Distributed Services Architecture in dLibra Digital Library Framework

Cezary Mazurek and Marcin Werla

Poznan Supercomputing and Networking Center
Poznan, Poland
{mazurek,mwerla}@man.poznan.pl

Abstract. Architecture is one of the key factors influencing all distributed system components. It often decides about overall functionality, performance and flexibility. In this article we intend to describe the design of the first Polish digital library framework called dLibra, which has been developed in PSNC since 1999. We show how architecture based on modular and distributed services can be used to split digital library functionality. Such division gives opportunities for improving services availability and expansion possibilities. We also want to present mechanisms that we use to assure stable and continuous work of our distributed digital library framework, making it independent of various negative circumstances.

1 Introduction

The dLibra Digital Library Framework [1, 2] has been developed in Poznan Supercomputing and Networking Center since 1999. The first dLibra-based digital library (DL) was the Digital Library of the Wielkopolska Region (WBC) [3]. It was started in October 2002 and now it consists of over 3000 various publications grouped into four thematic collections: cultural heritage, regional materials, educational materials and music notes. Such number of publications makes from WBC one of the largest Polish digital libraries. In the end of November 2004 the second dLibra-based digital library was deployed – the Wroclaw University of Technology Digital Library (BCPWr) [4]. There are also four other test dLibra installations academic libraries, and in the nearest future three new regional digital libraries will be started.

Due to the diversity of mentioned digital libraries, many different aspects must be taken into account during the dLibra development. Each DL has its own specific publications – for example the majority of WBC publications are relicts of writing and old documents associated with social life of the Wielkopolska Region. Such resources are mostly stored in a graphical form, in formats like DjVu, PDF or JPG scans. All that publications consist of quite large files and their content is often not searchable. On the other hand, there are academic DL systems, like one of test dLibra installations in AGH University of Science and Technology, where a typical publication is an academic script stored as a set of HTML pages or small PDF file with searchable text content. Such differences requires support in many areas – from format dependent publication structure analysis during the publication upload process, to sophisticated mechanisms for content indexing and searching, to different publication view and download possibilities. Another important element is the amount of stored publications. When there are many gathered resources, and a large number of readers accessing it, the overall system performance becomes a crucial parameter.

In the following sections we want to show the way in which we designed the internal dLibra architecture and its basic mechanisms to create an efficient, flexible, distributed and error-resistant digital library system. The next section describes the structure of distributed services architecture developed in the dLibra Digital Library Framework. We also show an overview of these services functionality. The third section is focused on mechanisms for improving services reliability and availability in the dLibra framework. We try to demonstrate our approach to improving the dLibra stability in some extreme situations, like very heavy user requests load or network communication errors. Those mechanisms are an integral part of dLibra services and service management system. In the last section we point out some directions of our future, digital library-related works.

2 dLibra Architecture

The initial dLibra architecture and design was based on experiences from previous PSNC projects. We assumed that the dLibra environment should consist of a number of portable, distributed services. Portability was achieved by choosing Java™ as the programming language. Further works and practice from dLibra deployments formed the current dLibra structure. This structure is based on a set of cooperating remote services. All these services together create the complete dLibra-based digital library. Each service can be started on a separate computer, but they can also be connected into service groups running on the same machine. When services are started on different hosts, they use Java RMI technology to communicate with each other [5]. Six of dLibra services give together the entire dLibra server functionality. These services are:

- The Metadata Server – gives a possibility to define, modify and remove metadata attributes that are used to describe digital library publications. It also gives access to dictionaries and thesauri with values of all attributes. It is responsible for managing digital library directories and collections. In addition, it allows adding, modifying and deleting publications, and it has possibilities to manage lists of languages defined in the DL system. Moreover, it has a module for performing periodic metadata consistency test.
- The Content Server – gives access to all gathered digital content. Before sending content to the client, this service is able to compress it or encrypt and send securely. The Content Server is also used to store the publications content. Resuming is supported during both publication upload and download.
- The Search server – allows users to search through all gathered content and metadata. It also contains indexing functionality, which prepares indexes used during search.
- The Distributed Search Server – is used to harvest remote dLibra instances by means of the OAI-PMH [6] protocol. It also gives the user a possibility to search through gathered remote metadata. In fact, any OAI-PMH-enabled repository can be harvested and searched using that service.

- The User Server – contains all user-related data and allows users authentication and authorization. It is also used to create groups of users and to grant users different digital library rights, from library administration to simple publication view.

As we mentioned before, all the above services give together the entire dLibra digital library functionality. However, at least two more elements are required to create a fully functional system. There must be a possibility to connect all these services and create an entry point to the system for both external applications and users.

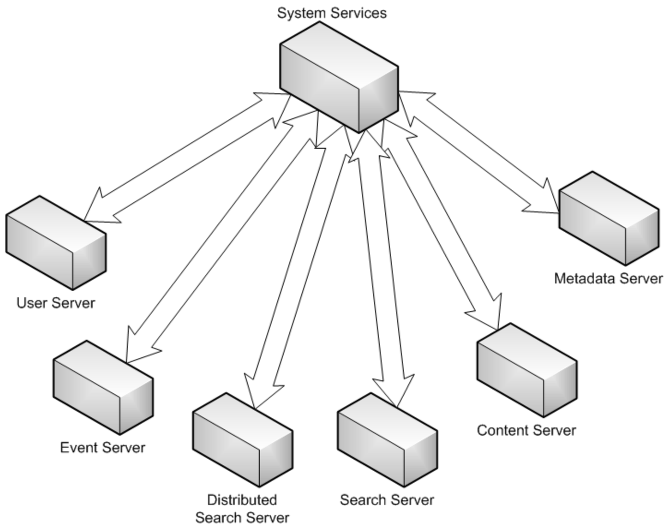

Fig. 1. Distributed dLibra services architecture

Connection between services is achieved with two additional services (see Figure 1). The first of them is a service called the System Services. It can be treated as a broker of services for single instance of the distributed digital library. It allows inter service communication and handles services addresses resolving, connecting and authorization. For example, when the Search Server wants to refresh its indexes, it asks the System Services for the Content Server and the Metadata Server. The System Service checks if such services are registered, if they are available and if the Search Server is authorized to access them. If all those conditions are met, as a response the Search Server receives references to the requested services. In order to become available to other services, each service must register itself in the chosen System Services. Services registered in one System Services create a digital library.

The second additional system level service is the Event Server. It allows services to communicate with the event messaging system. It is very useful when one service wants to notify some other services about a particular event. A good example of this mechanism can again be a process of refreshing search indexes (see Figure 2). Just after start, the Search Server registers in the Event Server for events related with the modification of gathered content and metadata. When a new publication is added, modified or removed, the Metadata Server sends an event notification to the Event Server. Next, the Event Server forwards this event to all services registered for this event type. After receiving such event, the Search Server can decide if index refreshing is required or maybe just some data should be removed from the index.

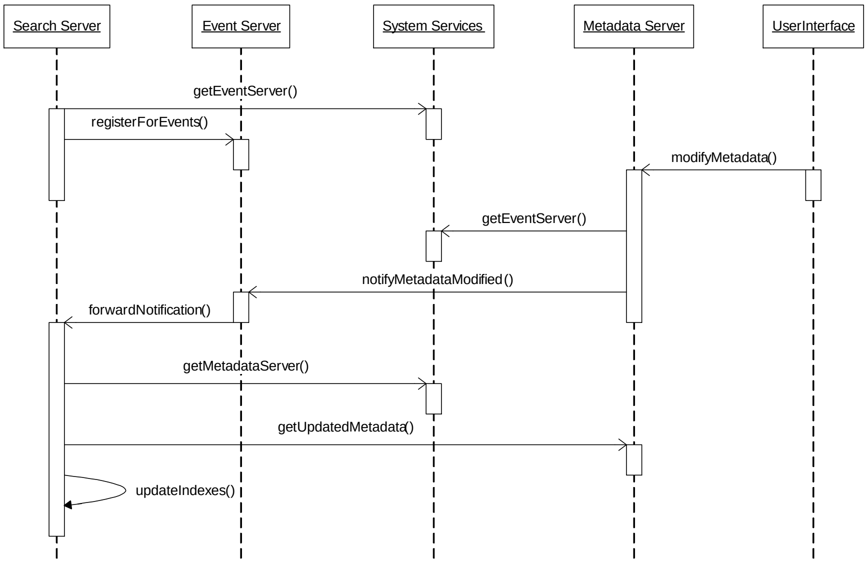

Fig.2. Sequence diagram describing Event Server event passing

In Figure 2 there is an element called User Interface. This corresponds to two additional parts of the dLibra Framework. One of them is WWW Service and the second is the Editor/Administrator application. The WWW Service is designed as a read only entry point to the system. It can be used by readers to access gathered resources. Content browse and searching are the main functionality of this service, but it is also an OAI-PMH data provider, and it has many user-friendly features like RSS [7] feeds with information newly added publications, publications ranking etc. This functionality is realized with the use of all other dLibra services reached through the System Services. The Editor/Administrator Application is an application for librarians and library administrators. It allows adding and modifying library resources and managing all library items.

3 Improving services availability in dLibra framework

The distributed architecture of the described dLibra services requires additional mechanisms for improving system reliability and availability. When one of the services stops responding, the library may become less functional (for example in case of the Search Server failure) or may not be functional at all - when the User

Server or Metadata Server fails. To prevent such situations, a number of mechanisms were introduced.

The first of them is the way of service resolving done by the System Services. There is a possibility to create such services configuration, in which multiple instances of the most crucial services are started. Before the System Services gives one service access to another service, it tests the requested service functionality. When the tested instance of a given service fails, the System Services can return reference to other instance. Such instance switch is transparent to other services. With addition of services load monitoring functionality, this mechanism can also be used for load balancing between service instances.

The second mechanism is internal services monitoring. Each service is periodically checked if it is responding or has enough processor time for its tasks. This check is performed by a special service wrapper based on an open-source Java ServiceWrapper project [8]. The Wrapper can restart or shutdown the service when, for some reason, it stops responding or when host processors are overloaded for a longer period of time (for example during DoS attacks [9]). This service monitoring is done locally so it is independent of the state of network connections.

Another reliability improving mechanism is implemented in events sending and receiving parts of the dLibra framework. When service generates an event, it is not directly sent to the Event Server, but it is stored in a persistent storage. This storage is implemented with Hibernate [10], so it can be based on many types of relational databases. All stored events are read by a specialized Event Sender thread. This thread tries to send events to the Event Server. If connection to the Event Server is lost, all events stay in the storage until there is a possibility to send them again. On the other hand, when the Event Server retrieves an event, it also stores the event before trying to send it to registered services. Each service, while registering for events, gives the Event Server special timeout parameter. This parameter describes how long the events should be stored in the Event Server, if the registered service becomes unavailable. If the registered service becomes available again, all events stored for this service will be passed to it.

4 Future works

We think that next dLibra development stages will bring this distributed digital library framework closer to grid technologies. In order to do so, it will be necessary to extend our services model. Each service should gain the ability of describing itself with metadata. On top of the System Services there must be some kind of a new, much more advanced service – a dynamic distributed digital library services broker.

This should allow automated service discovery and creation of virtual DL organizations. Such active organizations of services could be used to create distributed digital collections from resources gathered in heterogeneous DL systems. We can also imagine Information Retrieval Grid services based on different distributed digital libraries [11, 12]. By creating an environment for advanced cooperation of computational grid services, grid data management systems and digital libraries we want to give an opportunity for advanced usage of digital libraries in sophisticated grid scenarios [13].

References

[1] Gruszczyński, P.; Mazurek, C.; Osinski S.; Swedrzynski A.; Szuber S. "dLibra Content Maintenance for Digital Libraries" in *Euromedia'2002*, pages 28–32, 7*th* Annual Scientific Conference, April 2002.

[2] Mazurek C.; Stroiński M.; Swędrzyński A.; „dLibra – Integrated Framework for Publishers and Libraries" – poster at 7th European Conference on Digital Libraries, Torndheim, Norway, August 2003.

[3] Digital Library of Wielkopolska Region. http://www.wbc.poznan.pl/.

[4] Wroclaw University of Technology Digital Library. http://dlib.bg.pwr.wroc.pl/.

[5] Hicks, M.; Jagannathan, S.; Kesley, R.; Moore, J.-T.; Ungureanu, C. "Transparent Communication for Distributed Objects in Java". ACM Java Grande Conference, pages 160-170, June 1999.

[6] Lagoze, C.; Van de Sompel, H. – "The Open Archives Initiative: Building a low-barrier interoperability framework", pages 54-62, Proceedings of the 1st ACM/IEEE-CS Joint Conference on Digital Libraries, Roanoke, VA, USA, June 2001.

[7] Hammersley , B. "Content Syndication with RSS". O'Reilly. 1st Edition. March 2003.

[8] Mortenson, L. "What is the Java Service Wrapper?". http://wrapper.tanukisoftware.org/doc/english/introduction.html

[9] CERT Coordination Center. "Denial of Service Attacks" http://www.cert.org/tech_tips/denial_of_service.html

[10] Cengija, D. "Hibernate Your Data". O'Reilly ONJava. 2004. http://www.onjava.com/pub/a/onjava/2004/01/14/hibernate.html

[11] Larson, R. R. "Distributed IR for Digital Libraries" in LNCS 2769, p. 487 – 498, 7th European Conference on Digital Libraries, Torndheim, Norway, August 2003.

[12] Dovey, M. J.; Gamiel, K.; "GRID IR — GRID Information Retrieval". Poster at EuroWeb 2002. Accessed from http://www.gridir.org/

[13] Kosiedowski, M.; Mazurek, C; Werla, M. – „Digital Library Grid Scenarios" in European Workshop on the Integration of Knowledge, Semantics and Digital Media Technology, 25-26.05.2004, London, U.K. Workshop Proceedings, p. 189 – 196.