the data-collaborative process, the operation authority and data range of the Data-synchronization must be declared for each application node. We can solve the above problems through policies, and using PolicyAssignment to specify the scope of authorization. So that the DCAgent's operation can be executed under the premise and controllable situation.

5.2.1 Policy Representation and Assignment Language PRAL

PRAL (Policy Representation and Assignment Language) is designed to be easy for users to understand and use, has broad application demand coverage of common language, to support the policy of declarative representation and configuration description. PRAL provides a structured and object oriented representation, is used to define the configure of policy. The first-order logic based on concept instance is used to represent the policy definition and configuration details. The PRAL entity class (including the characteristics and relationship) and complexity (with multiple characteristics) are defined as "concept".

5.2.2 Policy definition

Representation structure of the Policy is defined as a policy embedded in PRAL.

```
Type PolicyType: base_type string, restriction enumeration
(Select, Insert, Update, Delete,RollBake,Bake,Restore);
Concept  Policy
    Name:  type string  mode necessary;
    PolicyType:  type PolicyType  mode necessary;
    Processing:  type ProcessingType;
    Target:  type condition;
    Update:  type date mode necessary;
End  Policy
```

PolicyType can be divided into 7 categories: query, insert, modify, delete, rollback, backup, recovery. Target refers to the target range using conditional expression. Corresponding to data synchronization service that DCAgent provide.

5.2.3 Policy Assignment

Policy definition and configuration separately, facilitates the reuse policy, policy allocation and revocation of convenience. The policy configuration with PRAL embedded Policy Assignment concept definition independent representation of Policy definition and Policy assignment is help to promote the performance of policy on reuse and convenient The policy Assignment defined by with embedded **PolicyAssignment** concept in PRAL.

```
Type ModalityType: base_type string, restriction
enumeration (+, −);
Concept  PolicyAssignment
    Policy:  type string  mode necessary;
    Modality:  type ModalityType  val "+";
  Subject :type *RoleAssignment mode necessary;
    Delegator:  type string;
    TemporalConstraint:  type TemporalLogic;
End  PolicyAssignment
```

5.3 DEVELOPMENT PLAT FOR DCVO

5.3.1 Research on the DCVO development platform

At present, the unit to which I have developed the application domain and modelling platform (ARMF), DCAgent development platform (RADF). In order to make the develop and deploy of DCVO. Simple, we also need to design following tool, the way of the research on mapping file (for relational database and Domain ontology) generation and verification tool, PRAL language develop tool.

5.3.2 Research on Data-synchronization component

The development of data system components mainly involves two key technologies: high performance distributed middleware ICE and high performance of real-time database Berkeley DB. Use high performance distributed middleware platform ICE (Internet Communications Engine) to realize files and database synchronization [18]. Use Berkeley DB to realize the data cache in all the data synchronization node [19].

6 Conclusion

With the continuous deepening of information technology, the way of system development from Independent development into system integration so as to provide experience to solve problems occurs during data-synchronization such as semantic fuzziness, uncontrollable, security, synchronous efficiency. This study is mainly face to the common problem of distributed application system, to enhance the utility of the mechanism, need to further study on the specific application environment, especially how to improve the automation ability configuration after a the new synchronization node is increased.

Acknowledgements

Foundation items: 1. The Education Department of Zhejiang Province, general scientific research project (Y201018715). 2 The National Natural Science Fund Project (61375071). 3 The priority theme emphases project of Zhejiang province, China (Grant 2010C11045).

References

[1] Li X 2003 The role of digital resources integration based on the information island *Library Forum* **6**(23) 121-2
[2] Beners-Lee T, Hall W, Hendler J, Shadbolt N, Weitzner D J 2006 Creating a science of the web *Computer Science* **313**(5788) 769-71
[3] Chen Q 2009 Cloud computing and its key techniques *Journal of Computer Application* **9**(26) 2562-7
[4] Papazoglou M P, Traverso P, Dustdar S, Leymann F 2007 *IEEE Computer* **40**(11) 38-45
[5] Mike P, Papazoglou M P, van den Heuvel W J 2007 Service oriented architectures: approaches, technologies and research issues *The VLDB Journal* **16** 389–415
[6] Demirkan H, Kauffman R J, Vayghan J A, Fill H-G, Karagiannis D, Maglio P P 2008 Service-oriented technology and management: Perspectives on research and practice for the coming decade *Electronic Commerce Research and Applications* **7** 356–76
[7] Wei I, Blake M B 2010 Service-Oriented Computing and Cloud Computing Challenges and Opportunities *IEEE Internet Computing* **14**(6) 72-5
[8] Lv J, Ma X 2006 Research and development of Internetware, Science in China (Series E) **36**(10) 1037-80
[9] Jones J, Parnin C, Sinharoy A, Rugaber S, Goel A K 2009 Teleological Software Adaptation *The 3rd IEEE International Conference on Self-Adaptive and Self-Organizing Systems* 198–205
[10] Paulo L, Paul V, Emmanuel A, 2009 Self-Adaptation for Robustness and Cooperation in Holonic Multi-Agent Systems *In Hameurlain Transaction on Large-Scale Data- & Knowl-Cent Sys I LNCS 5740* 267–88
[11] Ghosh D, Sharman R, Rao H R, Upadhyaya S 2007 Self-healing systems-survey and synthesis *Decision Support Systems* **42** 2164–85
[12] Gao J, Yuan C-X, Wang J 2005 SASA5: A Method System for Supporting Agent Social Activities, Chinese Journal of computers **5**(28) 838-48 (*in Chinese)*
[13] Hu J, Gao J ,Zhou B, Liao B-S, Chen J-J 2004 Ontology based agent services compatible matchmaking mechanism *Proceedings of the 3rd International Conference on Machine Learning and Cybernetics* Shanghai 111-6
[14] Dou D, Qin H, LePendu P 2010 Ontograte: Towards automatic integration for relational databases and the Semantic Web through an ontology-based framework *International Journal of Semantic Computing* **4**(1) 123-51
[15] Weyns D, Georgeff M 2010 *IEEE Software* **27**(1) 86-91
[16] Aly W H F, Lutfiyya H 2007 Dynamic adaptation of policies in data center management *Proceedings of the 8th IEEE International Workshop on Policies for Distributed Systems and Networks* 2007 266-72
[17] Calvanese D, De Giacomo G, Lembo D, Lenzerini M, Rosati R, Ruzzi M 2009 Using OWL in data integration in R. De Virgilio, F. Giunchiglia, L. Tanca ed. *Semantic Web Information Management -- a Model Based Perspective* Chapter 17 Springer 397-424
[18] Xu S, Johnson A 2009 Comparing the Performance of EPICS Channel Access with a New Implementation Based on ICE (the Internet Communications Engine) *the 16th IEEE – NPSS Real Time Conference (RT 2009)* 2009 113-6
[19] Langley N 2008 Embedded database Berkeley DB offers speed and management gains *Computer Weekly* 2008 34

Authors

Jianlong Ding, born on September 15, 1980, Hangzhou, China

Current position, grades: lecturer of computer science and application in Zhejiang Shuren University.
University studies: Master's degree in the computer science and application from Zhejiang University.
Scientific interest: intelligent software, distributed computing.
Publications: 6 patents, 7 papers.

Weifang Chen, born on June 22, 1966, Ningbo, Zhejiang, China

Current position, grades: lecturer of computer science and application in Zhejiang Shuren University.
University studies: electronic university of science and technology of Hangzhou.
Scientific interest: artificial intelligence, network computing.
Publications: 1 patents, 4 papers.
Experience: teaching and research work, in recent years.

Ji Gao, born in August, 1946, Shangdong, China

Current position, grades: professor at the department of computer science and engineering college of computer science and technology of Zhejiang University, doctoral tutor.
Scientific interest: network computing and pervasive computing, intelligent software, agent technology, software engineering, middleware technology.
Publications: 2pPatents, 131 papers.
Experience: dean at the school of information science and technology in Zhejiang Shuren university in 2000 -2008. Institute Director of Internet Computing and Intelligent Systems Research Institute.