

Documents of SDM-DS 2006

Proceedings of the 3rd International Workshop on
Software Development Methodologies for Distributed Systems

Edited by

Beijun Shen 沈备军
Haopeng Chen 陈昊鹏

Sponsored by



Shanghai Jiao Tong University
上海交通大学



上海科学技术文献出版社
Shanghai Scientific and Technologic Literature Publishing House

Series: SDM-DS 2

Documents of SDM-DS 2006

Proceedings of the Third International Workshop on
Software Development Methodologies for Distributed Systems

分布系统软件开发技术文集

SDM-DS 2006 年会议录

Edited by

Beijun Shen 沈备军
Haopeng Chen 陈昊鹏

Sponsored by

Shanghai Jiao Tong University
上海交通大学



上海科学技术文献出版社

图书在版编目(CIP)数据

分布系统软件开发技术文集/沈备军等主编. —上海:
上海科学技术文献出版社, 2006. 9
ISBN 7-5439-3028-5

I. 分... II. 沈... III. 分布式操作系统—软件开发—国际学术会议—文集 IV. TP311.52-53

中国版本图书馆CIP数据核字(2006)第108360号

责任编辑: 盛 强

封面设计: 徐 利

分布系统软件开发技术文集

沈备军 陈昊鹏 主编

*

上海科学技术文献出版社出版发行
(上海市武康路2号 邮政编码200031)

全国新华书店经销

江苏常熟人民印刷厂印刷

*

开本889×1194 1/16 印张6.25 字数244 000

2006年9月第1版 2006年9月第1次印刷

ISBN 7-5439-3028-5 / Z · 1111

定价: 20.00元

<http://www.sstlp.com>

内容简介——Brief Introduction

SDM-DS 2006 会议于 2006 年 5 月 29 日在上海交通大学徐汇校区顺利召开, 本届 SDM-DS 国际学术会议的主题为“从理论到实践”, 涉及的领域包括: 与分布式系统和技术相关的基础问题与教育问题, 适应分布式系统的开发过程, 分布式系统环境中的架构优缺点分析, 大规模分布式系统开发的案例研究, 分布式系统的测试与校验, 分布式系统的性能分析, 遗留分布式系统的逆向工程与图形化建模、工具和实践经验, 工业界会影响到分布式系统设计的趋势以及外包、BPO 趋势对软件开发方法学的影响。

本次会议借全球软件工程专家齐聚上海参加 ICSE2006 的良机, 邀请了专家学者讨论分布式软件系统的开发方法学的最新研究成果, 为推动该领域技术和理论的发展以及成果的推广做出了贡献。

本次会议及论文集得到了上海交通大学的资助。

SDM-DS 2006 was held on May 29, 2006 at Shanghai Jiao Tong University, Shanghai, China. The theme of the conference was “from theory to practice.” The topics covered in the conference include distributed system fundamentals, development process, architecture, case studies, test, evaluation, and performance. In addition, the conference also saw presentations on reverse engineering of legacy distributed systems, graphic modeling, tools, and practice in distributed system design and development, the trend of distributed systems, and the impact of outsource and BPO on distributed software development methodology.

This conference took place at an opportune time for two reasons: the gathering of ICSE 2006 in Shanghai and the increasing acceptance of distributed software development in China. Taking advantages of the ICSE 2006, the conference invited many scholars and professionals to present the newest results in distributed software development methodology and related topics. The talks given by the invited speakers proved to be one of the highlights of the SDM-DS 2006.

The conference was made possible by a grant from Shanghai Jiao Tong University.

**The Third International Workshop on
Software Development Methodologies of Distributed Systems
SDM-DS 2006
(ICSE 2006 co-location event)**

Shanghai, China
May 29, 2006

Conference Chair

Yuxi Fu, Shanghai Jiao Tong University, China

Organizing Committee

Chair

Zongyao Zhu, Shanghai Municipal Informatization Commission, China

Organizers

Haopeng Chen, Shanghai Jiao Tong University, China

Shihong Huang, Florida Atlantic University, USA

Biqing Huang, Tsinghua University, China

Conference Secretary

Xun Liu, Shanghai Jiao Tong University

Treasurer

Zhihua Su, Shanghai Jiao Tong University, China

Web Master

Jiaming Zhang, Shanghai Jiao Tong University, China

Program Committee

PC Chair

Beijun Shen, Shanghai Jiao Tong University, China

PC Members

Scott Tilley, Florida Institute of Technology, USA

Jing Dong, University of Texas at Dallas, USA

Sol M. Shatz, University of Illinois at Chicago, USA

Cyril S. Ku, William Paterson University, USA

Jingde Cheng, Saitama University, Japan

Keijiro Araki, Kyushu University, Japan

Jianwei Jiang, Shanghai Jiao Tong University, China

Jianjun Zhao, Shanghai Jiao Tong University, China

Hengming Zhou, Shanghai Jiao Tong University, China

Baowen Xu, SouthEast University, China

Advisory Committee

Zhiying Zhou, Tsinghua University, China

Dehua Ju, Institute of Software Chinese Academy of Science, China

Preface

This proceeding contains all the papers presented at the Third International Workshop on Software Development Methodologies of Distributed Systems (SDM-DS 2006), held in Shanghai Jiao Tong University, Shanghai, China on May 29, 2006.

While distributed system theories have matured over the past several decades, the design and implementation of actual distributed systems has proved to be an engineering challenge. Many factors such as fuzzy boundaries, fault tolerance, load balance, and data consistency have contributed to the complexity of the development of distributed systems.

SDM-DS 2006 attempts to bridge the gap between theory and practice. In fact, the theme of the workshop was “from theory to practice”. The goal was to bring together researchers and practitioners of distributed systems and software engineering to discuss and exchange innovative methods for designing, constructing, and managing large-scale distributed systems as well as to explore development methodologies for distributed systems.

Researchers interested in distributed system development focus on the underlying theory and structure such as software engineering processes, software architecture, and net-centric computing. Practitioners are concerned with quality attributes that affect the implementation of a distributed system. The quality attributes include such things as Internet standards, quality of service, usable tools, and budget and time constraints. Yet educators need to provide students with realistic and professional skills that are required in the design and development of distributed systems.

SDM-DS 2006 provided a forum for people of those aforementioned background settings to exchange ideas and results related to distributed system development. Thanks to the participation by authors, keynote speakers, and volunteers, the workshop achieved great success.

June 2006

Beijun Shen
Haopeng Chen

Table of Contents

Development Methods

Development Model and Realization of Web Report Applications Based on XML <i>Shangming Zhu , Guisheng Fan</i>	1
The DB and XML Design Solution of Workflow Organization Model Based on RBAC <i>Nan Li , Yan Zhang</i>	5
Evaluation of Transaction Management Based on Spring AOP <i>Cheng Zhang</i>	12

Algorithms and Analysis

An Analysis about the Business Transaction Based on Service-Oriented Architecture <i>Zhiyong Wang</i>	19
A Parallel Algorithm for Mining Association Rules in a Distributed System <i>Huiwei Guan , Songnian Yu , Horace Ip</i>	24
Web Service Discovery Based on Similarity of Words in Pervasive Computing <i>Qi Ye , Guoxing Song</i>	32
An Application of Probability Change Point Analysis to the Prognostication of Throughput of Grid Services <i>Haopeng Chen</i>	35

Software Architecture

Distributed Business Integration Using Enterprise Service Bus <i>Sibo Zhang</i>	39
A Framework Supporting Collaborative Computing on Grid <i>Dejun Wang , Linpeng Huang , Xiaohui Xu , Lijun Li</i>	46
Towards Semantic Data Driven Presentation Level Integration of Remote Applications and Services <i>Jingyu Song , Jun Wei , Shuchao Wan</i>	52
Software Process Support over Internet <i>Beijun Shen , Bo Jin</i>	63

Others

The Application of Dynamic Binary Translation in Distributed Virtual Execution Environments <i>Jun Li , Haibing Guan , Yuncheng Bao , Alei Liang</i>	67
Mobilizing Native Machine Code via Dynamic Binary Translation <i>Yuncheng Bao , Haibing Guan , Jun Li , Alei Liang</i>	73
Self-Repairing File Recovery <i>Zhifu Tang , Hengming Zou</i>	79
Continuous Bayesian Learning for Spam Filtering <i>Shaojun Ning , Hengming Zou</i>	85

Appendix

A Brief Summary of SDM-DS 2004 – 5(W) <i>Talasila Rao</i>	90
--	----

Development Model and Realization of Web Report Applications Based on XML *

Shangming Zhu, Guisheng Fan

(Department of Computer Science and Engineering, East China University of Science and
Technology, Shanghai, 200237)

{ zhusm@ecust.edu.cn, fanggss@yahoo.com.cn }

Abstract

The general development models based on XML under WEB environment are presented in this paper. A development model of web report applications based on XML is designed and discussed in detail. To realize web report applications based on XML, the development procedure includes two parts: server development and client development. At server, data is organized to XML document and uploaded on web site; at client, XML document is downloaded and transformed to web report according to XSL style by browser. The development model and realization method of web report applications have certain practicability and generality.

1 Introduction

It is difficult for the traditional client/server development model to deal with the more and more complicated applications and rapidly increasing data exchanges as computer technologies are commonly used in various businesses. Because a large amount of data is processed at client, data exchanging between client and server is continuously needed, which leads network system to be very strenuous. In addition, the distribution of applications brings about the difficulty of system maintenance and increases the cost of maintenance. Therefore, a new development model is demanded to solve this problem.

XML (extensible markup language) is a new extensible markup language on web network, and is very welcome in dealing with data presentation and exchange. XML originates from SGML (Standard Generalized Markup Language), it is a new tool to construct web network in the future^[1-2]. One of the

advantages of XML is its ability of describing hierarchical data, or presenting a clear structure to disorderly data.

The most basic XML standard includes three interconnection standards: XML, XSL (Extensible Style Language) and XLL (Extensible Linking Language). The three standards supplement each other, and XML becomes very powerful and flexible in data markup, presentation style and hypertext linking. It makes data exchange and integrity be very convenient.

Two general development models based on XML under WEB environment are presented, and a development model and its realization method of web report applications based on XML are designed and discussed in detail in this paper.

2 Development Models Based on XML under Web Environment

Next WWW environment based on XML is designed for web data, not only can it be compatible with existing web applications, but also can realize to share and exchange information on web network. In addition, because of XML's independence of platforms, XML is very suitable to be a middle data carrier of applications. This feature of XML satisfies the need of tie-layer development mode. In tie-layer development mode, XML data is transported among client, application server and database server.

The web applications based on XML can be divided into the followings: information transport in different databases of network clients; data downloaded from web server to client; different data presentation styles to same data source according to different users' demands; and natural language translation, web business

* This work was partially supported by the NSF of China under grants No. 60473055 and 60373075, Shanghai Pujiang Program under grant No. 05PJ14030.

applications, etc.

As a description language, XML is robust to application and independent on platforms, it can process information and realize web applications by programming. There are two general development models based on XML under web Environment: Web Data Retrieve Model and Visual Database Query Model.

2.1 Web Data Retrieve Model

Web data retrieve model is illustrated in Figure 1. On a client, browser finds target HTML page through hyperlink, gets related information, retrieves data, filters these information and creates structured XML document.

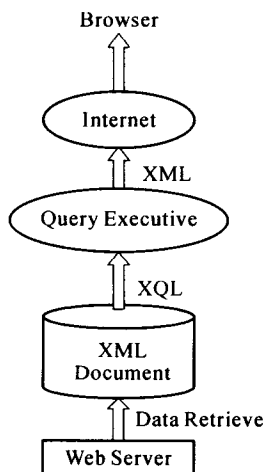


Figure 1. Web data retrieve model

2.2 Visual Database Query Model

Visual Database (VDB) query model is depicted in Figure 2. According to query requirement of user database and web application, VDB server extracts data from related web sites, and returns the query result to user application by XML.

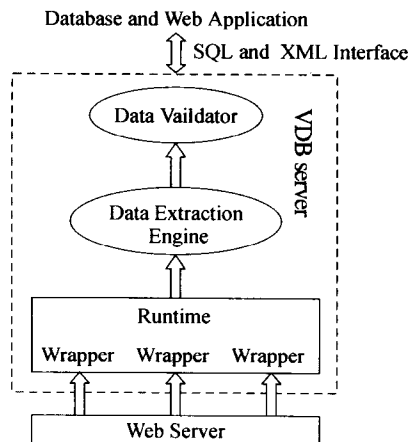


Figure 2. Visual database query model

3 A Development Model of Web Report Applications

To develop web report application, web data retrieve model, which is commonly used, is improved as Figure 3. In Fig. 3, client is simply regarded as a web browser, server uses XML to organize data and uploads XML document to web site. Because XML document is pure text document, it can be edited by any text editing tool online or offline. At client, XML document is downloaded, resolved by XSL style, and presented in browser.

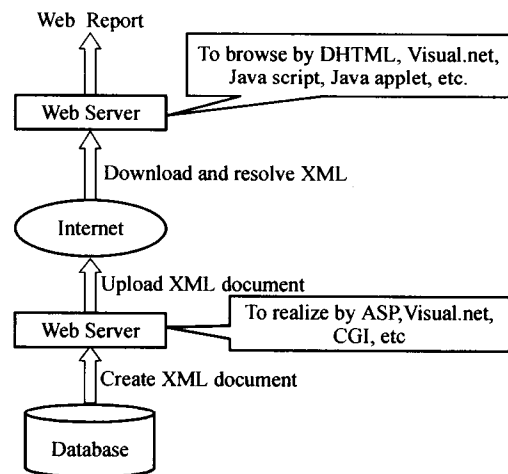


Figure 3. Model of Web report application

4 Realization of Web Report Applications

To realize the web report applications based on XML, the development procedure includes two parts: server development and client development. At server, Data is organized to XML document and uploaded on web pages; at client XML document is downloaded and transformed to web report according to XSL style in browser.

4.1 Creation of XML Document at Server

There are many methods to create a XML document at server. In this paper, Visual. Net technology is used to create XML document at server. XML and .NET is completely integrated, many server controls under .net development environment have the abilities to create XML document^[3]. For instance, DataSet class includes many methods to create XML document. In this paper, as an example, data is retrieved from database and filled in a DataSet, then data is output from dataset and saved as an XML document. DTD restrictions can also be added to ensure

the integrity of data.

In this paper, STR is assumed to be a SQL string, CON is assumed to be a SQL Connection object, DS is assumed to be a Data Set, and SDA is assumed to be a SQL Data Adapter. The following procedure illustrates a programming example to create XML document from dataset (Figure 4).

```
1  PROCEDURE CREATEXML();
2  SqlDataAdapter SDA = new SqlDataAdapter(S, CON);
3  CON.Open();
4  SDA.Fill(DS, "example");
5  DS.WriteXml("example.xml",
    XmlWriteMode.IgnoreSchema);
6  CON.Close();
```

Figure 4. Create XML procedure

Database is opened and data is retrieved from database and filled into Dataset in lines 2 – 4; dataset is saved as an xml file in line 5; and database is closed in line 6.

4.2 Resolution of XML Document

After XML document is created and transported from server to client through browser, it must be resolved at client. XML document can be resolved as any document that is consistent with XML format. In this paper, XSL (Extensible Style Language) is used to resolve XML^[4]. XSL can be defined by `<xsl: template match = "/">` element and `<xsl: template>` element. These elements are used to define a template for data in query result tree. All style information is contained in the templates defined by `<xsl: template>` elements in XSL document. To Create XSL document, firstly, `<xsl: template match = "/">` is used to define a template which matches the root element in XML document; then `<xsl: for-each>` element in template is used to search for each required element in XML document, and all the found elements are processed according to defined operations. At the same time, `<xsl: if>`, `<xsl: choose>` and `<xsl: when>` element in template are used to filter the related elements in XML document, and the filtered elements are processed according to styles and operations defined in result tree and style file. Element `<xsl: apply-templates>` can also be used to match the template, which can be regarded as a call procedure.

4.3 Presentation of XML Document at Client

To output web report to end user, XML data must be transformed to HTML document according to XSL style file. Three files are demanded to transform XML

data to HTML format: one is a source XML document, which is the object to be processed; one is a XSL style document, which is the tool to realize the conversion; the other is a HTML file, which is the output result — web report.

To transform XML data, `load()` and `Transform()` method of `XslTransform` class are used, the former's function is to load a XSL style file, the latter's function is to start the conversion. The following procedure illustrates a programming example to transform XML document to web report and present it at client (Figure 5).

```
1  PROCEDURE SHOWXML();
2  XmlDocument XmlInformation = new
    XmlDocument();
3  XmlInformation.Load("http://.../example.xml.
    xml");
4  XmlInformation.Save("C:\\...\\example.xml");
5  XmlDocument xmldoc = new XmlDocument();
6  xmldoc.Load("C:\\...\\example.xml");
7  XslTransform xslDoc = new XslTransform();
8  XslDoc.Load("C:\\...\\example.xsl");
9  Response.Clear();
10 Response.ContentType = "text/html";
11 Response.Clear();
12 XslDoc.Transform(xmldoc, null, Response.Output)
13 Response.End()
14 END PROCEDURE
```

Figure 5. Show XML procedure

XML file is downloaded from web server in lines 2 – 4; XML file is reloaded at client in lines 5 – 6; XSL file is loaded and bound to the Response object at client in lines 7 – 11; and XML file is transformed and output in lines 12 – 13.

XML document can be edited at client and uploaded to web server from client as well. XML view can be created to map dataset contents^[5], and data can be viewed and operated by relation or XML method, so that both data integrities are ensured and the accuracy of web report is ensured too.

5 Conclusion

Above procedures and examples are developed under .net framework; information is retrieved from remote database to dataset, dataset is transformed to XML data, XML data is downloaded to client. XML data can be edited and presented directly at client, which needn't keep a network connection between server and client any more, so the network traffic and maintenance cost is decreased and client can present

data in different report forms by different XSL styles. These procedures and examples also show that it is convenient to create web report based on XML and independent on platform, language or limitation. The development model and realization method of web report applications in this paper have certain practicality and generality.

References

- [1] Extensible Markup Language (XML) 1.0 [S] Second Edition, W3C Recommendation, <http://www.w3.org/TR/2000/REC-xml-20001006>, 200 - 10.
- [2] Ouahid Karmouch A. Converting Web pages into well-formed XML document. IEEE, ISBN 0 - 7803.5284 -X, 1999.
- [3] Mark B, Richard B. ASP and XML advanced programming. Qsinghua University Pressing, 2002.
- [4] Clark J. XSL Transforms (XSLT) version 1.0. W3C Recommendation, 1999.
- [5] Jacinto M H, Librelotto G R, Ramalho J C, *et al.* Bidirectional conversion between XML documents and relation databases. The 7th International Conference on Computer Supported Cooperative Work in Design 2002 [C], 2002: 437 - 443.

The DB and XML Design Solution of Workflow Organization Model Based on RBAC

Nan Li

School of Software ,
Shanghai Jiao Tong University
arctic@sjtu.edu.cn

Yan Zhang

School of Software ,
Shanghai Jiao Tong University

Abstract

The organization model of Workflow system should be highly flexible and highly reliable and suitable for most enterprise applications. RBAC reference model is an important authorization model of workflow system. This paper describes a workflow organization model based on RBAC and adapted to the complex reality of enterprise organization structure. Also to improve this model's availability and security, we design this organization model based on both DB and XML, and the latter can be used in Web Service and we encode it using SDES (Simplified DES) and RSA encoding algorithm when transferring it.

1 Introduction

With the acceleration of social network's development, E-business is applied in more and more fields and enterprises in the world. Workflow technology plays a key role as an enabler in E-Business applications, such as supply chains and CRM (Customer Relationship Management). We can divide a workflow system into two important perspectives: the process flow perspective and the organizational perspective. The process flow perspective describes the routing and coordination of tasks within a process. The organizational perspective deals with realities of the organization.

According to these two perspectives of a workflow system, it is obvious that organization model will affect no less than half of the system. Then it is very necessary for a workflow system to model organizational policies carefully so that the workflow system can determine who will do the work.

The nonstop development of network also makes the Computer Security problem more and more serious and the solutions of Computer Security more and more

important. Among these various solutions of Computer Security, Access Control, or called Authorization, is one of the most important ones. There are many traditional access control mechanisms^{[2][11]} such as DAC (Discretionary Access Control)^[14] and MAC (Mandatory Access Control)^[15]. But these mechanisms are not adaptive for the requirements of most systems especially distributed system^[5]. RBAC (Role-Based Access Control) reference model^{[1][3][6][12]} is an open model and can be well-expanded. It can solve the problems of traditional mechanisms perfectly and satisfy the security requirements of the distributed system exactly. So in our workflow system, we use RBAC reference model to solve the problems among the process flow perspective and the organizational perspective, so that the workflow system can determine who can do which work and what he can do during doing that work.

For the requirements of EAI (Enterprise Application Integration) and for better B2B (Business-to-Business) communication, for the requests of many new techniques such as BPEL (Business Process Execution Language) and SOA (Service-Oriented Architecture)^[7], it becomes imperative for us to consider this model redefine in XML style^[4] in order to use it in Web Service for many special requests. For example, if the Security Department of this company wants to get the organization data to check out whether the privileges assigned to each employee are appropriate, or if the parent company in Shanghai wants to get the organization data of its child company in Guangzhou, we can transfer this XML-styled organization model within Web Service. Also because when the organization data in a company is once set up, they usually would not be changed frequently. Then the XML-styled organization model would not be changed frequently too. And for security, we encode these data

using SDES (Simplified DES) and RSA encoding algorithm when transferring them.

2 The Relative Factors for Designing Workflow Organization Model

To design a flexible and reliable organization model, we should consider these three relative factors: the RBAC Reference Model, the reality of enterprise organization structure and some non-functional requests. And in the first two sub-sections of this section, we will only describe the relative DB design solution of our organization model. In the next section, we will describe our DB and xml-styled organization model in detail.

2.1 RBAC Reference Model

According to the requirement of permissions in different complexity, RBAC reference model defines three layers: Core RBAC, Hierarchical RBAC and Constrained RBAC.

2.1.1 Core RBAC

Core RBAC defines the basic requirements of all RBAC systems, it contains five basic elements: User, Role, Operation, Object and Privilege (relative to USERS, ROLES, OPS, OBS, and PRMS in Figure 1). Noted that Object is equivalent to Resource, and Privilege is equivalent to Permission.

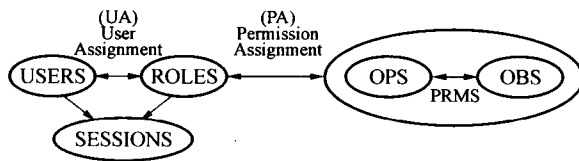


Figure 1. Core RBAC

Core RBAC pays attention on the relationship between Role and User and between Role and Permission, that is UA (User Assignment) and PA (Permission Assignment) in Figure 1. Both these two relationships between every two elements are many-to-many. That is to say a role can be assigned to many users and a user can play many roles; a role can have many permissions and a permission can be assigned to many roles.

Note: Figure 1 described the standard Core RBAC model. In Figure 1, there is another entity called SESSIONS. A SESSION means which ROLES a USER can have on a specific situation. As it is not a common use in organization model, our paper omits this entity in Figure 1. We also omit it in Figure 2 and 3.

To design DB organization model based on Core RBAC, we define entities as follows:

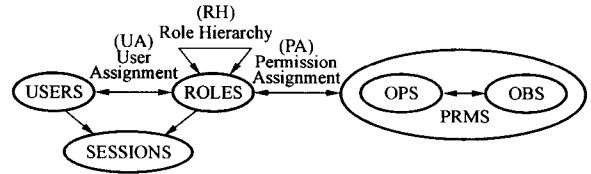


Figure 2. Hierarchical RBAC

Definition 1 ST_ROLE is an entity corresponding to Role in Core RBAC. It has the attributes Role_Id, Role_Name and so on. It will be described later.

Definition 2 ST_RESOURCE is an entity corresponding to Object (Resource) in Core RBAC. It has the attributes Resource_Id, Resource_Name, Resource_Object and so on.

Definition 3 ST_PERMISSION is an entity corresponding to Operation in Core RBAC. It has the attributes Permission_Id, Permission_Name, Operation_Type and so on.

And (ST_RESOURCE, ST_PERMISSION) is corresponding to Privilege (Permission) in Core RBAC. In order to be simple, we don't design any other entity corresponding to Privilege in Core RBAC.

Definition 4 ST_USER is an entity corresponding to User in Core RBAC. It has the attributes User_Id, User_Name, User_Pass, User_Age, User_Phone, User_Email and so on.

We also define relationship objects which describe the relationships among these entities:

Definition 5 ROLE_RESOURCE_PERMISSION is a relationship object which describes the relationship among entities: ST_ROLE, ST_RESOURCE and ST_PERMISSION. It has the attributes Role_Resource_Permission_Id, Role_Id, Resource_Id and Permission_Id.

Definition 6 USER_ROLE is a relationship object which describes the relationship between ST_USER and ST_ROLE. It has the attributes User_Role_Id, User_Id and Role_Id.

The relationships between these can be called constraints of this organization model, we divide them into three types: one-to-many, many-to-many and self-mapping. And we also describe them as follows:

2.1.2 Hierarchical RBAC

In Hierarchical RBAC, Role would be divided into different levels (See Figure 2). A Role can inherit from another Role. The inheriting Role will have all Privileges in the inherited Role. For example, if there are two Roles: Employee and Manager. And Manager will have all Privileges which Employee have, then we can say Manager inherit from Employee. And we

describe the chain as Employee \leftarrow Manager. So for Manager, Employee is its upper level role.

To design DB organization model suitable for Hierarchical RBAC, we update ST_ROLE's definition as follows:

Definition 1 (version 2) ST_ROLE is an entity corresponding to Role in Core RBAC. It has the attributes Role_Id, Role_Name, UpperRole_Id and so on. Note that UpperRole_Id references to the Role which is the parent Role of current Role. In the example of Employee \leftarrow Manager, Manager's UpperRole_Id is the Role_Id of Employee.

2.1.3 Constrained RBAC

In Constrained RBAC, it requests SOD (Separation of Duties) which means to avoid conflicts between Roles (See Figure 3). For example, suppose there are two Roles: Teacher and Assistant. And person can be either Teacher or Assistant but can't be both. Then we describe \langle Teacher, Assistant \rangle to represent this constraint and use SOD to separate these two Roles.

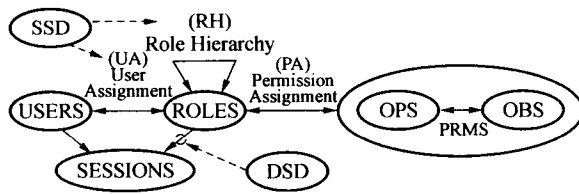


Figure 3. Constrained RBAC

Constrained RBAC provides two relations of SOD^[13]:

- **SSD (Static Separation of Duty)**: SSD relations describe the roles which are conflict to each other. All roles' assignments should observe these constraints. That means if a role A is assigned to a user and there is some role B which is conflict to that role A, then this user can't be assigned to that conflicting role B.
- **DSD (Dynamic Separation of Duty)**: DSD relations place constraints on the roles that can be activated in a user's session. If one role that takes part in a DSD relation is activated, the user cannot activate the related (conflicting) role in the same session. Because in our paper we omit SESSION entity, we also omit the realization of DSD.

To design DB organization model suitable for Constrained RBAC, we update ST_ROLE's definition as follows:

Definition 1 (version 3) ST_ROLE is an entity corresponding to Role in Core RBAC. It has the attributes Role_Id, Role_Name, UpperRole_Id, Sepa-

ratedRoles_Id and so on. Note that SeparatedRoles_Id references to the Roles which can't be applied to the same person with current Role. In the example of \langle Teacher, Assistant \rangle , Teacher's SeparatedRoles_Id contains the Role_Id of Assistant and vice versa. And if the Role has more than one conflict Roles, then we represent this Role's SeparatedRoles_Id as the set of {conflictRole1_Id, conflictRole2_Id, UpperRole_Id ...}

2.2 The Reality of Enterprise Organization Structure

The organization type of modern enterprise can be divided into four types^[8-9]:

- **Linear Type**: In this type there is no functional department and the leadership is linear style from the top level to the bottom level of management layer.
- **Linear and Functional Department Type**: Compared to the Linear Type, there are some functional departments which are set to serve different level of leaders. These functional departments deal with professional management and act as current leader's secretaries.
- **Business Department Type**: According to the different products or places or markets, the company would be separated into many semi-independent business units. Each unit is engaged in the activities from product design to product sales.
- **Matrix Type**: In this type, the company combines the departments which are divided by functions and the groups which are divided by projects together to compose a matrix. Any employee can be connected within his functional department and at the same time take part in the project work.

Obviously, the fourth type is the most common and complex one in E-business. So we design our organization model according to this type^[10]. The extra entities in our organization model are as follows:

Definition 7 ST_DEP is an entity which describes the functional departments. It has the attributes Dep_Id, Dep_Name, Dep_ManagerPostId, UpperDep_Id and so on. Noted that Dep_ManagerPostId references to the ST_POST which is the leader's ST_POST of this ST_DEP and UpperDep_Id references to the ST_DEP which is the superior ST_DEP of current ST_DEP.

Definition 8 ST_POST is an entity which describes the person's position in the functional departments. It has the attributes Post_Id, Post_Name, Dep_Id, UpperPost_Id and so on. Noted that Dep_Id references

to the ST_DEP which department this ST_POST belongs to and UpperPost_Id references to the ST_POST which is the superior ST_POST of current ST_POST.

Definition 9 ST_GROUP is an entity which describes groups divided by project. It has the attributes Group_Id, Group_Name, Group_ManagerId and so on. Note that Group_ManagerId references to the ST_USER who is this group's leader.

There are also some special situations may happen in the complicated enterprise organization structure, such as:

- A user: can have many posts; can belong to many departments; can be leaded directly by many leaders; can have many subordinates; can belong to many groups and so on.
- A department: can have only one superior department; can have many subordinate departments; can have many posts; can have many users and so on.
- A post: can have only one superior post; can have many subordinate posts; can have many users who are just in this post (for example we can have many directors of the same post) and so on.
- A group: can have many users in this group; can play many roles and so on.

So we define relationship objects which adapt to these special situations as follows:

Definition 10 USER_POST is a relationship object which describes the relationship between ST_USER and ST_POST. It has the attributes User_Post_Id, User_Id and Post_Id.

Definition 11 GROUP_USER is a relationship object which describes the relationship between ST_USER and ST_GROUP. It has the attributes Group_User_Id, User_Id and Group_Id.

Definition 12 ROLE_GROUP is a relationship object which describes the relationship between ST_ROLE and ST_GROUP. It has the attributes Role_Group_Id, Role_Id and Group_Id.

2.3 Some Non-functional Requests

2.3.1 The Reason of Representing Organization Model as Web Service

We build Web Service for this organization model for these reasons:

- 1) The organization data of a company are seldom changed once they are set up, so we can put them as a Web Service.
- 2) If we can't access or can't access easily the Database which contains these organization data, for

example we are in America but the Database lies in China, then we can access these data easily by Web Service.

3) As the development of BPEL and other network technology, we make these data as a Web Service so that we can interoperate with other companies (for example our sub-companies).

4) Making these data as a Web Service will ignore the platform of computers which use these data.

2.3.2 Network Security

Once we make organization model as Web Service, we should pay attention on the security problem. To solve this problem we design our access process (using RSA algorithm^[16] and SDES algorithm^[17]) as follows:

- 1) Client requests the Web Service for organization data.
- 2) Client calculates the public key and private key (using RSA algorithm).
- 3) Server uses the public key to calculate the encoded key of symmetry encoding algorithm.
- 4) Server transfers the encoded result *encodedKey* to Client.
- 5) Server transfers the encoded organization data *encodedData* which are encoded by symmetry encoding algorithm (using SDES algorithm).
- 6) Client decodes the *encodedKey* by its private key and then gets the result *symmetryKey*.
- 7) Client decodes the *encodedData* by *symmetryKey* using symmetry encoding algorithm (using SDES algorithm).

3 The Design of Workflow Organization Model

In this section, we will describe both DB and XML-styled Workflow Organization Model in detail.

3.1 The Design of DB Workflow Organization Model

In our DB organization model, we design 12 objects (including 7 entity objects and 5 relationship objects) which are mentioned above. See section 2.1 and section 2.2. The relationships between these entity objects can be mainly classified into 4 types: one-to-many mapping, many-to-many mapping, self-mapping and other special mappings. And we also describe these constraints as follows:

Constraints:

● Type 1 one-to-many mapping

One-to-many mapping can be found between ST_DEP and ST_POST:

- 1) $\forall post(ST_POST(post))$
 $\Rightarrow \exists ! dep(ST_DEP(dep) \wedge (post.Dep_Id = dep))$

$Dep_Id)))$

- 2) $\forall dep \exists post (ST_DEP(dep) \wedge ST_POST(post) \Rightarrow post.Dep_Id = dep.Dep_Id)$

● **Type 2 many-to-many mapping**

Many-to-many mapping can be found between ST_USER and ST_POST, ST_USER and ST_DEP, ST_USER and ST_GROUP, ST_USER and ST_ROLE, ST_ROLE and ST_GROUP, ST_ROLE and ST_RESOURCE, ST_ROLE and ST_PERMISSION.

We will take the mapping among ST_USER and ST_ROLE as an example; the others are as the same. And also we introduce the relationship object USER_ROLE;

- 3) $\forall ur \exists user (USER_ROLE(ur) \wedge ST_USER(user) \Rightarrow ur.User_Id = user.User_Id)$
 4) $\forall ur \exists role (USER_ROLE(ur) \wedge ST_ROLE(role) \Rightarrow ur.Role_Id = role.Role_Id)$
 5) $\forall a, b \in USER_ROLE (a.User_Id = b.User_Id \Rightarrow a.Role_Id \neq b.Role_Id)$
 6) $\forall a, b \in USER_ROLE (a.Role_Id = b.Role_Id \Rightarrow a.User_Id \neq b.User_Id)$

● **Type 3 self-mapping**

Self-mapping can be found in ST_DEP, ST_POST

and ST_ROLE;

We will take the mapping in ST_ROLE as an example; and the others are as the same:

- 7) $\forall role \in ST_ROLE (role.UpperRole_Id \neq null \Rightarrow (role.Role_Id \neq role.UpperRole_Id) \wedge (\exists uprole \in ST_ROLE, uprole.Role_Id = role.UpperRole_Id))$

● **Type 4 other special mappings.**

We will just take two of them as examples:

- 8) $\forall role \in ST_ROLE (role.SeparatedRoles_Id \neq null \Rightarrow (role.Role_Id \neq role.SeparatedRoles_Id) \wedge (\forall roleid \exists serole, role.SeparatedRoles_Id (roleid) \wedge ST_ROLE(serole) \Rightarrow (serole.Role_Id = roleid) \wedge (role.Role_Id \in serole.SeparatedRoles_Id)))$
 9) $\forall dep \exists post (ST_DEP(dep) \wedge ST_POST(post) \Rightarrow dep.Dep_ManagerPostId = post.Post_Id)$

And the relationship graph of all 12 objects (corresponding to tables in DB) is as follows:

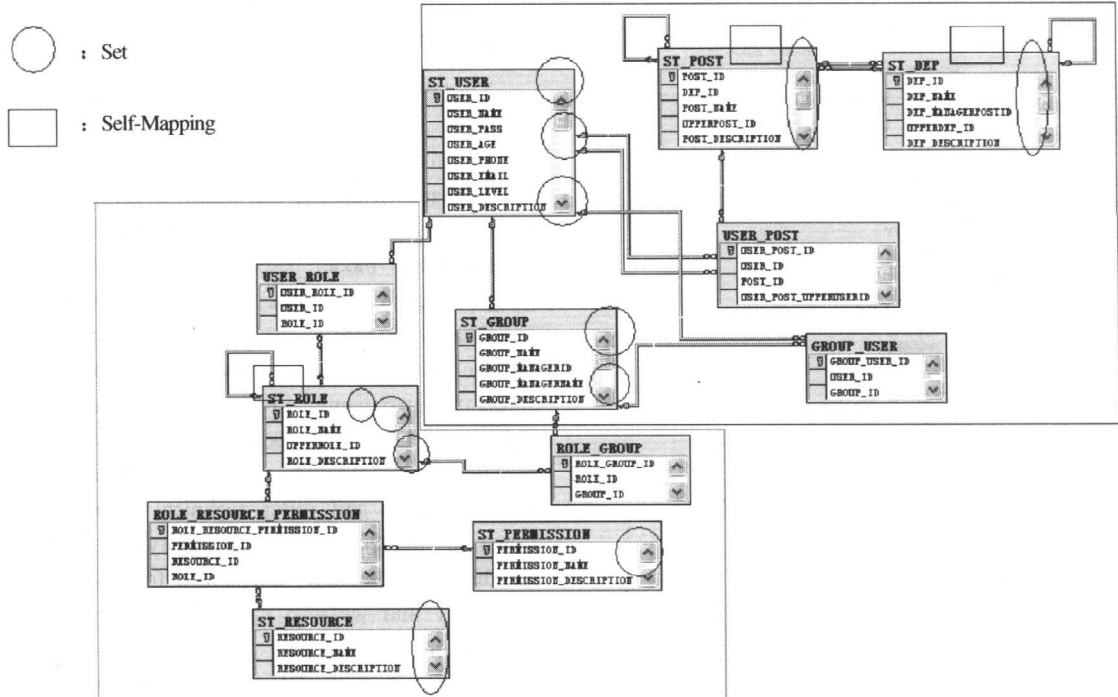


Figure 4. The relationship graph of DB Workflow Organization Model

Noted that the green-bordered polygon (the left polygon) in Figure 4 represents the RBAC part of this model and the red-bordered rectangle (the right rectangle) in Figure 4 represents the real enterprise organization structure part of this model. And the most

important relationships which connect these two parts are the USER_ROLE and ROLE_GROUP relationship objects.

In implementation of DB Workflow Organization Model, we use Hibernate to set up persistent layer based

on this model. In Figure 4, the blue-bordered rectangles represent the self-mappings and the blue-bordered ellipses represent the sets which are defined in Persistent Layer using Hibernate technology.

3.2 The Design of XML Workflow Organization Model

In our implementation of workflow system, we use this model mainly in web service layer. In fact, this model can be used independently without DB Workflow Organization Model. It can act as XML Database for organization data perfectly.

When we begin to define the structure of this model, there are three points that we must take notice of:

- 1) For the self-mappings, we design every entity

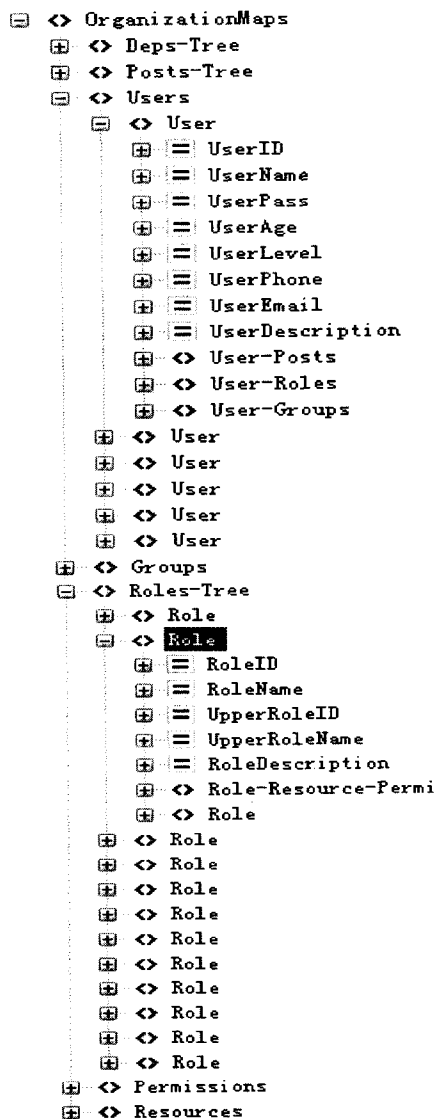


Figure 5. Structure graph of XML Workflow Organization Model

which contains self-mapping as a Tree. That is just one of the most particular characters that XML owns.

2) For the one-to-many mapping that is to say for the relationship between ST_DEP and ST_POST, we design the attribute DepId in every ST_POST record to reference its department. That is the same as for some other special mappings (See section 3.1).

3) For the many-to-many mappings, we design the reference also but just in some particular objects' records.

4) As mentioned in section 3.1, the most important relationships which connect these two parts are the USER_ROLE and ROLE_GROUP relationship objects.

So for RBAC part, we design references to both ST_RESOURCE and ST_PERMISSION in ST_ROLE's records. For real enterprise organization structure part, we design references to both ST_POST and ST_GROUP in ST_USER's records. And for the relationships connecting these two parts that is to say the relationships USER_ROLE and ROLE_GROUP, we design references to ST_ROLE in both ST_USER and ST_GROUP.

The Figure 5 is the structure graph of our XML Workflow Organization Model.

4 Summary

The Workflow Organization Model plays a very important role in workflow system. The flexibility, applicability and reliability are three of most important factors that will impact the whole quality of workflow system. To design a model of high-flexibility, we developed this model based on RBAC and supported many particular situations in that may happen in the complicated enterprise organization structure, such as that an employee can have many posts and be belonged to many departments. To design a model of high-applicability, we developed the XML-styled organization model, so that the model can be used in Web Service, BPEL and other technical applications. To design a model of high-reliability, we developed encoding and decoding mechanism for these organization data, and we also can add client authorization mechanism for these data.

In this paper, the DB organization model and XML organization model are independent at a certain extent. These two models especially XML organization model can be used respectively and independently. That's also one of the aims of our design.

References

- [1] Sandhu R, Coyne E J, Feinstein H L, *et al.* Role based