Optimization for Web-based Online Document Management

Wenzhi Cheng[1], Yi Yang[2], Liao Zhang[3], Lian li[4]
School of Information Science & Engineering, Lanzhou University, Lanzhou, 730000, China
chengwenzhi@126.com[1], yy@lzu.edu.cn[2], zhangliao06@gmail.com[3], lil@lzu.edu.cn[4]
Corresponding Author: yy@lzu.edu.cn

***Abstract*—In this paper, we construct a web-based document life-cycle management model. The model manages documents which consist of the institute library from their creation to the archive state. For an online office system, we aim at solving three issues: network delay, version storage problems and deletion strategy. To solve network delay, we propose both local and online document synchronized editing model. In addition, we combine the longest recursive chain with recursive chain time to optimize the system response time. In order to optimize documents to be deleted, we propose a two-step optimized method. In the performance test, the effectiveness of the method is confirmed to solve the issues of documents management.**

Keywords-synchronized editing modely; version storage optimization; deletion strategy; two-step optimized method;

I. Introduction

Every company has its own official documents, how to manage these documents effectively becomes a problem, which reflected in geographical isolation, numerous documents, changing requirements and unmet specifications. These risks have potential to turn any companies into a mess. It is possible to minimize these risks by improving Web-based Life-cycle Process Management (WLPM).

With more companies start online office, we have developed an online document management system: The Wanwei User-defined Layout Document Management System (WULDM). In this system, an ordinary user could create a document and manage it, while the administrator has the ordinary user's rights and can reviews all the documents. With the system, the company is able to manage documents effectively.

It is well known that the online office is confronted with many problems. Overall, there are three main points: (1) network delay or login failure, (2) file storage problems, and (3) deleted file recovery problems. These three issues will be solved well in this paper.

II. Model Structure

Document Life-cycle Management System (DLMS) is a system which manages the creation, modification, finalization, submission, audit, archive and deletion for each document. In order to ensure the effective implementation of the system, we have used the following techniques.

- Permission control, the administrator can easily define the various roles and permissions to ensure access security. [1]
- Cross-platform capabilities, java technology, which is unlimited from operating system. [2]
- Process monitoring capabilities, which could real-time track the workflow runtime, and graphically display the progress of the workflow runtime as well as workflow running statistics. [3, 4]

Document Life-cycle Management is an office automation system [5], shown as Figure 1. One document can be created and then modified by one user. After finished, this document could be finalized. If the administrator is satisfied with the document, it could be archived into the server; otherwise, sent it back for modification.

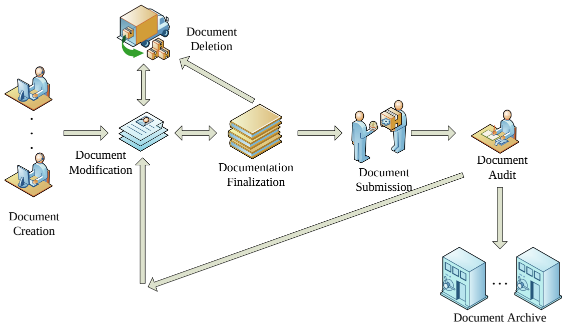

Figure 1. Document Process Management.

A. Document Creation

A user can create a user-defined document. At the same time, the system will record information, including: creation time and file type.

In addition, the administrator will get all the information of the document except content, for real-time monitoring.

B. Document Modification

Document modification will be recorded automatically, including its name, modification time and content. For well review, each modification will lead a new version to save.

One document can be removed by its creator, once it doesn’t meet the specification. To avoid mistake, each deleted document will be stored in the Recycle Bin, and they can be restored at any time.

For real-time monitors the progress of a document, the administrator can view all the information except its content.

C. Document Finalization

To distinguish the completed documents from the uncompleted ones easily, one document could be finalized after completion. In this state, the system only records the finalized information, but not send the content to the administrator for review. For better user experience, one finalized document can be modified again and then finalized.

Document information will be automatically recorded by the system, including: modification time, finalization time, name and size. To manage documents effectively, the administrator will know recorded information from system.

D. Document Submission

Document submission is the bridge for the user and the administrator. Once one document has been completed, the user will submit it to the administrator for review. Meanwhile, the system will record the document information, including submission time, name and size. For better management, the system will send a message to the administrator for attention.

E. Document Audit

One document will be audited after the administrator received audit information. For facilitation management, he will review whether the document conforms to the specification. If the document doesn't meet specifications, he will notice the user to modify it. On the other hand, if one document is approved, it will be archived for inspection.

At the same time, the system will record the audit information, including: the audit time, name, size, author and the information whether the document passes the audit.

F. Document Archive

The document will be on the archived state after it is approved. The archived document uses distributed storage, which back-ups multiple copies in multiple servers. Multiple copies prevent the documents from lost in accident, such as storage server damage, disk crash, and illegal operation.

To save the archived documents permanently, we have taken a cloud storage strategy. When the archived documents reach 64M, the system will merge these documents into a 64M file, compress it, and then upload to the cloud.

An archived document has it own permission: (1) the document's author, (2) the administrator, and (3) the user who is allowed by the administrator.

Different documents are put into different categories. With the increasing archived documents, they will eventually become a library. As the document library is large enough, the user could take advantages of the library. When the archived document is used as references, the user writes a new document easily. The library will save their time and improve their work efficiency.

Of course, the library follows the rights management, which ensures the interests of the institute and prevents the users from divulging documents.

In addition, the system has a full-text search function. If similar documents exist in the library, the administrator gives permission to the user. With the similar documents, the user can also save the writing time, improve work efficiency.

G. Document Deletion

If one document doesn't meet the requirements, such as contents error, deviating topic or audit failure, then it may be deleted. However, if you regret to delete it, but it has already been deleted, this will cause some loss. Therefore, the deletion management is very necessary. In subsequent chapters, the paper will give a strategy to solve this problem.

From A to G, the paper has introduced a simple DLMS. The system can perfectly solve the problem of an archived document from creation to archive. In the next section, this paper will focus on the optimization of the system.

III. SYSTEM OPTIMIZATION

As the previous section described, an online system will emerge three main problems, including: (1) network delay or login failure, (2) file storage problems, and (3) deleted file recovery problems. In this section, we will focus on solving these problems, and giving the optimization strategy.

A. Solving the problems of network

Network problem mainly has two parts: (1) network delay; (2) network disconnected. [6] This paper uses the following method to completely solve network problems.

1) Network delay

Network delays plagued many online systems, and affect the performance of those systems, such as the E-mail system. Assume that when you have written an email with a browser, readying to send, and then experiencing network latency issues, all the content you wrote may be lost, which will cause significant loss and you need to re-write the E-mail.

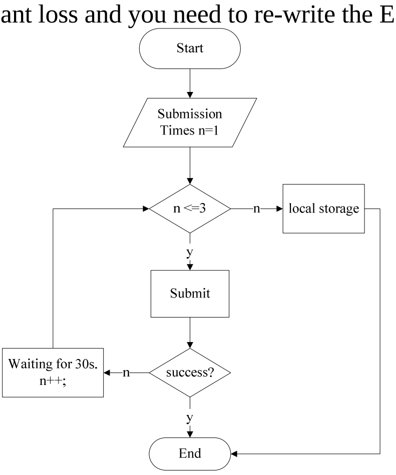

Figure 2. The Submission Process of Network Delay

In WULDM system, we use the delay control to ensure that the written document is not lost. This process can be described by the flowchart, as shown in Figure 2.

- Every five minutes, the system automatically submits data to the server for storage. Meanwhile, the server will save it into a temporary file.
- When the user saves a document or the system automatically save it, the network delays. Then, the system will wait 30 seconds, and continue to submit the document. If re-submits failure, the system will continue this process. When this submission process more than three times, the system will notify the user, allowing the user to handle this event.

- When the system notifies the submission failure, the user can choose to save it in the local file system.

To log onto the system, we take the VPN (Virtual Private Network) login strategy. As VPN is set up in the institute, a user login to the system easily when he is on a business trip.

In practice, we have proved that the use of VPN login and submission process strategy can be a good solution to the issue of network latency.

2) Network disconnected

Assuming that the network can't connect, the institute will be a mess. Meanwhile, the institute's subsidiaries also can’t work properly, which will bring a fatal blow to the institute. To solve the problems, we use a client, which is actually a browser plug-in.

- The document takes local and server storage at the same time. When the user log in the system, the file has priority to be stored on the server. As the user saves a file, the system will automatically download the latest documents from the server.
- When the user can’t connect to the institute server, he could open local document for work. The next time when the system detects that the network connects, the system will upload the local document to the server, and saved as a new version.
- As shown in Figure 2, the system runs to n>3, and then the document will be stored to the local.
- To prevent local files and server files conflict, we set a timestamp. When the local modification time is later than the server, the system uploads the local file. Or the file will be downloaded from the server.
- The issue of priority. When the user is able to log onto the server, the server will get a higher priority, or local file has a higher priority.

Through the above strategy, even if with the broken network, the users can also work, which minimizes the loss of the institute.

B. Documents storage

In the system, the server will save a new version when the user saves the document. Obviously, there are many advantages: (1) the user can view the previous version information to facilitate document editing; (2) Facilitating comparison with previous versions, the user could select the best version; (3) The user can restore the previous document.

However, too many versions will give the server heavy burden. Supposed that a user has a 10M document, and then he modified the 100 version, the document will occupy 1G storage. Therefore, this section proposes a better storage strategy and compares the strategy with SVN (Subversion).

1) Version of the Storage Optimization

At present, most of the online systems use SVN for version control, which stores differences between successive versions. Retaining the differences between successive versions, this method stores a number of versions effectively.

However, if versions are too many, the speed of reading a document will becomes very slow. Supposed that a file has 100 versions, the system will be recursive call to the 100 version when someone opens it. The recursive process is shown in Figure 3, which is a classic SVN mode. [7]

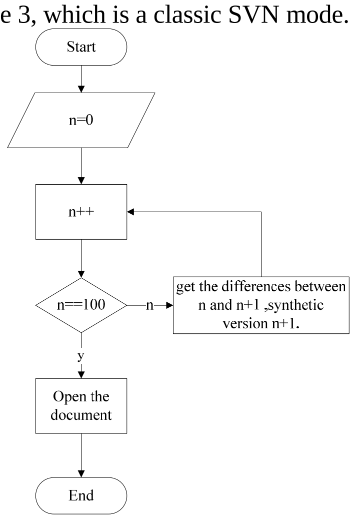
Start
n=0
n++
n==100
n
get the differences between n and n+1 ,synthetic version n+1.
y
Open the document
End

Figure 3. The Recursive Process of SVN

In Figure 3, if the recursive chain is too long, it will inevitably affect the reading time. Actually, the document will be taken some time to download. If the user's waiting time is too long, the user experience will be affected. The paper proposes the following method to solve this problem.

We define that reading the recursive chain time on the server is t_1, downloading the document's time is t_2, and the waiting time is t, where $t = t_1 + t_2$.

In practice, if the user waits more than 5 seconds, his work will be affected. Accordingly, as long as the total time t<5s, his work won’t be affected.

In the system, we know the size of the file from database, and then using 1M as the reference bandwidth, which is easy to calculate the download time t_2. Now, we use the formula $t = t_1 + t_2$ to calculate the recursive chain time t_1.

Therefore, we only handle the recursive chain time t_1 to optimize for version control. This paper uses the following algorithm to optimize recursive chain time. This process can be described the following flowchart, as shown in Figure 4.

- Firstly, we control the longest recursive chain. We set the longest recursive chain interval for 50 versions. If recursion interval is no more than 50 versions, the system will save the differences between successive versions. Otherwise, the system will update the current version as a new recursive file. Thus, we use 50 versions for intervals, which will reduce the recursive chain match time.
- Furthermore, we take the maximum time control. Some files may exceed the predetermined recursive chain time. At this time, we use the calculated recursive chain time t_1 as the maximum time. When the recursive chain runs more than a predetermined time t_1, the system get the current recursive document as a new recursive serial number d, and stored the number d in the database. Meanwhile, the system will save the document which is the new recursive serial number d as a new document.

Finally, after the above two-step training, the response time of opening a document becomes fast. In the flowchart, d represents the recursive serial number, which stored in the database (DB); k represents the recursion interval; N represents the number of current version; T represents the recursive chain time t_1.

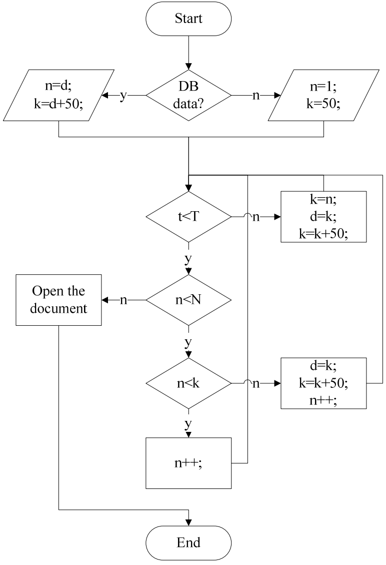

Figure 4. The Recursive Chain Time Control

In practice, the system is able to control the document's opening time less than 5 seconds, which meets the user needs.

2) Compared with SVN

In figure 3, this is a typical SVN system. While in figure 4, we improve a new version control system. Compared with SVN, our system has some advantages as follows.

- Waiting time is shorter, for reducing opening time.
- Excellent performance. Taking a two-step training method, we optimize the system's performance.
- Stronger anti-interference. In SVN, if any one of the version information files is missing, the document couldn't be opened. While in our system, if the version files suddenly lost one, the document can also be opened.

In the actual test, given 1M bandwidth, our system is more excellent than SVN in the same environment.

C. Deletion strategy

In the previous chapter, the user may mistakenly delete the document, and if there are no measures to restore the document, it will take some losses. In our system, we take the two-step Remove Snap to solve this problem.

- The first step, when a user deletes a file, it will be moved to a temporary folder, which is specifically set to store the deleted files.
- The second step, if the user finds some useful documents in the temporary folder, he could restore them and continue to modify. Meanwhile, he can delete some useless ones; these deleted files will be moved to the recycle bin. The recycle bin files will be stored ten days and then be automatically deleted. Before automatically deleted, files could be restored.

In practice, using a two-step optimizing method can minimize the losses for the users.

IV. Conclusion and Future Work

In this paper, we propose a document life-cycle management system. In addition, we address the issues of network delay, documents storage and deletion strategy. Through both the local and online synchronous editing method, we solve the network problems. Furthermore, we take the control of the longest recursive chain and recursive time to solve the problem of file delay. Finally, we take the two-step Remove Snap to manage the deleted files. In the actual test, the effectiveness of our methods is confirmed to solve the issues of documents management.

Further, we plan to put the system into a demonstration system of document process management in Gansu Wanwei trying to solve problems of documents management.

Acknowledgment

This work was supported by the Natural Science Foundation of P. R. of China (90912003, 90812001 and 61073193), the Key Science and Technology Foundation of Gansu Province (1102FKDA010), Natural Science Foundation of Gansu Province (1107RJZA188), the Fundamental Research Funds for the Central Universities (lzujbky-2012-47, lzujbky-2012-48).

References

[1] Yang Zhi-Guang and Ma Zi-Qin, "Research on Web-Based Typical Process Management," 2010 International Conference on Electrical and Control Engineering, 2010,pp. 2983-2987.

[2] Karlie Hutchens, Michael Oudshoom and Kevin Maciunas, "Web-Based Software Engineering Process Management," 1997 IEEE, pp. 676–685.

[3] Xiaoqing Gong, "Research on Web-based Distributed Workflow Management System services," Xi'an, Xibei University, 2004, pp. 12-20.

[4] He JunHua, "Research the office automation system software based on workflow technology," Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on, May 2011, pp. 428-431.

[5] Wei Wu, "The analysis and design of office automation system based on workflow," Electronic and Mechanical Engineering and Information Technology (EMEIT), 2011 International Conference on, Aug. 2011, pp. 223-225.

[6] Ishii, Sejima and Watanabe, "Effects of delayed presentation of self-embodied avatar motion with network delay," Universal Communication Symposium (IUCS), 2010 4th International, 2010, pp. 262 – 267.

[7] Ifrah, S. and Lorenz, D.H., "Crosscutting revision control system," Software Engineering (ICSE), 2012 34th International Conference on, June 2012, pp. 321-330.