

Application of Java Content Repository in Electronic Official Document Flow

Changgen Jiang
 School of Software
 Jiangxi Normal University
 Nanchang, China
 J741229@163.com

Abstract-Electronic official document flow system mainly involves official document, attachment and other binary objects, database is not suitable to handle these unstructured data, file system does not support transaction, versioning and query language. This paper adopted java content repository to handle these binary objects, solves the storage, transaction, versioning, lock, access control and full-text query of binary objects.

Keywords-Official Document Flow, Java Content Repository, Electronic Government Introduction

I. INTRODUCTION

Electronic government is digital interactions between a government and citizens, government and businesses, government and employees, and also between government and governments. E-government is one of the most important areas of information technology. Along with the completion of the construction of hardware and network, government departments strengthening the construction of the software, have developed many software systems, these software systems aid the government departments improve the efficiency and ability, so provide more science and effective service for enterprise, citizen, economy and social.

Processing official documents is one of the very important work of the government department, it embody the function of the government department. The traditional way is carry through paper documents, this way has follow short, such as long cycle, low efficiency, high cost and so on. With the work flow and digital signature technology more and more mature, so processing official documents through network is possible, the demand of government department is more and more urgent. Processing official documents through network can improve efficiency and save cost.

The official document flow system implements all business processes, such as draft, review, countersign, audit, sign and file. In the process of the official document flow, the objects to be involved include official document, form and attachment and so on. some systems handle these objects through database, although database is very suitable to handle structured data, but not suitable to handle unstructured data, such as image and document. Some systems handle these objects through file system, although file system has more performance, but did not provide query language, also did not support transaction and reflect the relationship between the data.

Content Repository API for Java is a specification for a Java platform application programming interface to access content repositories in a uniform manner. It very suitable to the application, which require storing documents and other binary objects, in addition to object storage, the JCR provides: APIs for versioning of data; transactions; observation of changes in data. Because JCR does not bound to any particular underlying architecture, so can share data between different architecture platform.

II. JCR

A. Overview

With the increase in demand for content management system, software company have developed a large number of content management system, different company provides different API to access their content repository, this way can not share data between different content repository. Content repository API for Java is a specification for a Java platform application programming interface to access content repository in a uniform manner. A content repository is a hierarchical content store with support for structured and unstructured content, full text search, versioning, transactions, observation, and more. A JCR is a type of object database tailored to storing, searching, and retrieving hierarchical data. The JCR API grew out of the needs of content management systems, which require storing documents and other binary objects with associated metadata; however, the API is applicable to many additional types of application. In addition to object storage, the JCR provides: APIs for versioning of data; transactions; observation of changes in data; and import or export of data to XML in a standard way. The specification was developed under the Java Community Process as JSR-170 and as JSR-283.

B. Content Repository Model

Content Repository consists of one or more workspace, the data in a workspace consists of a tree of nodes with associated properties. Data is stored in the properties, which may hold simple values such as numbers and strings or binary data of arbitrary length, the type of the property can be one of STRING, BINARY, LONG, DOUBLE, BOOLEAN, DATE, PATH, NAME, REFERENCE and UNDEFINED. Node is used to build the path of the data, there are two categories of node types, primary and mixin. Every node has a primary node type assigned to it upon

creation. In addition, a mixin node type may be added to a node later in its lifecycle. The primary node type of a node usually defines node structure related to the problem domain being modeled. Mixin node types usually specify additional properties or child nodes related to a capability being added to the node. These capabilities may include generic repository-level functions as in the case of the built-in mixins `mix:versionable` and `mix:lockable`, for example, or domain-level capabilities such as a (hypothetical) `myapp:Emailable` mixin type that adds the property `myapp:emailAddress` to a node. Content repository model as shown in Fig. 1.

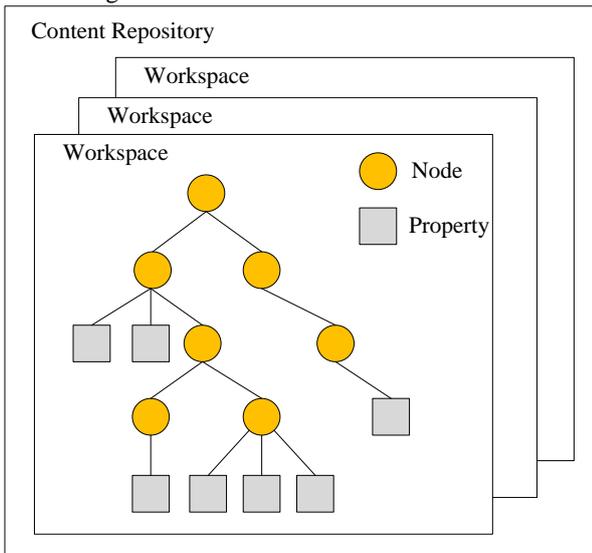


Figure 1. Java Content Repository Model

Every primary node type must be the subtype of at least one existing node type. Mixin node types do not have to have supertypes. The `Item` is the base interface of `Node` and `Property`, each node type of `Property` has one and only one parent node type of `Node`, each node type of `Node` has only one parent node type of `Node` and one or more child node type of `Item`. Node class diagram as shown in Fig. 2.

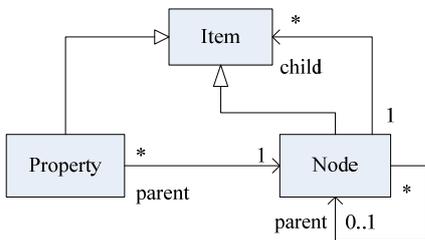


Figure 2. Class Diagram

C. Java Content Repository Implementation

The Apache Jackrabbit_ content repository is a fully conforming implementation of the Content Repository for Java Technology API. Jackrabbit support for structured and

unstructured content, full text search, versioning and so on, Jackrabbit implements the function of Level1, Level2 and Advanced Options defined by JCR. Jackrabbit-webapp component supports web-based application, Jackrabbit-ocm component is an object/content mapping framework, implements the mapping between Java object and JCR node, Jackrabbit-spi component support for remote access, Jackrabbit-core component contains the core of the fully JSR 170 compliant Apache Jackrabbit content repository implementation.

III. APPLICATION OF JCR

D. Repository Model

There are two types of the official document, receive and publish. Every official document have properties, title, category, drafter, draft time, secret level, document number, document file, attachment file and so on. The official document organized by year. Repository model as shown in Fig. 3.

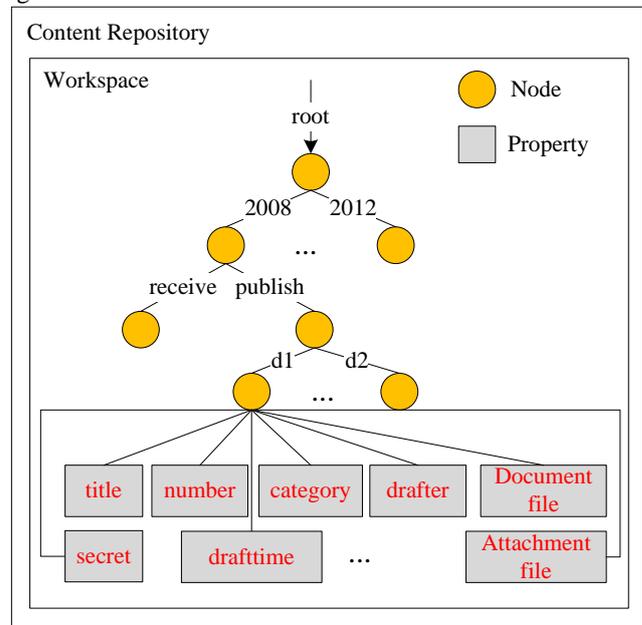


Figure 3. Repository Model

E. Initialize Content Repository

Firstly, it creates a transient repository proxy that will use the repository configuration file and home directory specified in system properties `org.apache.jackrabbit.repository.conf` and `org.apache.jackrabbit.repository.home`. Then authenticates the user using the supplied credentials invoke login method of transient repository proxy. Lastly get `VersionManager` object, `LockManager` object, `QueryManager` object and `AccessControlManger` object. Snippet of code as following.

```
String path =
ServletContext.getContext().getRealPath(_/_);
Properties props=new Properties();
```

```

props.setProperty(RepositoryFactoryImpl.REPOSITORY_C
ONF, path+_/config/repository.xml_);
props.setProperty(RepositoryFactoryImpl.REPOSITORY_
HOME, path+_/repository_);
System.getProperties(props);
repository=new TransientRepository();
session=repository.login(new
SimpleCredentials("admin","admin".toCharArray()));
ws=session.getWorkspace();
vm=ws.getVersionManager();
lm=ws.getLockManager();
qm=ws.getQueryManager();
acm=session.getAccessControlManager();
ws.getNamespaceRegistry().registerNamespace("bzoa",
"http://localhost/bzoa");

```

F. Store Official Document

Official document includes two kinds of data, primary type data and binary type data. A official document needs to be handled by different peoples at different step in the flow process, different people has different privilege, some peoples can modify official document file, other peoples can only read file. Official document file to be finalized after draft, review and countersign, it can not be changed after issued. Program implements versioning, access control and lock by add the mixin node type named mix:versionable, mix:lockable and rep:AccessControllable to file node. Snippet of code as following.

```

Node pnode=session.getNode("2012/publish");
Node dnode=pnode.addNode("d1",
NodeType.NT_FOLDER);
dnode.setProperty("bzoa:title", title);
dnode.setProperty("bzoa:number", onumber);
dnode.setProperty("bzoa:category", categoryid);
dnode.setProperty("bzoa:drafter", userid);
File file=new File(path+"/d1.pdf");
FileInputStream in=new FileInputStream(file);
Binary data=new BinaryImpl(in);
in.close();
Node fileNode=dnode.addNode("file",
NodeType.NT_FILE).addNode(Node.JCR_CONTENT,
NodeType.NT_RESOURCE);
fileNode.setProperty(Property.JCR_ENCODING, "utf-8");
fileNode.setProperty(Property.JCR_MIMETYPE,
"application/pdf");
Calendar lastModified = Calendar.getInstance ();
lastModified.setTimeInMillis (file.lastModified());
fileNode.setProperty(Property.JCR_LAST_MODIFIED,
lastModified);
fileNode.setProperty(Property.JCR_DATA, data);
fileNode.addMixin(NodeType.MIX_VERSIONABLE);
fileNode.addMixin(NodeType.MIX_LOCKABLE);
fileNode.addMixin(NameConstants.REP_ACCESS_CONT
ROLLABLE.toString());
fileNode.addNode("rep:policy",
NameConstants.REP_ACL.toString());

```

```

session.save();
vm.checkin(fileNode.getPath());

```

G. Full-Text Search

Search official document is a common demand, generally, official document by title, number and drafter. Some times, we needs to search content of official document file. We can adopt full-text search tools to meet the demand. But JCR provides more simpler way to implement full-text search, text extraction is done in a background thread when add or change data of property, we can search content of official document file by QueryManager which JCR built-in function, like search data by JDBC. snippet of code as following.

```

List<OfficialDocument> list;
String sql="select * from [nt:resource] as file where
contains(file.*, 'paper')";
Query query=qm.createQuery(sql, Query.JCR_SQL2);
QueryResult qs=query.execute();
List=documentWrapper.wrap(qs);
return list;

```

CONCLUSION

This paper adopted java content repository to handle binary objects in electronic official document flow, such way not only solves the storage of binary objects, but also meets the advanced demands, versioning, lock and access control, provides a more suitable approach than database and file system to handle binary objects.

REFERENCES

- [1] S.J. Xue and M. Cheng, "Research on Java Content Repository and Application in CMS", Computer Technology and Development, vol. , No. 1, 241-244, Jan. 2009
- [2] S.Y. Jing, "JCR based content management system", Journal of Changchun University of Technology(Natural Science Edition), vol. 32, No. 4, 381-384, Aug. 2011.
- [3] L.J. Wang, Y. Li and Y.C. Wang, "Information Warehouse System of Tobacco Company Based on Content Management", Computer Engineering, Vol.36, No. 23, 40-43, December 2010.
- [4] Q. Wang and Z.S. Li, "Design and realization of enterprise information releasing system based on JCR", Computer Applications, Vol. 28, No. 12, Dec. 2008.
- [5] S.L. Liang, and X.Y. Zhang, "Two-level document flow system for universities", Journal of Computer Applications, Vol. 30 Suppl. 2, 328-331, Dec. 2010.
- [6] <http://jackrabbit.apache.org/>
- [7] <http://www.ibm.com/developerworks/cn/opensource/os-cn-jcr/index.html>(August 11, 2011) | N. Dong
- [8] http://en.wikipedia.org/wiki/Content_repository_API_for_Java