

Dickson Lukose
Zhongzhi Shi (Eds.)

LNAI 4078

Multi-Agent Systems for Society

8th Pacific Rim International Workshop
on Multi-Agents, PRIMA 2005
Kuala Lumpur, Malaysia, September 2005
Revised Selected Papers


 Springer

Table of Contents

Keynote Papers

Application of Information Processing Technology by Multiagent Simulation (Extended Abstract)	1
<i>Hideyuki Nakashima</i>	
Cognitive Architectures and Multi-agent Social Simulation	7
<i>Ron Sun</i>	

Full Papers

A Co-operative Intelligent Assisting Agent Architecture for Web Searching and Desktop Management.....	22
<i>Nandan Chaturbhuj, V. Shyam Prakash, and Shivashankar B. Nair</i>	
Prisoner's Dilemma Game on Network	33
<i>Masahiro Ono and Mitsuru Ishizuka</i>	
Addressing Constraint Failures in Agent Interaction Protocol	45
<i>Fadzil Hassan, Dave Robertson, and Chris Walton</i>	
Reasoning about Success and Failure in Intentional Agents	60
<i>Timothy William Cleaver, Abdul Sattar, and Kewen Wang</i>	
Agent-Base Workflow Management System Architecture – An Example on Conference Info System	73
<i>Feng-Chao Yang</i>	
Diverse Applications of an Automated Argument System Based on the Logic of Multiple-Valued Argumentation	88
<i>Gowun Jeong and Hajime Sawamura</i>	
Dynamic Interaction Protocol Load in Multi-Agent System Collaboration.....	103
<i>Maoguang Wang, Zhongzhi Shi, and Wenpin Jiao</i>	
Towards a Component-Based Architecture for TeleAgents	114
<i>Ockmer L. Oosthuizen and Elizabeth M. Ehlers</i>	
Development of a Lightweight Middleware Technologies Supporting Mobile Agents	128
<i>Gu Su Kim, Jaegon Kim, Hyun-jin Cho, Won-taek Lim, and Young Ik Eom</i>	

The Roles of Active Perception in Intelligent Agent Systems	139
<i>Raymond So and Liz Sonenberg</i>	
Fairness in Cooperating Multi-agent Systems – Using Profit Sharing as an Example	153
<i>Ming-Chih Hsu and Von-Wun Soo</i>	
Risk Strategies and Risk Strategy Equilibrium in Agent Interactions Modeled as Normal Repeated 2 X 2 Risk Games	163
<i>Ka-man Lam and Ho-fung Leung</i>	
A Crowd Model for Emergency and Panic Situations: Structure, Characteristics and Behaviours	178
<i>Mohd Bazli Ab Karim, Abdullah Zawawi Talib, Ahmad Izani Ismail, and Bahari Belaton</i>	
Learning the Structure of Utility Graphs Used in Multi-issue Negotiation through Collaborative Filtering (Preliminary Version).....	192
<i>Valentin Robu and Han La Poutré</i>	
ARTISTE: Agent Organization Management System for Multi-Agent Systems	207
<i>Atsushi Terauchi, Osamu Akashi, Mitsuru Maruyama, Kensuke Fukuda, Toshiharu Sugawara, Toshio Hirotsu, and Satoshi Kurihara</i>	
Sensor Management Model Using Multi-agent in Small-Area Sensor Networks	222
<i>Keonsoo Lee and Minkoo Kim</i>	
Conflict Resolution Method for Multi-context Situation	231
<i>Keonsoo Lee and Minkoo Kim</i>	
Multi-auction Approach for Solving Task Allocation Problem	240
<i>Chi-Kong Chan and Ho-Fung Leung</i>	
Using Classification Learning in Companion Modeling	255
<i>Daisuke Torii, Francois Bousquet, Toru Ishida, Guy Trébuil, and Chirawat Vejjpas</i>	
Modeling a MultiAgent Mobile Robotics Test Bed Using a Biologically Inspired Artificial Immune System	270
<i>Tejbanta Singh Chingtham and Shivashankar B. Nair</i>	
Deploying Multi-Agents for Intelligent Aspect-Oriented Web Services ...	284
<i>Santokh Singh, John Hosking and John Grundy</i>	
Multi-Agent System Using G-XMDR for Data Synchronization in Pervasive Computing Environments	297
<i>Youn-Gyou Kook, R. Young-Chul Kim, and Young-Keun Choi</i>	

Multi-Agent System Using G-XMDR for Data Synchronization in Pervasive Computing Environments

Youn-Gyou Kook*, R. Young-Chul Kim, and Young-Keun Choi

School of Information Technology Engineering,
Korea University of Technology and Education
Cheonan, Chungnam, 330-708, Korea
ykkook@kut.ac.kr

Dept. of Computer & Information Comm., Hongik University
Jochiwon, 339-701, Korea
bob@hongik.ac.kr

Dept. of Computer Software, Kwangwoon University
Seoul, 139-701, Korea
ygchoi@kw.ac.kr

Abstract. This paper extends XMDR[19] into G-XMDR (Global-Extended Metadata Registry) to ensure the interoperability of data, and to describe multi agent system using G-XMDR. The proposal G-XMDR will be solved the problem of data heterogeneity and inconsistency in sharing and exchanging each local data in virtual pervasive computing environments. Data heterogeneity is generated by different definition or mismatched expression of the same meaning data. Therefore, we define and design G-XMDR based on XMDR(ISO/IEC 11179) specification to interoperate various each local data in pervasive computing. We also describe our synchronizing multi agent system, which uses G-XMDR to solve data heterogeneity for data interoperability through synchronizing data. It provides more efficient agent through reducing errors that happened on synchronizing requests and adding new pervasive computing environments with their own each local data.

1 Introduction

As the necessity of information sharing for interoperating distributed data in synchronized pervasive computing environments is increased, current researches for data synchronization are progressing widely. Data synchronization in pervasive computing environments should be considered two aspects for trustworthy data sharing [21]. First one is the aspect of system, which has many particular features, such as diverse types of platform, autonomous data manipulations, and network traffics. We solve these problems with multi agent mechanism to guarantee 1) the independence from system platform, 2) the autonomy of data manipulations that is inserted, updated and deleted, and 3) the effectiveness on network traffics [1,7,9].

Second is the data aspect of the system, which has the data heterogeneity and inconsistency on synchronizing data. The data heterogeneity is classified with the

* Corresponding author.

schema and the data value heterogeneity. The former usually occurs on which have different structures at the same information or different naming at the same structure. The latter occurs on mismatched expression at the same data [4,8,18,19].

Therefore, we propose a synchronizing mechanism with the multi agent system for the interoperation of the distributed data in pervasive computing environments. The proposed multi agent system uses G-XMDR(Global-Extended Metadata Registry) to resolve the data heterogeneity. G-XMDR is designed with 1) XML as web document standard and 2) XMDR as data integration standard. XMDR is suggested by ISO/IEC 11179 which standardizes and registers data elements, and describes a data elements registry [15]. Therefore we design to extend XMDR (called G-XMDR) for shared data. It is used by multi agent system for conversion from transmitted XML to a data of local system, vice versa.

The remainder of this paper is organized as follows. The section 2 describes the structure of G-XMDR, section 3 describes a synchronizing mechanism with multi agents system, and section 4 presents implementation of the system. Finally, we conclude in section 5.

2 Design of G-XMDR(Global Extended Metadata Registry)

ISO/IEC 11179 part 3 [15] mentions that a metadata model for data processing and electronic data interchange from the each XMDR that is we called G-XMDR heavily relies on accurate, reliable, controllable, and verifiable. It provides the introduction and standard for the data element's constructions which are shared by users or data processing systems on the distributed environments.

Figure 1 shows a conceptual view of G-XMDR which defines G-XMDR from local XMDR within a domain through designing Metadata Registry for interoperating data on distributed local domains.

We define G-XMDR for data sharing and exchanging through the designed L-XMDR(Local XMDR), which designs based on XMDR(ISO/IEC 11179-3) specification.

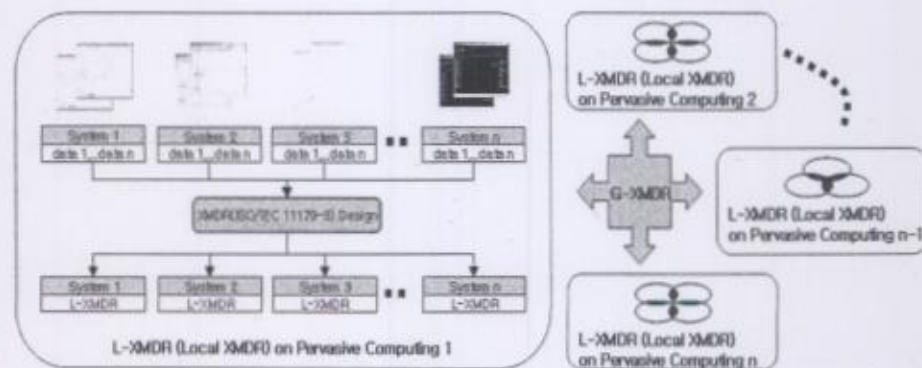


Fig. 1. The outline of G-XMDR on pervasive computing environments

The Metadata Registry for data processing and data interchange is shown in figure 1. Through analyzing and gathering each local XMDR from each local domain, we can make Metadata Registry which includes all local XMDRs of all local systems like figure 2.

Figure 2 shows a part of local XMDR from A whole L-XMDR designed for data sharing and exchange in a local domain. We will locate each local own XMDR for guaranteeing the independency and extensions of data.

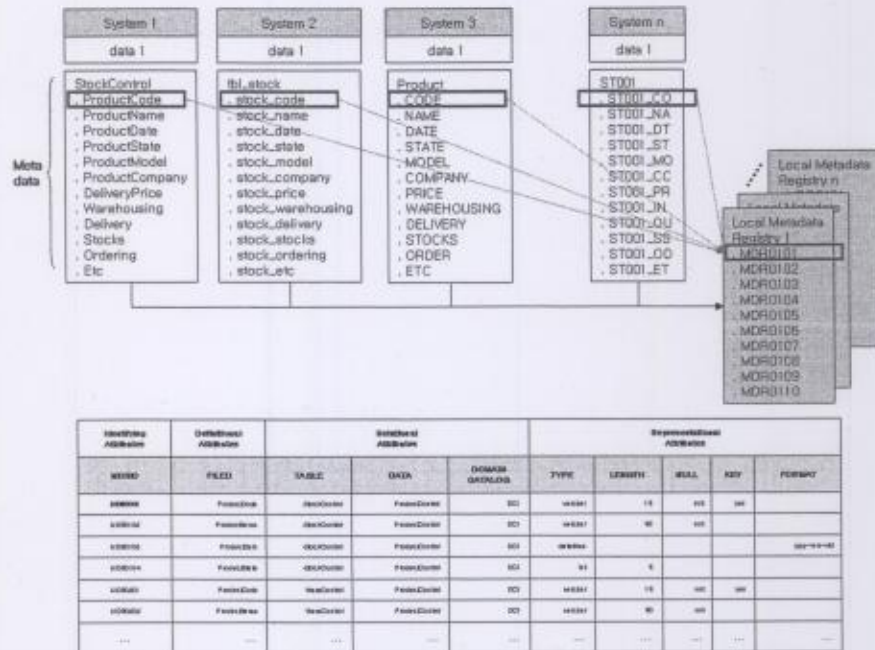


Fig. 2. Designing each local XMDR which is interoperated in each local domain for global domain

Figure 3 shows to extend local XMDRs for data synchronization on the environment of pervasive computing.

In XMDR, there are classified by 4 field areas, that is, four attribute's types, which contain their own attributes as follows:

- Identifying : attributes that are applicable for the identification of a data element (MDRID)
- Definitional : attributes that describe the semantic aspects of a data element (FIELD)
- Relational : attributes that have associations among data elements and associations between data elements and classification schemes, data element concepts, objects, entities. DOMAIN_CATALOG is a local domain of pervasive computing. (TABLE, DATA, DOMAIN_CATALOG)
- Representational : attributes that describe representational aspects of a data element (TYPE, LENGTH, NULL, KEY, FORMAT)

When the data was modified at the local system, the system requests the synchronization to the other systems. If the synchronization starts, data and query of local system will be converted to the synchronization XML document using G-XMDR mapping. After each local system receives the synchronization data, it converts XML documents to the local-acceptable data and query using its own G-XMDR mapping. This procedure and system will be described at figure 5.

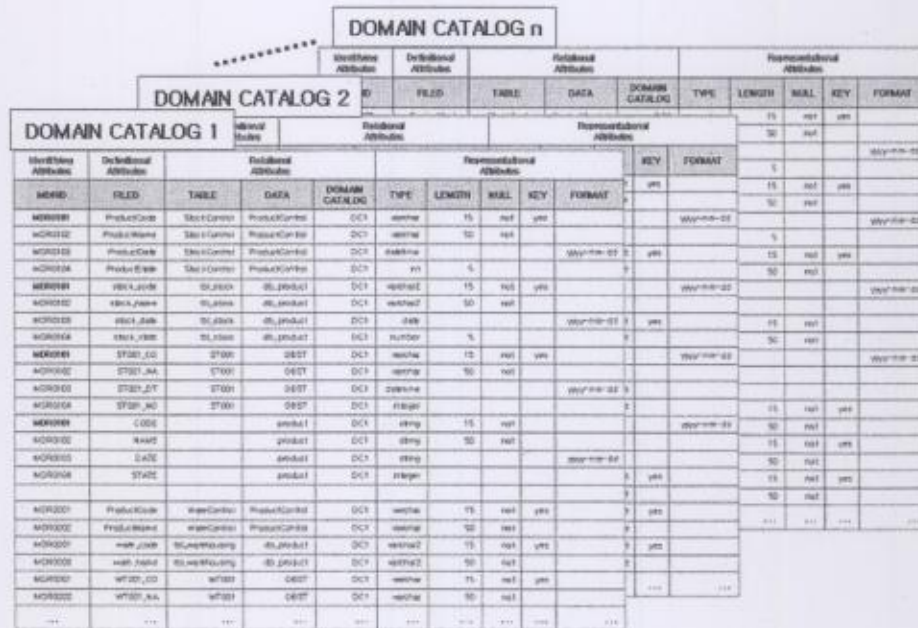


Fig. 3. Designing G-XMDR based on ISO/IEC 11179-3

After we will partition each local own XMDR from a whole G-XMDR in figure 3, send each local own XMDR to each local domain.

G-XMDR should be designed based on predefined XMDR, and it is necessary to make the standard for the transmission of local data from local domains during synchronization process. So, G-XMDR has the validation restriction through definition of document structure. G-XMDR's structure is standardized by defining XML schema with all local XMDR's composition information, so XMDR's modification does not affect the G-XMDR's structure, because the reliance of document and data exchange have to be guaranteed.

G-XMDR's schema defines the elements which are suitable to the synchronizing data structure. DOMAIN_CATALOG element can be created multiply and it is constructed for the one synchronizing data set. DOMAIN_CATALOG has DATA, TABLE and MDRID as child elements, and MDRID including FIELD as data column's attribute. FIELD includes data type and length, and existence of null and primary key can have 'yes' or 'no' attribute.


```

<?xml version="1.0" encoding="euc-kr" ?>
<G-XMDR xmlns="http://doc.kut.ac.kr" schemaLocation="XMDR.xsd">
  <DOMAIN_CATALOG do="1" db="yes">
    <DATA did="1">ProductControl</DATA>
    <TABLE tid="1">StockControl</TABLE>
    <MDRID mid="MDR0101">
      <FIELD type="varchar" length="15" null="no" key="yes" format="no">ProductCode</FIELD>
    </MDRID>
    <MDRID mid="MDR0102">
      <FIELD type="varchar" length="50" null="no" key="no" format="no">ProductName</FIELD>
    </MDRID>
    <MDRID mid="MDR0103">
      <FIELD type="datetime" length="0" null="no" key="no" format="yyyy-mm-dd">ProductDate</FIELD>
    </MDRID>
    <MDRID mid="MDR0104">
      <FIELD type="int" length="4" null="yes" key="no" format="no">ProductState</FIELD>
    </MDRID>
    .....
  </DOMAIN_CATALOG>
  <DOMAIN_CATALOG do="1" db="yes">
    <DATA did="1">ProductControl</DATA>
    <TABLE tid="2">WareControl</TABLE>
    <MDRID mid="MDR0201">
      <FIELD type="varchar" length="15" null="no" key="yes" format="no">ProductCode</FIELD>
    </MDRID>
    <MDRID mid="MDR0202">
      <FIELD type="varchar" length="50" null="no" key="no" format="no">ProductName</FIELD>
    </MDRID>
    .....
  </DOMAIN_CATALOG>
</G-XMDR>

```

Fig. 4. Defining G-XMDR Document of each Local Domain

Figure 4 shows to define G-XMDR with local XMDR of each local domain based on predefined XMDR of figure 3. This has restricted to the document validation of G-XMDR schema.

In this paper, we would like to mention that the types of data heterogeneity for data interoperation are very diverse and that semantic heterogeneity, type heterogeneity and expression heterogeneity will also be resolved. The characteristics for resolving data heterogeneity are described below. The data semantic heterogeneity is generated from the different structure at the same information, so the elements which make meaning sameness are included. DATA and TABLE are relational elements. MDRID element identifies the same semantic data sets as a target of synchronization, and it has FIELD as child elements. FIELD defines the attributes of data, and it has many attributes to represent data. The type heterogeneity is resolved by the type, length, null and key attributes in FIELD, and the expression heterogeneity can be eliminated by format attribute in FIELD. The remained elements in Schema are used for the semantic heterogeneity. Figure 4 is G-XMDR document which defines local domain's XMDR. It is located at the local synchronizing multi agent system, and resolves the distributed data heterogeneity through converting local data to XML data for global transmission. Also the distributed data's autonomy and independent are assured by using G-XMDR.

When the global synchronization is requested by local data's event, local database and table information is classified by 'did' and 'tid' which are their own attribute, and create the same attribute at the synchronizing XML. Also local data's column information is mapped MDRID value and synchronizing XML is created. This created synchronizing XML is transmitted globally.

The local systems in a local domain which receive the synchronizing XML map synchronizing XML to the local repository's data and query by G-XMDR. The

relational attributes, table and column information are extracted, and data type, length and expression are converted to local suitable type. This conversion resolves the data heterogeneity, and this system which is in charge of converting them is described next section.

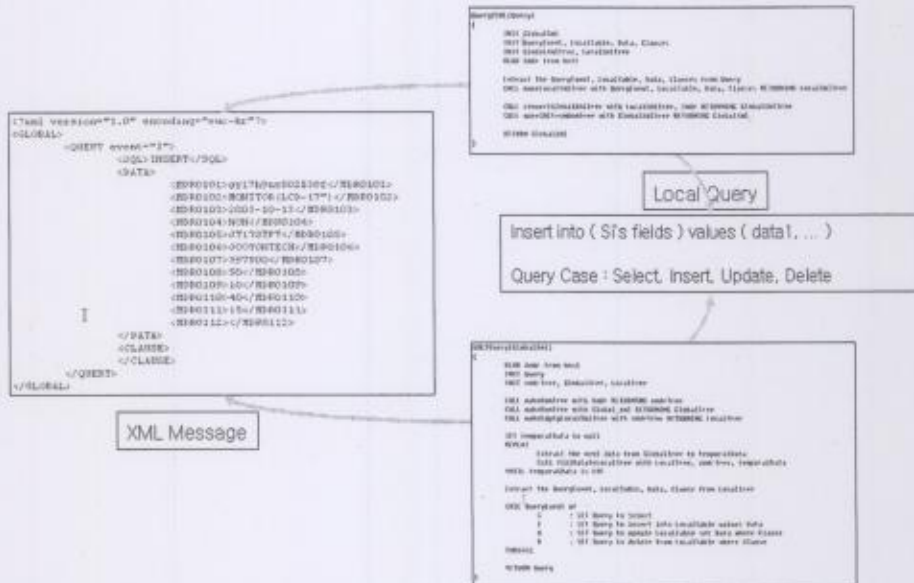


Fig. 5. Converting XML from/to Local Query using G-XMDR in a system

Figure 5 shows a procedure to convert XML into query through G-XMDR for data interoperability on a local system of each local domain. A query happened from local system can generate XML through this conversion's procedure, which is transmitted to all global area. Moreover, through XML-to-query algorithm, we can automatically convert XML, which transmitted from global one, into local query for which is applied on local system.

When the query manager converts data to the XML, it is composed differently as the kind of data changed event. The data changed event is classified with the data insert, delete and update. The synchronizing XML includes query attributes, QUERY and SQL which includes events as content. The QUERY attribute is composed differently as the event. The insert event's synchronizing XML includes data's MDRID and the other contents followed by MDRID. The update event's synchronizing XML includes CLAUSE element which is the where clause in SQL, and MDRID which will be updated, and data which will use for updating. The delete event's synchronizing XML includes only CLAUSE attribute for deleting. So the query of agent are available on the database which is used SQL, we need to research available query language[20] on pervasive computing environments as future works.

3 Multi Agent System

This section describes a synchronizing mechanism with the multi agent system which resolves the problem of data heterogeneity. Figure 6 shows a whole infrastructure of multi agent system. We would like to use G-XMDR to solve the problem of data heterogeneity with service agent against heterogeneous differences between pervasive computing environments.

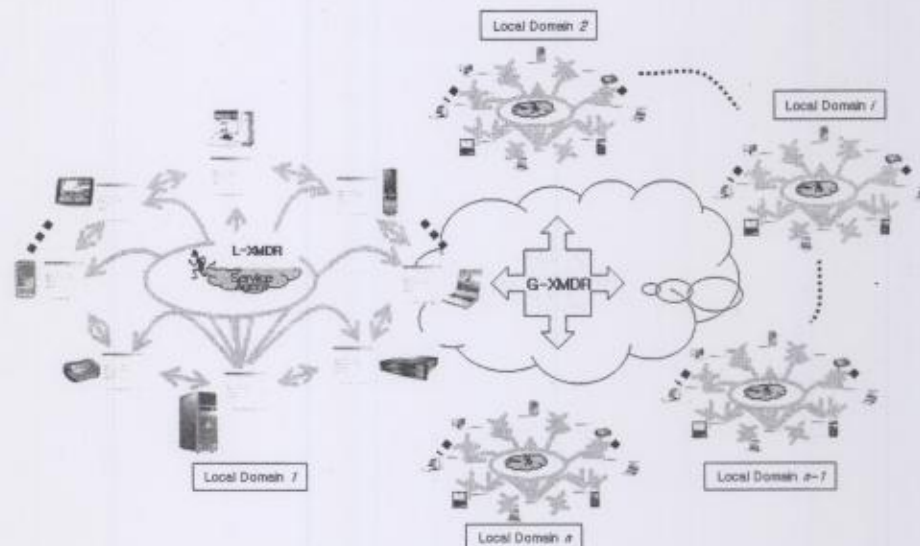


Fig. 6. The infrastructure of Multi Agent System

The direction of arrows in figure 6 appears the migration method of service agent. Each Legacy system must have its autonomous mechanism with multi agent system as the independent distributed system. In our approach, multi agent can be extended as service agent for data synchronization, and also include global XML converted with local query.

The synchronizing multi agent system consists of four layers. Each layer is communication module, system management module, data access module, and service module respectively. First, communication module should be a role of communication between agent systems and also consist of communication manager, security manager and log manager. Communication manager initiates and manages to communicate between one system launched and other system which launches service agent. Security manager is responsible for the authorization of service agent. Log manager manages and registers all events happened in agent system and all transmission/receipt of service agent. In this paper, this manager uses Java's basic security package to authenticate mobile agents. So the security policies of service agent's transmission are limited to the agent server's authentication, and further researches are necessary.

Second, system management module sets the priority of synchronization according to the target system's path and importance of system, and defines the basic information to access data. This module includes agent manger and system manger. The agent

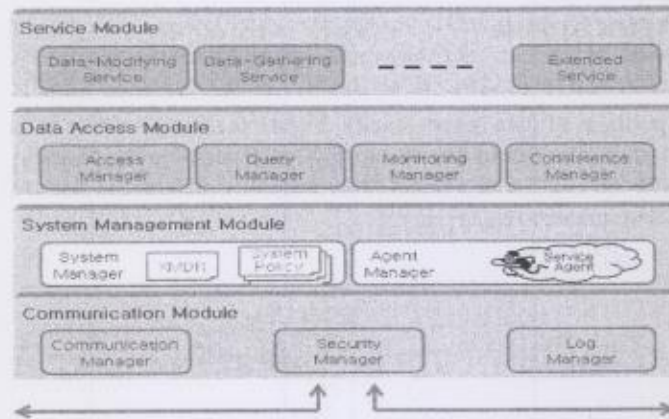


Fig. 7. The architecture of Multi Agent System

manager manages and controls policies between synchronizing systems, and records all the events from the synchronizing multi agent system. Also it authenticates synchronizing agent and manages related agents. The system manager defines G-XMDR based on L-XMDR which is a standard for data exchange and XML which is the document standard. G-XMDR is used for the conversion of XML, local data and query when synchronizing data's global transmission upon synchronizing request.

Third, data access module accesses data of legacy system. This module includes access manager, query manager, monitoring manager, and consistence manager. The access manger manages association among data, the query manger manages local query and global XML through G-XMDR, the monitoring manger requires data synchronization on global after recognizing the happened events from each autonomous and independent legacy system, and consistence manger avoids congestion from the data synchronization, and detects the event which changed data, and requests synchronization globally. This module's data consistency managing process is shown at [17].

Finally, service module may be very important one for practical application of data and also for providing diverse services with service agent. This module also includes data-modifying service and data-gathering service.

4 Implementation of the System

For the data synchronization at the distributed environment, 1) the platform should be independent, and 2) the data autonomy for update and delete operations should be assured and 3) the network traffic problem should be considered. To solve these problems, we use the service agent which reduces the network load and traffic, and runs at the local system after receiving data and code at run time from remote system. Java is the development language in this paper. In addition, XML is used as the standard document of resolving data heterogeneity for the interoperation, so JAXP1.0 is selected for the XML parser.

4.1 System Policy

The policy at the synchronizing multi agent system is consists of 3 parts, which are database information, path information and system priority. First, the database